Package ‘medfate’

April 23, 2025

Type Package

Title Mediterranean Forest Simulation

Version 4.8.1

Date 2025-04-22

Description Simulate Mediterranean forest functioning and dynamics using cohort-based description of vegetation [De Caceres et al. (2015) <doi:10.1016/j.agrformet.2015.06.012>; De Caceres et al. (2021) <doi:10.1016/j.agrformet.2020.108233>].

License GPL (>= 2) | LGPL (>= 3)

URL https://emf-creaf.github.io/medfate/,
https://github.com/emf-creaf/medfate

BugReports https://github.com/emf-creaf/medfate/issues

LazyLoad yes

Depends R (>= 3.5.0)

Imports ggplot2 (>= 3.0.0), meteoland (>= 2.0.0), Rcpp (>= 1.0.6), shiny

Suggests testthat (>= 3.0.0), knitr, rmarkdown

LinkingTo Rcpp, meteoland

Encoding UTF-8

NeedsCompilation yes

RoxygenNote 7.3.2

Config/testthat/edition 3

LazyData true

Author Miquel De Cáceres [aut, cre, cph] (<https://orcid.org/0000-0001-7132-2080>),
Nicolas Martin-StPaul [aut] (<https://orcid.org/0000-0001-7574-0108>),
Víctor Granda [aut] (<https://orcid.org/0000-0002-0469-1991>),
Antoine Cabon [aut] (<https://orcid.org/0000-0001-6426-1726>),
Arsène Druel [aut] (<https://orcid.org/0000-0002-3938-0085>),
Julien Ruffault [aut] (<https://orcid.org/0000-0003-3647-8172>),

Jordi Martínez-Vilalta [ctb] (<https://orcid.org/0000-0002-2332-7298>),
Maurizio Mencuccini [ctb] (<https://orcid.org/0000-0003-0840-1477>),
François Pimont [ctb] (<https://orcid.org/0000-0002-9842-6207>),
Hervé Cochard [ctb] (<https://orcid.org/0000-0002-2727-7072>),
Aitor Améztegui [ctb] (<https://orcid.org/0000-0003-2006-1559>),
Léa Veuillen [ctb] (<https://orcid.org/0000-0002-2790-3627>),
Shengli Huang [ctb] (<https://orcid.org/0000-0003-3927-7042>),
John Burkardt [cph] (Copyright holder of C++ code in 'incgamma.cpp')

Maintainer Miquel De Cáceres <miquelcaceres@gmail.com>

Repository CRAN

Date/Publication 2025-04-23 07:40:02 UTC

Contents

`defaultControl` *Control parameters for simulation models*

Description

Creates a list control parameters default values for simulations

Usage

```
defaultControl(transpirationMode = "Granier", soilDomains = "buckets")
```

Arguments

`transpirationMode`
: Transpiration model (either 'Granier', 'Sperry' or 'Sureau'). See `spwbInput`.

`soilDomains`
: Soil hydrology model (either 'buckets', 'single' or 'dual'). See `hydrology_soilWaterBalance`.

Details

The function returns a list with default parameters. Users can change those defaults that need to be set to other values and use the list as input for model functions. The relevant parameters are different for each model function.

Value

A list, with the following options (default values in brackets):

General:

- `verbose` [= `TRUE`]: Boolean flag to indicate console output during calculations. In function `fordyn` `verbose` is always set to FALSE.
- `fillMissingRootParams` [= `TRUE`]: Boolean flag to indicate that initializing functions should provide estimates for Z50 and Z95 if these are missing in the forest data. Note that if `fillMissingRootParams` is set to `FALSE` then simulations may fail if the user does not provide values for Z50 and Z95 in tree or shrub data.
- `fillMissingSpParams` [= `TRUE`]: Boolean flag to indicate that initializing functions should provide estimates for functional parameters if these are missing in the species parameter table `SpParams`. Note that if `fillMissingSpParams` is set to `FALSE` then simulations may fail if the user does not provide values for required parameters.

- `fillMissingWithGenusParams [=TRUE]`: Boolean flag to indicate that initializing functions should provide estimates from genus value, if species-level values are missing in the species parameter table `SpParams` but genus-level ones are not.
- `standResults [= TRUE]`: Boolean flag to keep stand-level results (in a data frame called 'Stand').
- `soilResults [= TRUE]`: Boolean flag to keep soil-level results (in a list called 'Soil').
- `snowResults [= TRUE]`: Boolean flag to keep snow results (in a data frame called 'Snow').
- `plantResults [= TRUE]`: Boolean flag to keep plant-level water/energy/photosynthesis results (in a list called 'Plants').
- `labileCarbonBalanceResults [= TRUE]`: Boolean flag to keep plant-level labile carbon balance results (in a list called 'LabileCarbonBalance').
- `plantStructureResults [= TRUE]`: Boolean flag to keep plant-level structure results (in a list called 'PlantStructure').
- `growthMortalityResults [= TRUE]`: Boolean flag to keep plant-level growth and mortality results (in a list called 'GrowthMortality').
- `leafResults [= TRUE]`: Boolean flag to keep leaf-level results (in elements called 'SunlitLeaves' and 'ShadeLeaves').
- `temperatureResults [= TRUE]`: Boolean flag to keep temperature results (in elements called 'Temperature' and 'TemperatureLayers').
- `subdailyResults [= FALSE]`: Boolean flag to force subdaily results to be stored (as a list called 'subdaily' of `spwb_day` objects, one by simulated date) in calls to `spwb`. In function `fordyn` `subdailyResults` is always set to FALSE.
- `fireHazardResults [= FALSE]`: Boolean flag to force calculation of daily fire hazard.
- `fireHazardStandardWind [= NA]`: Wind speed (in m/s) for fire-hazard estimation. If missing, actual wind-speed is used.
- `fireHazardStandardDFMC [= NA]`: Dead fuel moisture content for fire-hazard estimation. If missing, estimation from current weather is used.

Water balance (functions `spwb`, `pwb` or `spwb_day`):

- `transpirationMode [= "Granier"]`: Transpiration model (either 'Granier', 'Sperry' or 'Sureau'). See `spwbInput`.
- `soilFunctions [= "VG"]`: Soil water retention curve and conductivity functions, either 'SX' (for Saxton) or 'VG' (for Van Genuchten). If `transpirationMode` is 'Sperry' or 'Sureau' then soilFunctions is forced to `'VG'`. Only simulations with 'Granier' are allowed to use Saxton functions.
- `VG_PTF`: String indicating the pedotransfer functions for van Genuchten parameters (either 'Toth' or 'Carsel').
- `ndailysteps [= 24]`: Number of steps into which each day is divided for determination of soil water balance, stomatal conductance, transpiration and photosynthesis (24 equals 1-hour intervals).
- `max_nsubsteps_soil [= 300]`: Maximum number of substeps for soil water balance solving.
- `defaultWindSpeed [= 2.5]`: Default wind speed value (in m/s) to be used when missing from data.
- `defaultCO2 [= 386]`: Default atmospheric (abovecanopy) CO2 concentration (in micromol·mol-1 = ppm). This value will be used whenever CO2 concentration is not specified in the weather input.

- defaultRainfallIntensityPerMonth [= c(1.5, 1.5, 1.5, 1.5, 1.5, 1.5, 5.6, 5.6, 5.6, 5.6, 5.6, 1.5)]: A vector of twelve values indicating the rainfall intensity (mm/h) per month. By default synoptic storms (1.5 mm/h) are assumed between December and June, and convective storms (5.6 mm/h) are assumed between July and November.
- leafPhenology [= TRUE]: Boolean flag to indicate the simulation of leaf phenology for winter-deciduous species.
- bareSoilEvaporation [= TRUE]: Boolean flag to indicate the simulation of evaporation from bare soil.
- unlimitedSoilWater [= FALSE]: Boolean flag to indicate the simulation of plant transpiration assuming that soil water is always at field capacity.
- unfoldingDD [= 300]: Degree-days for complete leaf unfolding after budburst has occurred.
- interceptionMode [= "Gash1995"]: Infiltration model, either "Gash1995" or "Liu2001".
- infiltrationMode [= "GreenAmpt1911"]: Infiltration model, either "GreenAmpt1911" or "Boughton1989".
- infiltrationCorrection [= 5.0]: Factor to correct infiltration amount in the GreenAmpt1911 model in single-domain simulations.
- soilDomains [= "buckets"]: Either 'buckets' (for multi-bucket model), 'single' (for single-domain Richards model) or 'dual' (for dual-permeability model). See hydrology_soilWaterBalance.
- rhizosphereOverlap [= "total"]: A string indicating the degree of rhizosphere spatial overlap between plant cohorts:
  * "none" - no overlap (independent water pools).
  * "partial" - partial overlap determined by coarse root volume.
  * "total" - total overlap (plants extract from common soil pools).
- verticalLayerSize [= 100]: Size of vertical layers (in cm) for the calculation of light extinction (and photosynthesis).
- windMeasurementHeight [= 200]: Height (in cm) over the canopy corresponding to wind measurements.
- segmentedXylemVulnerability [= TRUE/FALSE]: If FALSE leaf and root vulnerability curves will be equal to those of stem. By default, segmentedXylemVulnerability = TRUE for transpirationMode = "Sperry" and segmentedXylemVulnerability = FALSE for transpirationMode = "Sureau".
- leafCavitationEffects, stemCavitationEffects [= FALSE/TRUE]: A flag indicating whether cavitation effects on conductance of leaves and stem are applied. Only relevant for transpirationMode = "Sperry".
- leafCavitationRecovery, stemCavitationRecovery [= "annual"]: A string indicating how recovery of previous cavitation leaf/stem xylem is done (only relevant for functions spwb and spwb_day):
  * "none" - no recovery.
  * "annual" - every first day of the year.
  * "rate" - following a rate of new leaf or sapwood formation.
  * "total" - instantaneous complete recovery.
- cavitationRecoveryMaximumRate [= 0.05]: Maximum rate of daily refilling of embolized conduits as sapwood area per leaf area (in cm2·m-2·day-1).
- lfmcComponent [= "fine"]: Plant component used to estimate LFMC, either "leaf" or "fine" (for fine fuel).

Water balance (functions spwb, pwb or spwb_day when traspirationMode = "Granier" only):

- hydraulicRedistributionFraction [= 0.1]: Fraction of plant transpiration corresponding to hydraulic redistribution.

Water balance (functions spwb, pwb or spwb_day when traspirationMode = "Sperry" or traspirationMode = "Sureau"):

- nsubsteps_canopy [= 3600]: Number of substeps into which each step is divided for multi-layer canopy energy balance solving.
- multiLayerBalance [= FALSE]: Flag to indicate multiple canopy energy balance. If FALSE, canopy is considered a single layer for energy balance.
- sapFluidityVariation [= TRUE]: Flag to indicate that temperature affects sap fluidity (and indirectly plant conductance).
- TPhase_gmin [= 37.5]: Temperature for transition phase of gmin.
- Q10_1_gmin [= 1.2]: Temperature dependance of gmin when T less than or equal to TPhase.
- Q10_2_gmin [= 4.8]: Temperature dependance of gmin when T greater than TPhase.
- taper [= TRUE]: Whether taper of xylem conduits is accounted for when calculating aboveground stem conductance from xylem conductivity.
- thermalCapacityLAI [= 1000000]: Thermal canopy capacitance per LAI unit.
- rootRadialConductance [= 4]: Radial conductance in roots (mmol·s-1·m-2·MPa-1).
- averageFracRhizosphereResistance [= 0.15]: Fraction to total continuum (leaf+stem+root+rhizosphere) resistance that corresponds to rhizosphere (averaged across soil water potential values).
- boundaryLayerSize [= 2000]: Size of the boundary layer (in cm) over the canopy (relevant for multi-layer canopy energy balance).

Water balance (functions spwb, pwb or spwb_day when traspirationMode = "Sperry" only):

- numericParams: A list with the following elements:
  * maxNsteps [= 400]: Maximum number of steps in supply function.
  * ntrial [= 200]: Number of iteration trials when finding root of equation system.
  * psiTol [= 0.0001]: Tolerance value for water potential.
  * ETol [= 0.0001]: Tolerance value for flow.

Water balance (functions spwb, pwb or spwb_day when traspirationMode = "Sureau" only):

- plantCapacitance [= TRUE]: Whether the effect of (symplasmic) plant water compartments is considered in simulations.
- cavitationFlux [= TRUE]: Whether the effect of water flux generated by cavitation of apoplasmic tissues is considered in simulations.
- soilDisconnection [= FALSE]: Whether the ability of the plants to physically disconnect their root system from the soil is considered in simulations.
- leafCuticularTranspiration [= TRUE]: Whether the effect of leaf cuticular transpiration is considered in simulations.
- stemCuticularTranspiration [= FALSE]: Whether the effect of stem cuticular transpiration is considered in simulations.
- C_SApoInit [= 2.0e-5]: Maximum capacitance of the stem apoplasm (mmol·m-2).
- C_LApoInit [= 1.0e-5]: Maximum capacitance of the leaf apoplasm (mmol·m-2).

- k_SSym [= 0.26]: Conductance from stem apoplasm to stem symplasm (mmol·s-1·m-2·MPa-1).
- fractionLeafSymplasm [= 0.5]: Fraction of the leaf resistance from leaf apoplasm to leaf symplasm ([0-1]).
- gs_NightFrac [= 0.05]: Stomatal conductance at night as fraction of maximum stomatal conductance ([0-1]).
- stomatalSubmodel [= "Baldocchi"]: Stomatal regulation sub-model, either "Jarvis" or "Baldocchi".
- JarvisPAR [= 0.003]: Parameter regulating the response of stomatal conductance to light (PAR) in the Jarvis model.
- fTRBToLeaf [= 0.8]: Fraction of surface of bark exposed to air per leaf area.

Forest growth (functions growth or growth_day):

- subdailyCarbonBalance [= FALSE]: Boolean flag to indicate that labile carbon balance should be conducted at sub-daily steps (applies only to transpirationMode = "Sperry").
- allowDessication [= TRUE]: Boolean flag to indicate that mortality by dessication is allowed.
- allowStarvation [= TRUE]: Boolean flag to indicate that mortality by starvation is allowed.
- sinkLimitation [= TRUE]: Boolean flag to indicate that temperature and turgor limitations to growth are applied.
- shrubDynamics [= TRUE]: Boolean flag to allow the application of demographic processes to shrubs.
- herbDynamics [= TRUE]: Boolean flag to allow dynamic herb leaf area as a function of shading due to leaf area of woody cohorts.
- allocationStrategy [= "Al2As"]: Strategy for allocation (either "Plant_kmax", for constant maximum plant conductance, or "Al2As" for constant Huber value).
- phloemConductanceFactor [= 0.2]): Factor to transform stem xylem conductance to stem phloem conductance (only for transpirationMode = "Sperry").
- nonSugarConcentration [= 0.25]: Non-sugar (inorganic) solute concentration (mol·l-1) in cells.
- equilibriumOsmoticConcentration [= c(leaf = 0.8, sapwood = 0.6)]: Equilibrium osmotic concentrations (mol·l-1) for leaf and sapwood cells. The difference between leaf and sapwood values helps maintaining phloem transport. The equilibrium sugar concentration is equilibriumOsmoticConcentration - nonSugarConcentration defaults to [= c(leaf = 0.55, sapwood = 0.35)].
- minimumRelativeStarchForGrowth [= 0.50]: Default minimum concentration of storage carbon (starch), relative to the maximum storage capacity, for sapwood growth to occur, when not specified via SpParams (RSSG).
- constructionCosts [= c(leaf = 1.5, sapwood = 1.47, fineroot = 1.30)]: Default construction costs, including respiration and structural carbon, per dry weight of new tissue (g gluc · g dry -1) when not specified via SpParams (CCleaf, CCsapwood and CCfineroot).
- senescenceRates [= c(sapwood = 0.0001261398, fineroot = 0.001897231)]: Default senescence rates (day-1) for sapwood and fineroots when not specified via SpParams (SRsapwood and SRfineroot). Defaults are equivalent to 9%, 5% and 50% annual turnover for gymnosperm sapwood, angiosperm sapwood and fine roots, respectively.

- maximumRelativeGrowthRates [= c(leaf = 0.09, cambium = 0.005, sapwood = 0.002, fineroot = 0.1)]: Default maximum relative growth rates for leaves (m2 leaf ·cm-2 sapwood· day-1), tree sapwood (cm2 sapwood· cm-1 cambium · day-1), shrub sapwood (cm2 sapwood ·cm-2 sapwood· day-1) and fine roots (g dw · g dw -1 · day -1) when not specified via SpParams (RGRleafmax, RGRcambiummax , RGRsapwoodmax and RGRfinerootmax, respectively).
- mortalityMode [= "density/deterministic"]: String describing how mortality is applied. Current accepted values are combinations of "cohort" vs "density" (for whole-cohort mortality vs reduction of stem density) and "deterministic" vs. "stochastic".
- mortalityBaselineRate [= 0.0015]: Default deterministic proportion or probability specifying the baseline reduction of cohort's density occurring in a year (for mortalityMode = "density/deterministic" or "density/stochastic").
- mortalityRelativeSugarThreshold [= 0.4]: Threshold of stem sugar concentration relative to the equilibrium sugar concentration, resulting in an increased starvation mortality rate/probability whenever levels are below.
- mortalityRWCThreshold [= 0.4]: Threshold of stem relative water content resulting in increased mortality rate/probability whenever levels are below.
- recrTreeDBH [= 1]: Default DBH (cm) for recruited trees (when species parameter RecrTreeDBH is missing).
- recrTreeDensity [= 3000]: Default density (ind·ha-1) for recruited trees (when species parameter RecrTreeDensity is missing).
- ingrowthTreeDBH [= 7.5]: Default DBH (cm) for ingrowth trees (when species parameter RecrTreeDBH is missing).
- ingrowthTreeDensity [= 127]: Default density (ind·ha-1) for ingrowth trees (when species parameter RecrTreeDensity is missing).

Forest dynamics (function fordyn):

- allowSeedBankDynamics [= TRUE]: Boolean flag to indicate that seed production and seed bank dynamics is simulated.
- allowRecruitment [= TRUE]: Boolean flag to indicate that recruitment from seeds is allowed.
- allowResprouting [= TRUE]: Boolean flag to indicate that resprouting is allowed.
- recruitmentMode [= "stochastic"]: String describing how recruitment from seeds is applied. Current accepted values are "deterministic" or "stochastic".
- removeEmptyCohorts [= TRUE]: Boolean flag to indicate the removal of cohorts whose density is too low.
- minimumTreeCohortDensity [= 1]: Threshold of tree density resulting in cohort removal.
- minimumShrubCohortCover [= 0.01]: Threshold of shrub cover resulting in cohort removal.
- dynamicallyMergeCohorts [= TRUE]: Boolean flag to indicate that cohorts should be merged when possible. This option speeds up calculations but results in a loss of cohort identity and reinitialization of many state variables.
- keepCohortsWithID [= TRUE]: Boolean flag to indicate that cohorts having a non-missing value in a column "ID" (if present) should not be merged or removed.
- seedRain [= NULL]: Vector of species names whose seed rain is to be added to seed bank, regardless of local seed production.

- seedProductionTreeHeight [= 300]: Default minimum tree height for producing seeds (when species parameter SeedProductionHeight is missing).
- seedProductionShrubHeight [= 30]: Default minimum shrub height for producing seeds (when species parameter SeedProductionHeight is missing).
- probRecr [= 0.05]: Default annual probability of seed-recruitment (when species parameter ProbRecr is missing).
- minTempRecr [= 0]: Default threshold of minimum average temperature of the coldest month necessary for recruiting from seeds (when species parameter MinTempRecr is missing).
- minMoistureRecr [= 0.3]: Default threshold of minimum moisture index (annual precipitation over annual ETP) necessary for seed-recruiting (when species parameter MinMoistureRecr is missing).
- minFPARRecr [= 10]: Default threshold of minimum fraction of PAR (in %) reaching the ground necessary for recruiting (when species parameter MinFPARRecr is missing).
- recrTreeHeight [= 620]: Default height (cm) for recruited trees (when species parameter RecrTreeHeight is missing).
- recrShrubCover [= 1]: Default cover (%) for shrubs recruited from seed (when species parameter RecrShrubCover is missing).
- recrShrubHeight [= 25]: Default height (cm) for recruited shrubs (when species parameter RecrShrubHeight is missing).
- recrTreeZ50 [= 100]: Default value for Z50 (mm) in seed-recruited trees (when species parameter RecrZ50 is missing).
- recrShrubZ50 [= 50]: Default value for Z50 (mm) in seed-recruited shrubs (when species parameter RecrZ50 is missing).
- recrTreeZ95 [= 1000]: Default value for Z95 (mm) in seed-recruited trees (when species parameter RecrZ50 is missing).
- recrShrubZ50 [= 500]: Default value for Z95 (mm) in seed-recruited shrubs (when species parameter RecrZ50 is missing).

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwbInput, spwb, growth, fordyn

---

defaultManagementFunction

Default forest management actions

Description

Function defaultManagementFunction implements actions for 'regular' and 'irregular' management models of monospecific or mixed stands, whereas function defaultManagementArguments returns a list with default values for the parameters regulating management. Both functions are meant to be used in simulations with fordyn.

Usage

```
defaultManagementFunction(x, args, verbose = FALSE)

defaultManagementArguments()
```

Arguments

x	An object of class forest
args	A list of arguments regulating management actions, e.g. the list returned by defaultManagementArguments
verbose	A logical flag enabling console printing

Details

This function implements silvicultural actions following either 'regular' or 'irregular' management models. Irregular models are implemented by executing thinning operations only, whereas regular models include both thinning and a set of final cuts. Thinning occurs anytime a stand-level metric (e.g. basal area) crosses a given threshold, and different kinds of thinning operations are allowed. Unrealistic high frequency thinning can be avoided by setting a minimum number of years to happen between thinning operations. Final cuts start whenever mean DBH exceeds a given threshold, and may include different cuts separated a number of years. The function can be applied to target management of specific taxa (instead of assuming a monospecific stand), but the thresholds that determine thinning operations apply to stand-level metrics. Mean DBH will be calculated for the target species only. Planting is only allowed under regular management models, and is applied after the last final cut. Understory clearings are assumed to occur anytime there is an intervention on trees, an only a residual shrub cover is left.

Thinning types:

- `above`: Extract largest trees (according to DBH) until thinning objective is met.
- `below`: Extract smallest trees (according to DBH) until thinning objective is met.
- `systematic`: Extract equally from all size classes until thinning objective is met.
- `above-systematic`: Extract half the objective as systematic thinning and the other hald as above thinning.
- `below-systematic`: Extract half the objective as systematic thinning and the other hald as below thinning.
- `free string`: A string specifying the proportion of tree cuts from size classes, with size classes separated by "/" and each one composed of a number specifying the upper limit and a number indicating its proportion, separated by "-" (e.g. "10-50/40-30/60-20").

Value

Function `defaultManagementFunction` returns a list with the following items:

- `"action"`: A string identifying the action performed (e.g. "thinning").
- `"N_tree_cut"`: A vector with the density of trees removed.
- `"Cover_shrub_cut"`: A vector with the cover of shrubs removed.

- "planted_forest": An object of class forest with the new plant cohorts resulting from tree/shrub planting.
- "management_args": A list of management arguments to be used in the next call to the management function.

Function defaultManagementArguments returns a list with default arguments:

- "type": Management model, either 'regular' or 'irregular'.
- "targetTreeSpecies": Either "all" for unspecific cuttings or a numeric vector of target tree species to be selected for cutting operations.
- "thinning": Kind of thinning to be applied in irregular models or in regular models before the final cuts. Options are 'below', 'above', 'systematic', 'below-systematic', 'above-systematic' or a string with the proportion of cuts to be applied to different diameter sizes (see details).
- "thinningMetric": The stand-level metric used to decide whether thinning is applied, either 'BA' (basal area), 'N' (density) or 'HB' (Hart-Becking index).
- "thinningThreshold": The threshold value of the stand-level metric causing the thinning decision.
- "thinningPerc": Percentage of stand's basal area to be removed in thinning operations.
- "minThinningInterval": Minimum number of years between thinning operations.
- "yearsSinceThinning": State variable to count the years since the last thinning ocurred.
- "finalMeanDBH": Mean DBH threshold to start final cuts.
- "finalPerc": String with percentages of basal area to be removed in final cuts, separated by '-' (e.g. "40-60-100").
- "finalPreviousStage": Integer state variable to store the stage of final cuts ('0' before starting final cuts).
- "finalYearsBetweenCuts": Number of years separating final cuts.
- "finalYearsToCut": State variable to count the years to be passed before new final cut is applied.
- "plantingSpecies": Species code to be planted. If missing, planting does not occur and only natural regeneration is allowed.
- "plantingDBH": Initial DBH (cm) of planted species.
- "plantingHeight": Initial height (cm) of planted species.
- "plantingDensity": Initial density (ind./ha) of the planted species.
- "understoryMaximumCover": Percentage of overall shrub cover to be left after any silvicultural intervention. If missing, shrub cover will not be left unmodified.

Author(s)

Miquel De Cáceres Ainsa, CREAF

Aitor Améztegui, UdL

Jose-Ramon Gonzalez Olabarria, CTFC

See Also

fordyn

Examples

```
# Load example forest object
data(exampleforest)

# Define arguments
args = defaultManagementArguments()

# Call management function
f = defaultManagementFunction(exampleforest, args)

#list names
names(f)

# Action performed
f$action

# Number of trees cut for each cohort
f$N_tree_cut

# Percent cover of shrubs removed
f$Cover_shrub_cut
```

---

defaultSoilParams *Default soil parameters*

Description

Creates a data frame with default soil physical description for model functions

Usage

```
defaultSoilParams(n = 4)
```

Arguments

n	An integer with the number of soil layers (between two and five).

Details

The function returns a data frame with default physical soil description, with soil layers in rows. Users can change those that need to be set to other values and use the list as input for function soil.

Value

A data frame with layers in rows and the following columns (and default values):

- `widths` (= `c(300,700,1000,2000)`: Width of soil layers (in mm).
- `clay` (= `25`): Clay percentage for each layer (in %).
- `sand` (= `25`): Sand percentage for each layer (in %).
- `om` (= `NA`): Organic matter percentage for each layer (in %) (optional).
- `nitrogen` (= `NA`): Sum of total nitrogen (ammonia, organic and reduced nitrogen) for each layer (in g/kg) (optional).
- `bd` (= `1.5`): Bulk density for each layer (in g/cm3).
- `rfc` (= `c(20,40,60,85)`): Percentage of rock fragment content (volume basis) for each layer.

Note

While this function is limited to five soil layers, user defined data frames can discretize soils using an unlimited number of soil layers.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`soil`, `soil_redefineLayers`, `defaultControl`, `SpParamsMED`

Examples

```
defaultSoilParams(4)
```

`droughtStress` *Drought stress indicators*

Description

Calculates plant drought stress indices, at different temporal scales, from simulation results.

Usage

```
droughtStress(x, index = "NDD", freq = "years", bySpecies = FALSE, draw = TRUE)
```

Arguments

x	An object of class spwb, pwb, growth or fordyn.
index	A string with the index to be calculated, either "DI", "NDD", "ADS", "MDS" or "WSI" (see details).
freq	Frequency of stress statistics (see cut.Date). Normally, either "years" or "months" for yearly-based or monthly-based indices.
bySpecies	Allows aggregating output by species.
draw	A boolean flag to indicate that a plot should be returned.

Details

The currently available drought stress indices are:

- "ADS": Average of daily drought stress values for the period considered.
- "MDS": Maximum daily drought stress during the period considered.
- "DI": Drought intensity, as defined in De Cáceres et al. (2015).
- "NDD": Number of drought days, as defined in De Cáceres et al. (2015).
- "WSI": Water stress integral, as defined in Myers (1988).

Value

A data frame with periods (e.g., years or months) in rows and plant cohorts (or species) in columns. Values are the calculated stress index. If `draw=TRUE` a ggplot is returned instead.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

De Cáceres M, Martínez-Vilalta J, Coll L, Llorens P, Casals P, Poyatos R, Pausas JG, Brotons L. (2015) Coupling a water balance model with forest inventory data to predict drought stress: the role of forest structural changes vs. climate changes. Agricultural and Forest Meteorology 213: 77-90 (doi:10.1016/j.agrformet.2015.06.012).

Myers BJ (1988) Water stress integral - a link between short-term stress and long-term growth. Tree Physiology 4: 315–323 (doi: 10.1093/treephys/4.4.315)

See Also

summary.spwb, waterUseEfficiency

emptyforest — *Creation of an empty forest*

Description

Creates an empty forest object.

Usage

```
emptyforest(ntree = 0, nshrub = 0, nseed = 0)
```

Arguments

ntree, nshrub	Number of tree and shrub cohorts, respectively.
nseed	Number of species in the seed bank.

Value

An empty forest object.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

forest, tree2forest, summary.forest, forest_mapWoodyTables, forest_mergeTrees, plot.forest

Examples

```
# Initializes forest with 2 tree cohorts and 1 shrub cohort
emptyforest(ntree = 2, nshrub = 1)
```

evaluation — *Evaluation of simulations results*

Description

Functions to compare model predictions against observed values.

Usage

```
evaluation_table(
  out,
  measuredData,
  type = "SWC",
  cohort = NULL,
  temporalResolution = "day"
)

evaluation_stats(
  out,
  measuredData,
  type = "SWC",
  cohort = NULL,
  temporalResolution = "day"
)

evaluation_plot(
  out,
  measuredData,
  type = "SWC",
  cohort = NULL,
  temporalResolution = "day",
  plotType = "dynamics"
)

evaluation_metric(
  out,
  measuredData,
  type = "SWC",
  cohort = NULL,
  temporalResolution = "day",
  metric = "loglikelihood"
)
```

Arguments

`out` An object of class `spwb`, `growth` or `pwb`.

`measuredData` A data frame with observed/measured values. Dates should be in row names, whereas columns should be named according to the type of output to be evaluated (see details).

`type` A string with the kind of model output to be evaluated. Accepted values are:

- `"SWC"`: Soil water content (percent volume). See details for specific soil layers.
- `"RWC"`: Relative water content (relative to field capacity). See details for specific soil layers.
- `"REW"`: Relative extractable water. See details for specific soil layers.

- "ETR": Total evapotranspiration.
- "SE+TR": Modelled soil evaporation + plant transpiration against observed total evapotranspiration
- "E": Transpiration per leaf area
- "LE": Latent heat (vaporisation) turbulent flux
- "H": Canopy sensible heat turbulent flux
- "GPP": Stand-level gross primary productivity
- "LFMC": Live fuel moisture content
- "WP": Plant water potentials
- "BAI": Basal area increment
- "DI": Diameter increment
- "DBH": Diameter at breast height
- "Height": Plant height

cohort
: A string of the cohort to be compared (e.g. "T1_68"). If NULL results for the first cohort will be evaluated.

temporalResolution
: A string to indicate the temporal resolution of the model evaluation, which can be "day", "week", "month" or "year". Observed and modelled values are aggregated temporally (using either means or sums) before comparison.

plotType
: Plot type to draw, either "dynamics", "pointdynamics" or "scatter".

metric
: An evaluation metric:

- "MAE": Mean absolute error.
- "MAE.rel": Mean absolute error in relative terms.
- "r": Pearson's linear correlation coefficient.
- "NSE": Nash-Sutcliffe model efficiency coefficient.
- "NSE.abs": Modified Nash-Sutcliffe model efficiency coefficient (L1 norm) (Legates & McCabe 1999).
- "loglikelihood": Logarithm of the likelihood of observing the data given the model predictions, assuming independent Gaussian errors.

Details

Users should provide the appropriate columns in measuredData, depending on the type of output to be evaluated:

- "SWC", "RWC", "REW": A column with the same name should be present. By default, the first soil layer is compared. Evaluation can be done for specific soil layers, for example using "RWC.2" for the relative water content of the second layer.
- "ETR" or "SE+TR": A column named "ETR" should be present, containing stand's evapotranspiration in mm/day (or mm/week, mm/month, etc, depending on the temporal resolution). If type="ETR" observed values will be compared against modelled evapotranspiration (i.e. sum of transpiration, soil evaporation and interception loss), whereas if type= "SE+TR" observed values will be compared against the sum of transpiration and soil evaporation only.
- "LE": A column named "LE" should be present containing daily latent heat turbulent flux in MJ/m2.

- "H": A column named "H" should be present containing daily sensible heat turbulent flux in MJ/m2.
- "E": For each plant cohort whose transpiration is to be evaluated, a column starting with "E_" and continuing with a cohort name (e.g. "E_T1_68") with transpiration in L/m2/day on a leaf area basis (or L/m2/week, L/m2/month, etc, depending on the temporal resolution).
- "GPP": A column named "GPP" should be present containing daily gross primary productivity in gC/m2.
- "LFMC": For each plant cohort whose transpiration is to be evaluated, a column starting with "LFMC_" and continuing with a cohort name (e.g. "LFMC_T1_68") with fuel moisture content as percent of dry weight.
- "WP": For each plant cohort whose transpiration is to be evaluated, two columns, one starting with "PD_" (for pre-dawn) and the other with "MD_" (for midday), and continuing with a cohort name (e.g. "PD_T1_68"). They should contain leaf water potential values in MPa. These are compared against sunlit water potentials.
- "BAI": For each plant cohort whose growth is to be evaluated, a column starting with "BAI_" and continuing with a cohort name (e.g. "BAI_T1_68") with basal area increment in cm2/day, cm2/week, cm2/month or cm2/year, depending on the temporal resolution.
- "DI": For each plant cohort whose growth is to be evaluated, a column starting with "DI_" and continuing with a cohort name (e.g. "DI_T1_68") with basal area increment in cm/day, cm/week, cm/month or cm/year, depending on the temporal resolution.
- "DBH": For each plant cohort whose growth is to be evaluated, a column starting with "DBH_" and continuing with a cohort name (e.g. "DBH_T1_68") with DBH values in cm.
- "Height": For each plant cohort whose growth is to be evaluated, a column starting with "Height_" and continuing with a cohort name (e.g. "Height_T1_68") with Height values in cm.

Additional columns may exist with the standard error of measured quantities. These should be named as the referred quantity, followed by "_err" (e.g. "PD_T1_68_err"), and are used to draw confidence intervals around observations.

Row names in measuredData indicate the date of measurement (in the case of days). Alternatively, a column called "dates" can contain the measurement dates. If measurements refer to months or years, row names should also be in a "year-month-day" format, although with "01" for days and/or months (e.g. "2001-02-01" for february 2001, or "2001-01-01" for year 2001).

Value

- Function evaluation_table returns a data frame with dates, observed and predicted values.
- Function evaluation_stats returns evaluation statistics (a vector or a data frame depending on type):
  - Bias: Mean deviation (positive values correspond to model overestimations).
  - Bias.rel: Bias in relative terms (%).
  - MAE: Mean absolute error.
  - MAE.rel: Mean absolute error in relative terms (%).
  - r: Pearson's linear correlation coefficient.
  - NSE: Nash-Sutcliffe model efficiency coefficient.

- NSE.abs: Modified Nash-Sutcliffe model efficiency coefficient (L1 norm) (Legates & McCabe 1999).

- Function `evaluation_plot` returns a ggplot object.
- Function `evaluation_metric` returns a scalar with the desired metric.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

Legates, D.R., McCabe, G.J., 1999. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation. Water Resour. Res. 35, 233–241.

See Also

spwb, growth, optimization, exampleobs

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x1 <- spwbInput(exampleforest, examplesoil, SpParamsMED, control)

#Call simulation function
S1 <- spwb(x1, examplemeteo, latitude = 41.82592, elevation = 100)

#Load observed data (in this case the same simulation results with some added error)
data(exampleobs)

#Evaluation statistics for soil water content
evaluation_stats(S1, exampleobs)

#NSE only
evaluation_metric(S1, exampleobs, metric="NSE")

#Comparison of temporal dynamics
```

```
evaluation_plot(S1, exampleobs)

#Loglikelihood value
evaluation_metric(S1, exampleobs)
```

---

examplemeteo *Example daily meteorology data*

Description

Example data set of meteorological input.

Format

A data frame containing daily meteorology of a location in Catalonia (Spain) for year 2001:

`dates` Vector of `Date` objects.

`MinTemperature` Minimum daily temperature (in degrees Celsius).

`MaxTemperature` Maximum daily temperature (in degrees Celsius).

`Precipitation` Daily precipitation (in mm of water).

`MinRelativeHumidity` Minimum daily relative humidity (in percent).

`MaxRelativeHumidity` Maximum daily relative humidity (in percent).

`Radiation` Incoming radiation (in MJ/m2).

`WindSpeed` Wind speed (in m/s).

Source

Interpolated from weather station data (Spanish and Catalan meteorology agencies) using package 'meteoland'.

See Also

`spwb`

Examples

```
data(examplemeteo)
```

`exampleobs` *Example observed data*

Description

Example (fake) data set of variables measured in a plot.

Format

A data frame containing daily 'observed' values for year 2001:

`dates` Measurement dates.

`SWC` Soil moisture content (in m3/m3).

`ETR` Total evapotranspiration (mm).

`E_T1_148` Transpiration of Pinus halepensis cohort 'T1_148' (L/m2 of leaf area).

`E_T2_168` Transpiration of Quercus ilex cohort 'T2_168' (L/m2 of leaf area).

`FMC_T1_148` Fuel moisture content of Pinus halepensis cohort 'T1_148' (in percent).

`FMC_T2_168` Fuel moisture content of Quercus ilex cohort 'T2_168' (in percent).

`BAI_T1_148` Basal area increment for Pinus halepensis cohort 'T1_148' (in cm2).

`BAI_T2_168` Basal area increment for Quercus ilex cohort 'T2_168' (in cm2).

`DI_T1_148` Diameter increment for Pinus halepensis cohort 'T1_148' (in cm).

`DI_T2_168` Diameter increment for Quercus ilex cohort 'T2_168' (in cm).

Source

This data set was actually created by running a simulation and adding some gaussian error to the outputs.

See Also

`evaluation`

Examples

```
data(exampleobs)
```

extract *Extracts model outputs*

Description

Function `extract()` extracts daily or subdaily output and returns it as a tidy data frame.

Usage

```
extract(
  x,
  level = "forest",
  output = NULL,
  vars = NULL,
  dates = NULL,
  subdaily = FALSE
)
```

Arguments

`x`	An object returned by simulation functions spwb, pwb or growth.
`level`	Level of simulation output, either "forest" (stand-level results), "soillayer" (soil layer-level results), "cohort" (cohort-level results), "sunlitleaf" or "shadeleaf" (leaf-level results)
`output`	Section of the model output to be explored. See details.
`vars`	Variables to be extracted (by default, all of them).
`dates`	A date vector indicating the subset of simulated days for which output is desired.
`subdaily`	A flag to indicate that subdaily values are desired (see details).

Details

When `subdaily = FALSE`, parameter `output` is used to restrict the section in `x` where variables are located. For example `output = "Plants"` will correspond to variables "LAI", "LAIlive", "Transpiration", "StemPLC",... as returned by a call `names(x$Plants)`.

Option `subdaily = TRUE` only works when simulations have been carried using control option 'subdailyResults = TRUE' (see defaultControl). When using `subdaily = TRUE`, parameter `output` is not taken into account, and options for parameter `vars` are the following:

- Variables for `level = "forest"` or `level = "soillayer"`: Not allowed. An error is raised.
- Variables for `level = "cohort"`: "E","Ag","An","dEdP","RootPsi","StemPsi","LeafPsi","StemPLC","StemRWC","Le
- Variables for `level = "shadeleaf"` and `level="sunlitleaf"`: "Abs_SWR","Abs_PAR","Net_LWR","E","Ag","An",

Value

Function `extract()` returns a data frame:

- If `level` = `"forest"`, columns are "date" and variable names.
- If `level` = `"soillayer"`, columns are "date", "soillayer" and variable names.
- If `level` = `"cohort"`, `level` = `"sunlitleaf"` or `level` = `"shadeleaf"`, columns are "date", "cohorts", "species" and variable names.
- If `subdaily` = `TRUE`, columns are "datetime", "cohorts", "species" and variable names.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`summary.spwb`

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function (ten days)
S1<-spwb(x, examplemeteo[1:10, ], latitude = 41.82592, elevation = 100)

#Extracts daily forest-level output as a data frame
extract(S1, level = "forest")

#Extracts daily soil layer-level output as a data frame
extract(S1, level = "soillayer")

#Extracts daily cohort-level output as a data frame
extract(S1, level = "cohort")

#Select the output tables/variables to be extracted
extract(S1, level ="cohort", output="Plants", vars = c("PlantStress", "StemPLC"))
```

fireHazard *Fire hazard*

Description

Estimates potential fire behaviour at each daily step of a simulation

Usage

```
fireHazard(
  x,
  SpParams,
  forest = NULL,
  standardConditions = FALSE,
  freq = "days",
  fun = "max"
)
```

Arguments

x	An object of class spwb, spwb_day, pwb, growth, growth_day or fordyn.
SpParams	A data frame with species parameters (see SpParamsDefinition and SpParamsMED).
forest	An object of class forest (needed if x is not of class fordyn).
standardConditions	A logical flag to indicate that standard fire weather conditions are to be used (instead of deriving fuel moisture and windspeed from x).
freq	Frequency of summary statistics (see cut.Date).
fun	Summary function (by default, maximum values).

Details

Live fuel moisture of shrub and canopy layers is estimated from plant water status. Dead fuel moisture is estimated following Resco-de-Dios et al. (2015).

Value

A matrix with fire behaviour variables (columns) for each simulated day (rows) or coarser time steps if summaries are requested.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

Resco de Dios, V., A. W. Fellows, R. H. Nolan, M. M. Boer, R. A. Bradstock, F. Domingo, and M. L. Goulden. 2015. A semi-mechanistic model for predicting the moisture content of fine litter. Agricultural and Forest Meteorology 203:64–73.

Ruffault J, Limousin JM, Pimont F, Dupuy JL, De Cáceres M, Cochard H, Mouillot F, Blackman C, Torres-Ruiz JM, Parsons R, Moreno M, Delzon S, Jansen S, Olioso A, Choat B, Martin-StPaul N. 2023. Plant hydraulic modelling of leaf and canopy fuel moisture content reveals increasing vulnerability of a Mediterranean forest to wildfires under extreme drought. New Phytologist. (10.1111/nph.18614).

See Also

spwb, fuel_FCCS, fire_FCCS

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x1 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S1 <- spwb(x1, examplemeteo, latitude = 41.82592, elevation = 100)

#Evaluate fire hazard
F1 <- fireHazard(S1, SpParamsMED, exampleforest)
```

---

fire_behaviour *Fire behaviour functions*

Description

Function `fire_FCCS()` implements a modification of the fire behavior models described for the Fuel Characteristics Classification System (FCCS) in Prichard et al. (2013). Function `fire_Rothermel()` implements Rothermel's (1972) fire behaviour model (modified from package 'Rothermel' (Giorgio Vacchiano, Davide Ascoli)).

Usage

```
fire_FCCS(
  FCCSpropsSI,
  MliveSI = as.numeric(c(90, 90, 60)),
  MdeadSI = as.numeric(c(6, 6, 6, 6, 6)),
  slope = 0,
  windSpeedSI = 11
)

fire_Rothermel(
  modeltype,
  wSI,
  sSI,
  delta,
  mx_dead,
  hSI,
  mSI,
  u,
  windDir,
  slope,
  aspect
)
```

Arguments

`FCCSpropsSI`	A data frame describing the properties of five fuel strata (canopy, shrub, herbs, dead woody and litter) returned by `fuel_FCCS`.
`MliveSI`	Moisture of live fuels (in percent of dry weight) for canopy, shrub, and herb strata. Live moisture values are drawn from column `ActFCM` in `FCCSpropsSI` if available (see `fuel_FCCS`). Otherwise, moisture values supplied for `MliveSI` are used.
`MdeadSI`	Moisture of dead fuels (in percent of dry weight) for canopy, shrub, herb, woody and litter strata.
`slope`	Slope (in degrees).
`windSpeedSI`	Wind speed (in m/s) at 20 ft (6 m) over vegetation (default 11 m/s = 40 km/h)
`modeltype`	'S'(tatic) or 'D'(ynamic)
`wSI`	A vector of fuel load (t/ha) for five fuel classes.
`sSI`	A vector of surface-to-volume ratio (m2/m3) for five fuel classes.
`delta`	A value of fuel bed depth (cm).
`mx_dead`	A value of dead fuel moisture of extinction (percent).
`hSI`	A vector of heat content (kJ/kg) for five fuel classes.
`mSI`	A vector of percent moisture on a dry weight basis (percent) for five fuel classes.
`u`	A value of windspeed (m/s) at midflame height.
`windDir`	Wind direction (in degrees from north). North means blowing from north to south.
`aspect`	Aspect (in degrees from north).

Details

Default moisture, slope and windspeed values are benchmark conditions used to calculate fire potentials (Sandberg et al. 2007) and map vulnerability to fire.

Value

Both functions return list with fire behavior variables.

In the case of `fire_FCCS`, the function returns the variables in three blocks (lists `SurfaceFire`, `CrownFire` and `FirePotentials`), and the values are:

- `` SurfaceFire$`midflame_WindSpeed [m/s]` ``: Midflame wind speed in the surface fire.
- `SurfaceFire$phi_wind`: Spread rate modifier due to wind.
- `SurfaceFire$phi_slope`: Spread rate modifier due to slope.
- `` SurfaceFire$`I_R_surf [kJ/m2/min]` ``: Intensity of the surface fire reaction.
- `` SurfaceFire$`I_R_litter [kJ/m2/min]` ``: Intensity of the litter fire reaction.
- `` SurfaceFire$`q_surf [kJ/m2]` ``: Heat sink of the surface fire.
- `` SurfaceFire$`q_litter [kJ/m2]` ``: Heat sink of the litter fire.
- `SurfaceFire$xi_surf`: Propagating flux ratio of the surface fire.
- `SurfaceFire$xi_litter`: Propagating flux ratio of the litter fire.
- `` SurfaceFire$`ROS_surf [m/min]` ``: Spread rate of the surface fire(without accounting for faster spread in the litter layer).
- `` SurfaceFire$`ROS_litter [m/min]` ``: Spread rate of the litter fire.
- `` SurfaceFire$`ROS_windslopecap [m/min]` ``: Maximum surface fire spread rate according to wind speed.
- `` SurfaceFire$`ROS [m/min]` ``: Final spread rate of the surface fire.
- `` SurfaceFire$`I_b [kW/m]` ``: Fireline intensity of the surface fire.
- `` SurfaceFire$`FL [m]` ``: Flame length of the surface fire.
- `` CrownFire$`I_R_canopy [kJ/m2/min]` ``: Intensity of the canopy fire reaction.
- `` CrownFire$`I_R_crown [kJ/m2/min]` ``: Intensity of the crown fire reaction (adding surface and canopy reactions).
- `` CrownFire$`q_canopy [kJ/m2]` ``: Heat sink of the canopy fire.
- `` CrownFire$`q_crown [kJ/m2]` ``: Heat sink of the crown fire (adding surface and canopy heat sinks).
- `CrownFire$xi_surf`: Propagating flux ratio of the crown fire.
- `` CrownFire$`canopy_WindSpeed [m/s]` ``: Wind speed in the canopy fire (canopy top wind speed).
- `CrownFire$WAF`: Wind speed adjustment factor for crown fires.
- `` CrownFire$`ROS [m/min]` ``: Spread rate of the crown fire.
- `CrownFire$Ic_ratio`: Crown initiation ratio.
- `` CrownFire$`I_b [kW/m]` ``: Fireline intensity of the crown fire.

- `CrownFire$`FL [m]``: Flame length of the crown fire.
- `FirePotentials$RP`: Surface fire reaction potential ([0-9]).
- `FirePotentials$SP`: Surface fire spread rate potential ([0-9]).
- `FirePotentials$FP`: Surface fire flame length potential ([0-9]).
- `FirePotentials$SFP`: Surface fire potential ([0-9]).
- `FirePotentials$IC`: Crown initiation potential ([0-9]).
- `FirePotentials$TC`: Crown-to-crown transmission potential ([0-9]).
- `FirePotentials$RC`: Crown fire spread rate potential ([0-9]).
- `FirePotentials$CFC`: Crown fire potential ([0-9]).

Note

Default moisture, slope and windspeed values are benchmark conditions used to calculate fire potentials (Sandberg et al. 2007) and map vulnerability to fire.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

Albini, F. A. (1976). Computer-based models of wildland fire behavior: A users' manual. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Rothermel, R. C. 1972. A mathematical model for predicting fire spread in wildland fuels. USDA Forest Service Research Paper INT USA.

Prichard, S. J., D. V Sandberg, R. D. Ottmar, E. Eberhardt, A. Andreu, P. Eagle, and K. Swedin. 2013. Classification System Version 3.0: Technical Documentation.

See Also

fuel_FCCS

Examples

```
#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Calculate fuel properties according to FCCS
fccs <- fuel_FCCS(exampleforest, SpParamsMED)
fccs

#Calculate fire behavior according to FCCS
fire_FCCS(fccs)
```

fordyn *Forest dynamics*

Description

Function `fordyn` implements a forest dynamics model that simulates growth, mortality, recruitment and (optionally) management actions in a given forest stand during a period specified in the input climatic data.

Usage

```
fordyn(
  forest,
  soil,
  SpParams,
  meteo,
  control,
  latitude,
  elevation = NA,
  slope = NA,
  aspect = NA,
  CO2ByYear = numeric(0),
  management_function = NULL,
  management_args = NULL
)
```

Arguments

`forest`	An object of class `forest`. Alternatively, the output of a previous run, if continuing a previous simulation.
`soil`	An object of class `data.frame` or `soil`.
`SpParams`	A data frame with species parameters (see `SpParamsMED` and `SpParamsDefinition`).
`meteo`	A data frame with daily weather data series (see `spwb`).
`control`	A list with default control parameters (see `defaultControl`).
`latitude`	Latitude (in degrees).
`elevation`, `slope`, `aspect`	Elevation above sea level (in m), slope (in degrees) and aspect (in degrees from North).
`CO2ByYear`	A named numeric vector with years as names and atmospheric CO2 concentration (in ppm) as values. Used to specify annual changes in CO2 concentration along the simulation (as an alternative to specifying daily values in `meteo`).
`management_function`	A function that implements forest management actions (see details).
`management_args`	A list of additional arguments to be passed to the `management_function`.

Details

Function fordyn simulates forest dynamics for annual time steps, building on other simulation functions. For each simulated year, the function performs the following steps:

1. Calls function growth to simulate daily water/carbon balance, growth and mortality processes and update the forest object.
2. If required, calls function management_function, using as parameters the forest object and management_args, which may result in a density reduction for existing plant cohorts and/or a set of new planted cohorts.
3. Simulate natural recruitment (for species present in the stand or given in a seed rain input).
4. Prepares the input of function growth for the next annual time step.
5. Store forest status, management arguments, and summaries.

To enable forest management, the user needs to provide a function that implements it, which is passed to fordyn via its argument management_function. Such function should have the following arguments:

- "x": the forest object representing the stand to be managed.
- "args": a list of parameters regulating the behavior of the management function.
- "verbose": a logical flag to enable console output during the execution of the management function.

and return a list with the following elements:

- "action": A string identifying the action performed (e.g. "thinning").
- "N_tree_cut": A vector with the density of trees removed.
- "Cover_shrub_cut": A vector with the cover of shrubs removed.
- "planted_forest": An object of class forest with the new plant cohorts resulting from tree/shrub planting.
- "management_args": A list of management arguments to be used in the next call to the management function.

An example of management function is provided in defaultManagementFunction.

Value

A list of class 'fordyn' with the following elements:

- "StandSummary": A data frame with stand-level summaries (tree basal area, tree density, shrub cover, etc.) at the beginning of the simulation and after each simulated year.
- "SpeciesSummary": A data frame with species-level summaries (tree basal area, tree density, shrub cover, etc.) at the beginning of the simulation and after each simulated year.
- "CohortSummary": A data frame with cohort-level summaries (tree basal area, tree density, shrub cover, etc.) at the beginning of the simulation and after each simulated year.
- "TreeTable": A data frame with tree-cohort data (species, density, diameter, height, etc.) at the beginning of the simulation (if any) and after each simulated year.

- "DeadTreeTable": A data frame with dead tree-cohort data (species, density, diameter, height, etc.) at the beginning of the simulation and after each simulated year.
- "CutTreeTable": A data frame with cut tree data (species, density, diameter, height, etc.) after each simulated year.
- "ShrubTable": A data frame with shrub-cohort data (species, density, cover, height, etc.) at the beginning of the simulation and after each simulated year.
- "DeadShrubTable": A data frame with dead shrub-cohort data (species, density, cover, height, etc.) at the beginning of the simulation (if any) and after each simulated year.
- "CutShrubTable": A data frame with cut shrub data (species, density, cover, height, etc.) after each simulated year.
- "ForestStructures": A list with the forest object of the stand at the beginning of the simulation and after each simulated year.
- "GrowthResults": A list with the results of calling function growth for each simulated year.
- "ManagementArgs": A list of management arguments to be used in another call to fordyn.
- "NextInputObject": An object of class growthInput to be used in a subsequent simulation.
- "NextForestObject": An object of class forest to be used in a subsequent simulation.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

De Cáceres M, Molowny-Horas R, Cabon A, Martínez-Vilalta J, Mencuccini M, García-Valdés R, Nadal-Sala D, Sabaté S, Martin-StPaul N, Morin X, D'Adamo F, Batllori E, Améztegui A (2023) MEDFATE 2.9.3: A trait-enabled model to simulate Mediterranean forest function and dynamics at regional scales. Geoscientific Model Development 16: 3165-3201 (https://doi.org/10.5194/gmd-16-3165-2023).

See Also

growth, regeneration, plot.growth, defaultManagementFunction

Examples

```
#Load example daily meteorological data
data(examplemeteo)
#Prepare a two-year meteorological data with half precipitation during
#the second year
meteo2001 <- examplemeteo
meteo2002 <- examplemeteo
meteo2002$Precipitation <- meteo2002$Precipitation/2
meteo2002$dates <- seq(as.Date("2002-01-01"),
                       as.Date("2002-12-31"), by="day")
meteo_01_02 <- rbind(meteo2001, meteo2002)

#Load example plot plant data
data(exampleforest)
```

```
#Default species parameterization
data(SpParamsMED)

#Initialize control parameters
control <- defaultControl("Granier")

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Call simulation function
fd<-fordyn(exampleforest, examplesoil,
           SpParamsMED, meteo_01_02, control,
           latitude = 41.82592, elevation = 100)

#Stand-level summaries
fd$StandSummary

#Tree table by annual steps
fd$TreeTable

#Dead tree table by annual steps
fd$DeadTreeTable
```

---

forest *Description of a forest stand.*

Description

exampleforest is an example of forest stand description, whereas exampleforest2 is an alternative forest description where leaf area index and crown ratio are supplied instead of structural (density, DBH and cover) parameters.

Format

An object of class forest contains the description of the woody (tree or shrub) cohorts and herb layer of a forest patch. It has the following structure (see details):

- treeData: A data frame of tree cohorts (in rows) and the following columns:
  - Species: String with species (taxon) name or a non-negative integer for tree species identity (i.e., 0,1,2,...) matching SpParams.
  - Height: Total tree height (in cm).
  - DBH: Tree diameter at breast height (in cm).
  - N: Density (number of individuals/hectare) that the measured tree represents.
  - Z50: Depth (in mm) corresponding to 50% of fine roots.
  - Z95: Depth (in mm) corresponding to 95% of fine roots.

- `shrubData`: A data frame of shrub cohorts (in rows) and the following columns:
  - `Species`: String with species (taxon) name or a non-negative integer for shrub species identity (i.e., 0,1,2,...) matching SpParams.
  - `Height`: Average total height of plants (in cm).
  - `Cover`: Percent cover.
  - `Z50`: Depth (in mm) corresponding to 50% of fine roots.
  - `Z95`: Depth (in mm) corresponding to 95% of fine roots.
- `herbCover`: Percent cover of the herb layer (optional).
- `herbHeight`: Mean height (in cm) of the herb layer (optional).
- `seedBank`: A data frame containing seed bank information with the following columns:
  - `Species`: String with species (taxon) name or a non-negative integer for tree species identity (i.e., 0,1,2,...) matching SpParams.
  - `Percent`: Amount of seeds in relation to full seed bank (in %).

Details

The structure presented above for `forest` objects corresponds to the required data elements. A `forest` object can contain additional information when this is available. Data frames `treeData` and `shrubData` can contain additional columns:

- `LAI`: Leaf area index (m2/m2).
- `FoliarBiomass`: Standing dry biomass of leaves (kg/m2).
- `FuelLoading`: Fine fuel loading (kg/m2).
- `CrownRatio`: The ratio between crown length and total height (between 0 and 1)
- `Z100`: Depth (in mm) corresponding to 100% of fine roots (to specify a truncated root distribution).
- `ObsID`: A string identifying plant cohorts at the stage of forest sampling. Used to track the fate of particular plant cohorts through simulations. In `fordyn`, the use of these identifiers can be combined with the control option `keepCohortsWithObsID` so that these cohorts are not merged or removed during simulations.

Similarly, one can define `forest` list elements `herbLAI`, `herbFoliarBiomass` or `herbFuelLoading`. All these values are used to override allometry-based estimates of those variables when initializing inputs for functions `spwb` or `spwb_day`. Note that leaf area index, foliar biomass and fuel loading are related entities, and they are treated as such in medfate. Therefore, users are expected to supply one or the other, and not all of them at the same time.

Source

DGCN (2005). Tercer Inventario Forestal Nacional (1997-2007): Catalunya. Dirección General de Conservación de la Naturaleza, Ministerio de Medio Ambiente, Madrid.

See Also

`forest`, `spwb`, `spwbInput`

`summary.forest`, `emptyforest`, `plot.forest`

Examples

```
data(exampleforest)
data(exampleforest2)
```

---

forest_mapWoodyTables *Map forest plot data*

Description

Mapping functions to facilitate building forest objects from forest plot data

Usage

```
forest_mapTreeTable(x, mapping_x, SpParams, plot_size_x = NULL)

forest_mapShrubTable(y, mapping_y, SpParams, plot_size_y = NULL)

forest_mapWoodyTables(
  x = NULL,
  y = NULL,
  mapping_x = NULL,
  mapping_y = NULL,
  SpParams,
  plot_size_x = NULL,
  plot_size_y = NULL
)
```

Arguments

x	A data frame with tree records in rows and attributes in columns. Tree records can correspond to individual trees or groups of trees with an associated density.
mapping_x	A named character vector to specify mappings of columns in x into attributes of treeData data frames. Accepted names (and the corresponding specifications for the columns in x) are:
SpParams	A data frame with species parameters (see SpParamsMED) from which valid species names are drawn.
plot_size_x	The size of tree plot sampled area (in m2). Alternatively, 'plot_size_x' can be a column in x and specified in mapping_x to indicate that trees have been measured in different subplots and, therefore, they represent different densities per hectare.
y	A data frame with shrub records in rows and attributes in columns. Records can correspond to individual shrubs (with crown dimensions and height) or groups of shrubs with an associated cover estimate.
mapping_y	A named character vector to specify mappings of columns in y into attributes of shrubData data frames. Accepted names (and the corresponding specifications for the columns in y) are:

- `Species`: Species code (should follow codes in `SpParams`).
- `Species.name`: Species name. In this case, the species code will be drawn by matching names with species names in `SpParams`.
- `N`: Tree density (in individuals per ha).
- `Cover`: Shrub cover (in percent).
- `D1`: Shrub largest crown diameter (in cm).
- `D2`: Shrub crown diameter orthogonal to the largest one (in cm).
- `plot.size`: Plot size (in m2) to which each record refers to. This is used to calculate tree density (stems per hectare) when not supplied or shrub cover when shrub data is given at the individual level.
- `DBH`: Diameter at breast height (in cm).
- `Height`: Tree or shrub height (in cm).
- `Z50`: Depth (in mm) corresponding to 50 percent of fine roots.
- `Z95`: Depth (in mm) corresponding to 95 percent of fine roots.

`plot_size_y` The size of shrub plot sampled area (in m2). Alternatively, 'plot_size_y' can be a column in `y` and specified in `mapping_y` to indicate that shrubs have been measured in different subplots and, therefore, they represent different cover values.

Value

Functions `forest_mapTreeTable` and `forest_mapShrubTable` return a data frame with the structure of `treeData` and `shrubData` from `forest` objects. Function `forest_mapWoodyTable` returns directly a `forest` object.

Author(s)

Miquel De Cáceres Ainsa, EMF-CREAF

See Also

`forest`, `poblet_trees`, `forest_mergeTrees`, `tree2forest`

Examples

```
# Load species parameters
data(SpParamsMED)

# Create an empty forest object
f <- emptyforest()

# (1) Mapping tree data
# Load Poblet tree data
data(poblet_trees)

# Subset control plot
x <- subset(poblet_trees, Plot.Code=="POBL_CTL")

# Estimate sampled area (15-m radius plot)
sampled_area <- pi*15^2
```

```
# Define mapping
mapping_x <- c("Species.name" = "Species", "DBH" = "Diameter.cm")

# Map tree data for plot 'POBL_CTL'
f$treeData <- forest_mapTreeTable(x,
                    mapping_x = mapping_x, SpParams = SpParamsMED,
                    plot_size_x = sampled_area)

# (2) Mapping shrub individual data
#
# Create the individual shrub data frame
species <- c("Erica arborea","Cistus albidus", "Erica arborea", "Cistus albidus", "Cistus albidus")
H <- c(200,50,100,40,30)
D1 <- c(140,40,100, 35,30)
D2 <- D1
y <- data.frame(species, H, D1, D2)

# Define mapping (D1 and D2 map to variables with the same name)
mapping_y <- c("Species.name"= "species", "Height" ="H", "D1", "D2")

# Map individual shrub data to cover data (here each individual becomes a cohort)
# assuming that the sampled area was 4 m2
f$shrubData <- forest_mapShrubTable(y,
                     mapping_y = mapping_y, SpParams = SpParamsMED,
                     plot_size_y = 4)

# (3) Print forest attributes
summary(f, SpParamsMED)

# (4) Forest initialization in a single step
f <- forest_mapWoodyTables(x, y,
                           mapping_x = mapping_x, mapping_y = mapping_y,
                           SpParams = SpParamsMED,
                           plot_size_x = sampled_area, plot_size_y = 4)
summary(f, SpParamsMED)
```

forest_simplification *Forest complexity reduction*

Description

Functions forest_mergeTrees and forest_mergeShrubs merge cohorts of a forest object. Function forest_reduceToDominant performs a strongest simplification of plant cohorts (see details).

Usage

```
forest_mergeTrees(x, byDBHclass = TRUE, keepCohortsWithObsID = FALSE)

forest_mergeShrubs(x, byHeightclass = TRUE, keepCohortsWithObsID = FALSE)
```

```
forest_reduceToDominant(x, SpParams)
```

Arguments

x	An object of class forest.
byDBHclass	Logical flag to indicate that 5-cm tree DBH classes should be kept separated.
keepCohortsWithObsID	Logical flag to indicate that cohorts with non-missin ObsID should be spared from merging.
byHeightclass	Boolean flag to indicate that 10-cm shrub height classes should be kept separated.
SpParams	A data frame with species parameters (see SpParamsDefinition and SpParamsMED).

Details

Tree DBH classes are defined in 5-cm intervals, whereas shrub height classes are defined in 10-cm intervals. Tree DBH and shrub height classes are defined up to a specific size (i.e. larger plants are not merged) corresponding to 52.5 cm and 90 cm, respectively.

Function `forest_reduceToDominant` simplifies the input forest to the tree cohort of highest LAI, among those of the tree species with highest LAI. The leaf area index of the whole tree layer will be attributed to the chosen cohort. The same is performed for the shrub layer.

Value

Another `forest` object with simplified structure/composition, depending on the function.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`spwb`, `forest`, `forest_mapWoodyTables`, `fordyn`, `summary.forest`

Examples

```
# Example forest data
data("exampleforest")

# Reduce to dominant tree and dominant shrub
reduced <- forest_reduceToDominant(exampleforest, SpParamsMED)

# Check that overall LAI does not change
stand_LAI(exampleforest, SpParamsMED)
stand_LAI(reduced, SpParamsMED)
```

fuel_properties *Fuel stratification and fuel characteristics*

Description

Function `fuel_stratification` provides a stratification of the stand into understory and canopy strata. Function `fuel_FCCS` calculates fuel characteristics from a `forest` object following an adaptation of the protocols described for the Fuel Characteristics Classification System (Prichard et al. 2013).

Usage

```
fuel_stratification(
  object,
  SpParams,
  gdd = NA_real_,
  heightProfileStep = 10,
  maxHeightProfile = 5000,
  bulkDensityThreshold = 0.05
)

fuel_FCCS(
  object,
  SpParams,
  cohortFMC = as.numeric(c()),
  loadingOffset = as.numeric(c(0, 0, 0, 0, 0)),
  gdd = NA_real_,
  heightProfileStep = 10,
  maxHeightProfile = 5000,
  bulkDensityThreshold = 0.05,
  depthMode = "crownaverage"
)
```

Arguments

`object`	An object of class `forest`
`SpParams`	A data frame with species parameters (see `SpParamsMED`).
`gdd`	Growth degree-days.
`heightProfileStep`	Precision for the fuel bulk density profile.
`maxHeightProfile`	Maximum height for the fuel bulk density profile.
`bulkDensityThreshold`	Minimum fuel bulk density to delimit fuel strata.
`cohortFMC`	A numeric vector of (actual) fuel moisture content by cohort.

loadingOffset A vector of length five with fine fuel loading values (canopy, shrub, herb, woody and litter) to be added to loading estimations from forest.

depthMode Specifies how fuel depth (and therefore canopy and understory bulk density) should be estimated:

- "crownaverage": As weighed average of crown lengths using loadings as weights.
- "profile": As the difference of base and top heights in bulk density profiles.
- "absoluteprofile": As the difference of absolute base and absolute top heights in bulk density profiles.

Value

Function fuel_FCCS returns a data frame with five rows corresponding to fuel layers: canopy, shrub, herb, woody and litter. Columns correspond fuel properties:

- w: Fine fuel loading (in kg/m2).
- cover: Percent cover.
- hbc: Height to base of crowns (in m).
- htc: Height to top of crowns (in m).
- delta: Fuel depth (in m).
- rhob: Fuel bulk density (in kg/m3).
- rhop: Fuel particle density (in kg/m3).
- PV: Particle volume (in m3/m2).
- beta: Packing ratio (unitless).
- betarel: Relative packing ratio (unitless).
- etabetarel: Reaction efficiency (unitless).
- sigma: Surface area-to-volume ratio (m2/m3).
- pDead: Proportion of dead fuels.
- FAI: Fuel area index (unitless).
- h: High heat content (in kJ/kg).
- RV: Reactive volume (in m3/m2).
- MinFMC: Minimum fuel moisture content (as percent over dry weight).
- MaxFMC: Maximum fuel moisture content (as percent over dry weight).
- ActFMC: Actual fuel moisture content (as percent over dry weight). These are set to NA if parameter cohortFMC is empty.

Function fuel_stratification returns a list with the following items:

- surfaceLayerBaseHeight: Base height of crowns of shrubs in the surface layer (in cm).
- surfaceLayerTopHeight: Top height of crowns of shrubs in the surface layer (in cm).
- understoryLAI: Cumulated LAI of the understory layer (i.e. leaf area comprised between surface layer base and top heights).

- `canopyBaseHeight`: Base height of tree crowns in the canopy (in cm).
- `canopyTopHeight`: Top height of tree crowns in the canopy (in cm).
- `canopyLAI`: Cumulated LAI of the canopy (i.e. leaf area comprised between canopy base and top heights).

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

Prichard, S. J., D. V Sandberg, R. D. Ottmar, E. Eberhardt, A. Andreu, P. Eagle, and K. Swedin. 2013. Classification System Version 3.0: Technical Documentation.

Reinhardt, E., D. Lutes, and J. Scott. 2006. FuelCalc: A method for estimating fuel characteristics. Pages 273–282.

See Also

`fire_FCCS`, `spwb`

Examples

```
#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Show stratification of fuels
fuel_stratification(exampleforest, SpParamsMED)

#Calculate fuel properties according to FCCS
fccs <- fuel_FCCS(exampleforest, SpParamsMED)
fccs
```

`growth` *Forest growth*

Description

Function `growth` is a process-based model that performs energy, water and carbon balances; and determines changes in water/carbon pools, functional variables (leaf area, sapwood area, root area) and structural ones (tree diameter, tree height, shrub cover) for woody plant cohorts in a given forest stand during a period specified in the input climatic data.

Usage

```
growth(
  x,
  meteo,
  latitude,
  elevation,
  slope = NA_real_,
  aspect = NA_real_,
  CO2ByYear = numeric(0),
  waterTableDepth = NA_real_
)
```

Arguments

`x`	An object of class `growthInput`.
`meteo`	A data frame with daily meteorological data series (see `spwb`).
`latitude`	Latitude (in degrees).
`elevation, slope, aspect`	Elevation above sea level (in m), slope (in degrees) and aspect (in degrees from North).
`CO2ByYear`	A named numeric vector with years as names and atmospheric CO2 concentration (in ppm) as values. Used to specify annual changes in CO2 concentration along the simulation (as an alternative to specifying daily values in `meteo`).
`waterTableDepth`	Water table depth (in mm). When not missing, capillarity rise will be allowed if lower than total soil depth.

Details

Detailed model description is available in the medfate book. Simulations using the 'Sperry' or 'Sureau' transpiration modes are computationally much more expensive than those using the 'Granier' transpiration mode.

Value

A list of class 'growth' with the following elements:

- `"latitude"`: Latitude (in degrees) given as input.
- `"topography"`: Vector with elevation, slope and aspect given as input.
- `"weather"`: A copy of the input weather data frame.
- `"growthInput"`: A copy of the object `x` of class `growthInput` given as input.
- `"growthOutput"`: An copy of the final state of the object `x` of class `growthInput`.
- `"WaterBalance"`: A data frame where different water balance variables (see `spwb`).
- `"EnergyBalance"`: A data frame with the daily values of energy balance components for the soil and the canopy (only for `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"`; see `spwb`).

- "CarbonBalance": A data frame where different stand-level carbon balance components (gross primary production, maintenance respiration, synthesis respiration and net primary production), all in g C · m-2.
- "BiomassBalance": A data frame with the daily values of stand biomass balance components (in g dry · m-2.
- "Temperature": A data frame with the daily values of minimum/mean/maximum temperatures for the atmosphere (input), canopy and soil (only for transpirationMode = "Sperry" or transpirationMode = "Sureau"; see spwb).
- "Soil": A data frame where different soil variables (see spwb).
- "Stand": A data frame where different stand-level variables (see spwb).
- "Plants": A list of daily results for plant cohorts (see spwb).
- "SunlitLeaves" and "ShadeLeaves": A list with daily results for sunlit and shade leaves (only for transpirationMode = "Sperry" or transpirationMode = "Sureau"; see spwb).
- "LabileCarbonBalance": A list of daily labile carbon balance results for plant cohorts, with elements:
  - "GrossPhotosynthesis": Daily gross photosynthesis per dry weight of living biomass (g gluc · g dry-1).
  - "MaintentanceRespiration": Daily maintenance respiration per dry weight of living biomass (g gluc · g dry-1).
  - "GrowthCosts": Daily growth costs per dry weight of living biomass (g gluc · g dry-1).
  - "RootExudation": Root exudation per dry weight of living biomass (g gluc · g dry-1).
  - "LabileCarbonBalance": Daily labile carbon balance (photosynthesis - maintenance respiration - growth costs - root exudation) per dry weight of living biomass (g gluc · g dry-1).
  - "SugarLeaf": Sugar concentration (mol·l-1) in leaves.
  - "StarchLeaf": Starch concentration (mol·l-1) in leaves.
  - "SugarSapwood": Sugar concentration (mol·l-1) in sapwood.
  - "StarchSapwood": Starch concentration (mol·l-1) in sapwood.
  - "SugarTransport": Average instantaneous rate of carbon transferred between leaves and stem compartments via floem (mol gluc·s-1).
- "PlantBiomassBalance": A list of daily plant biomass balance results for plant cohorts, with elements:
  - "StructuralBiomassBalance": Daily structural biomass balance (g dry · m-2).
  - "LabileBiomassBalance": Daily labile biomass balance (g dry · m-2).
  - "PlantBiomassBalance": Daily plant biomass balance, i.e. labile change + structural change (g dry · m-2).
  - "MortalityBiomassLoss": Biomass loss due to mortality (g dry · m-2).
  - "CohortBiomassBalance": Daily cohort biomass balance (including mortality) (g dry · m-2).
- "PlantStructure": A list of daily area and biomass values for compartments of plant cohorts, with elements:
  - "LeafBiomass": Daily amount of leaf structural biomass (in g dry) for an average individual of each plant cohort.

  - "SapwoodBiomass": Daily amount of sapwood structural biomass (in g dry) for an average individual of each plant cohort.
  - "FineRootBiomass": Daily amount of fine root biomass (in g dry) for an average individual of each plant cohort.
  - "LeafArea": Daily amount of leaf area (in m2) for an average individual of each plant cohort.
  - "SapwoodArea": Daily amount of sapwood area (in cm2) for an average individual of each plant cohort.
  - "FineRootArea": Daily amount of fine root area (in m2) for an average individual of each plant cohort.
  - "HuberValue": The ratio of sapwood area to (target) leaf area (in cm2/m2).
  - "RootAreaLeafArea": The ratio of fine root area to (target) leaf area (in m2/m2).
  - "DBH": Diameter at breast height (in cm) for an average individual of each plant cohort.
  - "Height": Height (in cm) for an average individual of each plant cohort.
- "GrowthMortality": A list of daily growth and mortality rates for plant cohorts, with elements:
  - "LAgrowth": Leaf area growth (in m2·day-1) for an average individual of each plant cohort.
  - "SAgrowth": Sapwood area growth rate (in cm2·day-1) for an average individual of each plant cohort.
  - "FRAgrowth": Fine root area growth (in m2·day-1) for an average individual of each plant cohort.
  - "StarvationRate": Daily mortality rate from starvation (ind/d-1).
  - "DessicationRate": Daily mortality rate from dessication (ind/d-1).
  - "MortalityRate": Daily mortality rate (any cause) (ind/d-1).
- "subdaily": A list of objects of class growth_day, one per day simulated (only if required in control parameters, see defaultControl).

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

De Cáceres M, Molowny-Horas R, Cabon A, Martínez-Vilalta J, Mencuccini M, García-Valdés R, Nadal-Sala D, Sabaté S, Martin-StPaul N, Morin X, D'Adamo F, Batllori E, Améztegui A (2023) MEDFATE 2.9.3: A trait-enabled model to simulate Mediterranean forest function and dynamics at regional scales. Geoscientific Model Development 16: 3165-3201 (https://doi.org/10.5194/gmd-16-3165-2023).

See Also

growthInput, growth_day, plot.growth

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize model input
x1 <- growthInput(exampleforest, examplesoil, SpParamsMED, control)

#Call simulation function
G1 <- growth(x1, examplemeteo, latitude = 41.82592, elevation = 100)

#Switch to 'Sperry' transpiration mode
control <- defaultControl("Sperry")

#Initialize model input
x2 <- growthInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
G2 <-growth(x2, examplemeteo, latitude = 41.82592, elevation = 100)

#Switch to 'Sureau' transpiration mode
control <- defaultControl("Sureau")

#Initialize model input
x3 <- growthInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
G3 <-growth(x3, examplemeteo, latitude = 41.82592, elevation = 100)
```

---

growth_day *Single-day simulation*

Description

Function spwb_day performs water balance for a single day and growth_day performs water and carbon balance for a single day.

Usage

```
growth_day(
  x,
  date,
  meteovec,
  latitude,
  elevation,
  slope = NA_real_,
  aspect = NA_real_,
  runon = 0,
  lateralFlows = NULL,
  waterTableDepth = NA_real_,
  modifyInput = TRUE
)

spwb_day(
  x,
  date,
  meteovec,
  latitude,
  elevation,
  slope = NA_real_,
  aspect = NA_real_,
  runon = 0,
  lateralFlows = NULL,
  waterTableDepth = NA_real_,
  modifyInput = TRUE
)
```

Arguments

`x`	An object of class `spwbInput` or `growthInput`.
`date`	Date as string "yyyy-mm-dd".
`meteovec`	A named numerical vector with weather data. See variable names in parameter `meteo` of `spwb`.
`latitude`	Latitude (in degrees).
`elevation, slope, aspect`	Elevation above sea level (in m), slope (in degrees) and aspect (in degrees from North).
`runon`	Surface water amount running on the target area from upslope (in mm).
`lateralFlows`	Lateral source/sink terms for each soil layer (interflow/to from adjacent locations) as mm/day.
`waterTableDepth`	Water table depth (in mm). When not missing, capillarity rise will be allowed if lower than total soil depth.
`modifyInput`	Boolean flag to indicate that the input `x` object is allowed to be modified during the simulation.

Details

The simulation functions allow using three different sub-models of transpiration and photosynthesis:

- The sub-model corresponding to 'Granier' transpiration mode is illustrated by function `transp_transpirationGranie` and was described in De Caceres et al. (2015), and implements an approach originally described in Granier et al. (1999).
- The sub-model corresponding to 'Sperry' transpiration mode is illustrated by function `transp_transpirationSperry` and was described in De Caceres et al. (2021), and implements a modelling approach originally described in Sperry et al. (2017).
- The sub-model corresponding to 'Sureau' transpiration mode is illustrated by function `transp_transpirationSureau` and was described for model SurEau-Ecos v2.0 in Ruffault et al. (2022).

Simulations using the 'Sperry' or 'Sureau' transpiration mode are computationally much more expensive than 'Granier'.

Value

Function `spwb_day()` returns a list of class `spwb_day` with the following elements:

- `"cohorts"`: A data frame with cohort information, copied from `spwbInput`.
- `"topography"`: Vector with elevation, slope and aspect given as input.
- `"weather"`: A vector with the input weather.
- `"WaterBalance"`: A vector of water balance components (rain, snow, net rain, infiltration, ...) for the simulated day, equivalent to one row of 'WaterBalance' object given in `spwb`.
- `"Soil"`: A data frame with results for each soil layer:
  - `"Psi"`: Soil water potential (in MPa) at the end of the day.
  - `"HerbTranspiration"`: Water extracted by herbaceous plants from each soil layer (in mm).
  - `"HydraulicInput"`: Water entering each soil layer from other layers, transported via plant roots (in mm).
  - `"HydraulicOutput"`: Water leaving each soil layer (going to other layers or the transpiration stream) (in mm).
  - `"PlantExtraction"`: Water extracted by woody plants from each soil layer (in mm).
- `"Stand"`: A named vector with with stand values for the simulated day, equivalent to one row of 'Stand' object returned by `spwb`.
- `"Plants"`: A data frame of results for each plant cohort (see `transp_transpirationGranier` or `transp_transpirationSperry`).

The following items are only returned when `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"`:

- `"EnergyBalance"`: Energy balance of the stand (see `transp_transpirationSperry`).
- `"RhizoPsi"`: Minimum water potential (in MPa) inside roots, after crossing rhizosphere, per cohort and soil layer.

- "SunlitLeaves" and "ShadeLeaves": For each leaf type, a data frame with values of LAI, Vmax298 and Jmax298 for leaves of this type in each plant cohort.
- "ExtractionInst": Water extracted by each plant cohort during each time step.
- "PlantsInst": A list with instantaneous (per time step) results for each plant cohort (see transp_transpirationSperry).
- "LightExtinction": A list of information regarding radiation balance through the canopy, as returned by function light_instantaneousLightExtinctionAbsortion.
- "CanopyTurbulence": Canopy turbulence (see wind_canopyTurbulence).

Author(s)

- Miquel De Cáceres Ainsa, CREAF
- Nicolas Martin-StPaul, URFM-INRAE

References

De Cáceres M, Martínez-Vilalta J, Coll L, Llorens P, Casals P, Poyatos R, Pausas JG, Brotons L. (2015) Coupling a water balance model with forest inventory data to predict drought stress: the role of forest structural changes vs. climate changes. Agricultural and Forest Meteorology 213: 77-90 (doi:10.1016/j.agrformet.2015.06.012).

De Cáceres M, Mencuccini M, Martin-StPaul N, Limousin JM, Coll L, Poyatos R, Cabon A, Granda V, Forner A, Valladares F, Martínez-Vilalta J (2021) Unravelling the effect of species mixing on water use and drought stress in holm oak forests: a modelling approach. Agricultural and Forest Meteorology 296 (doi:10.1016/j.agrformet.2020.108233).

Granier A, Bréda N, Biron P, Villette S (1999) A lumped water balance model to evaluate duration and intensity of drought constraints in forest stands. Ecol Modell 116:269–283. https://doi.org/10.1016/S0304-3800(98)00205-1.

Ruffault J, Pimont F, Cochard H, Dupuy JL, Martin-StPaul N (2022) SurEau-Ecos v2.0: a trait-based plant hydraulics model for simulations of plant water status and drought-induced mortality at the ecosystem level. Geoscientific Model Development 15, 5593-5626 (doi:10.5194/gmd-15-5593-2022).

Sperry, J. S., M. D. Venturas, W. R. L. Anderegg, M. Mencuccini, D. S. Mackay, Y. Wang, and D. M. Love. 2017. Predicting stomatal responses to the environment from the optimization of photosynthetic gain and hydraulic cost. Plant Cell and Environment 40, 816-830 (doi: 10.1111/pce.12852).

See Also

spwbInput, spwb, plot.spwb_day, growthInput, growth, plot.growth_day

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)
```

```
#Default species parameterization
data(SpParamsMED)

#Define soil parameters
examplesoil <- defaultSoilParams(4)

# Day to be simulated
d <- 100
meteovec <- unlist(examplemeteo[d,-1])
date <- as.character(examplemeteo$dates[d])

#Simulate water balance one day only (Granier mode)
control <- defaultControl("Granier")
x1 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)
sd1 <- spwb_day(x1, date, meteovec,
                latitude = 41.82592, elevation = 100, slope=0, aspect=0)

#Simulate water balance for one day only (Sperry mode)
control <- defaultControl("Sperry")
x2 <- spwbInput(exampleforest, examplesoil, SpParamsMED, control)
sd2 <-spwb_day(x2, date, meteovec,
              latitude = 41.82592, elevation = 100, slope=0, aspect=0)

#Simulate water balance for one day only (Sureau mode)
control <- defaultControl("Sureau")
x3 <- spwbInput(exampleforest, examplesoil, SpParamsMED, control)
sd3 <-spwb_day(x3, date, meteovec,
              latitude = 41.82592, elevation = 100, slope=0, aspect=0)


#Simulate water and carbon balance for one day only (Granier mode)
control <- defaultControl("Granier")
x4  <- growthInput(exampleforest,examplesoil, SpParamsMED, control)
sd4 <- growth_day(x4, date, meteovec,
                latitude = 41.82592, elevation = 100, slope=0, aspect=0)

#Simulate water and carbon balance for one day only (Sperry mode)
control <- defaultControl("Sperry")
x5  <- growthInput(exampleforest,examplesoil, SpParamsMED, control)
sd5 <- growth_day(x5, date, meteovec,
                latitude = 41.82592, elevation = 100, slope=0, aspect=0)

#Simulate water and carbon balance for one day only (Sureau mode)
control <- defaultControl("Sureau")
x6  <- growthInput(exampleforest,examplesoil, SpParamsMED, control)
sd6 <- growth_day(x6, date, meteovec,
                latitude = 41.82592, elevation = 100, slope=0, aspect=0)
```

---

modelInput — *Input for simulation models*

Description

Functions spwbInput() and growthInput() take an object of class forest and a soil data input to create input objects for simulation functions spwb (or pwb) and growth, respectively.

Usage

```
spwbInput(x, soil, SpParams, control)

growthInput(x, soil, SpParams, control)
```

Arguments

x	An object of class forest.
soil	An object of class data.frame or soil, containing soil parameters per soil layer.
SpParams	A data frame with species parameters (see SpParamsDefinition and SpParamsMED).
control	A list with default control parameters (see defaultControl).

Details

Functions spwbInput() and growthInput() initialize inputs differently depending on control parameters.

IMPORTANT NOTE: Older function names forest2spwbInput and forest2growthInput are now deprecated, but they can still be used for back-compatibility.

Value

Function spwbInput() returns a list of class spwbInput with the following elements (rows of data frames are identified as specified by function plant_ID):

- control: List with control parameters (see defaultControl).
- soil: A data frame with initialized soil parameters (see soil).
- snowpack: The amount of snow (in mm) in the snow pack over the soil.
- canopy: A list of stand-level state variables.
- cohorts: A data frame with cohort information, with columns SP and Name.
- above: A data frame with columns H, CR and LAI (see function forest2aboveground).
- below: A data frame with columns Z50, Z95. If control$transpirationMode = "Sperry" additional columns are fineRootBiomass and coarseRootSoilVolume.
- belowLayers: A list. If control$transpirationMode = "Granier" it contains elements:
  - V: A matrix with the proportion of fine roots of each cohort (in rows) in each soil layer (in columns).
  - L: A matrix with the length of coarse roots of each cohort (in rows) in each soil layer (in columns).
  - Wpool: A matrix with the soil moisture relative to field capacity around the rhizosphere of each cohort (in rows) in each soil layer (in columns).

If control$transpirationMode = "Sperry" or control$transpirationMode = "Sureau" there are the following additional elements:

- VGrhizo_kmax: A matrix with maximum rhizosphere conductance values of each cohort (in rows) in each soil layer (in columns).
- VGroot_kmax: A matrix with maximum root xylem conductance values of each cohort (in rows) in each soil layer (in columns).
- RhizoPsi: A matrix with the water potential around the rhizosphere of each cohort (in rows) in each soil layer (in columns).

- paramsPhenology: A data frame with leaf phenology parameters:
  - PhenologyType: Leaf phenology type.
  - LeafDuration: Leaf duration (in years).
  - Sgdd: Degree days needed for leaf budburst (for winter decideous species).
  - Tbgdd: Base temperature for the calculation of degree days to leaf budburst.
  - Ssen: Degree days corresponding to leaf senescence.
  - Phsen: Photoperiod corresponding to start counting senescence degree-days.
  - Tbsen: Base temperature for the calculation of degree days to leaf senescence.
- paramsAnatomy: A data frame with plant anatomy parameters for each cohort:
  - Hmax: Maximum plant height (cm).
  - Hmed: Median plant height (cm).
  - Al2As: Leaf area to sapwood area ratio (in m2·m-2).
  - Ar2Al: Fine root area to leaf area ratio (in m2·m-2).
  - SLA: Specific leaf area (mm2/mg = m2/kg).
  - LeafWidth: Leaf width (in cm).
  - LeafDensity: Density of leaf tissue (dry weight over volume).
  - WoodDensity: Density of wood tissue (dry weight over volume).
  - FineRootDensity: Density of fine root tissue (dry weight over volume).
  - SRL: Specific Root length (cm·g-1).
  - RLD: Root length density (cm·cm-3).
  - r635: Ratio between the weight of leaves plus branches and the weight of leaves alone for branches of 6.35 mm.
- paramsInterception: A data frame with rain interception and light extinction parameters for each cohort:
  - kPAR: PAR extinction coefficient.
  - g: Canopy water retention capacity per LAI unit (mm/LAI).

  If control$transpirationMode = "Sperry" or control$transpirationMode = "Sureau" additional columns are:

  - gammaSWR: Reflectance (albedo) coefficient for SWR .
  - alphaSWR: Absorbance coefficient for SWR .
- paramsTranspiration: A data frame with parameters for transpiration and photosynthesis. If control$transpirationMode = "Granier", columns are:
  - Gswmin: Minimum stomatal conductance to water vapor (in mol H2O·m-2·s-1).

- Tmax_LAI: Coefficient relating LAI with the ratio of maximum transpiration over potential evapotranspiration.
- Tmax_LAIsq: Coefficient relating squared LAI with the ratio of maximum transpiration over potential evapotranspiration.
- Psi_Extract: Water potential corresponding to 50% relative transpiration (in MPa).
- Exp_Extract: Parameter of the Weibull function regulating transpiration reduction.
- VCstem_c, VCstem_d: Parameters of the stem xylem vulnerability curve (Weibull).
- WUE: Daily water use efficiency (gross photosynthesis over transpiration) under no light, water or CO2 limitations and VPD = 1kPa (g C/mm water).
- WUE_par: Coefficient regulating the influence of % PAR on gross photosynthesis.
- WUE_co2: Coefficient regulating the influence of atmospheric CO2 concentration on gross photosynthesis.
- WUE_vpd: Coefficient regulating the influence of vapor pressure deficit (VPD) on gross photosynthesis.

If control$transpirationMode = "Sperry" columns are:

- Gswmin: Minimum stomatal conductance to water vapor (in mol H2O·m-2·s-1).
- Gswmax: Maximum stomatal conductance to water vapor (in mol H2O·m-2·s-1).
- Vmax298: Maximum Rubisco carboxilation rate at 25ºC (in micromol CO2·s-1·m-2).
- Jmax298: Maximum rate of electron transport at 25ºC (in micromol photons·s-1·m-2).
- Kmax_stemxylem: Sapwood-specific hydraulic conductivity of stem xylem (in kg H2O·s-1·m-1·MPa-1).
- Kmax_rootxylem: Sapwood-specific hydraulic conductivity of root xylem (in kg H2O·s-1·m-1·MPa-1).
- VCleaf_kmax: Maximum leaf hydraulic conductance (in mmol H2O·s-1·m-2·MPa-1).
- VCleaf_c, VCleaf_d: Parameters of the leaf vulnerability curve (Weibull).
- VCstem_kmax: Maximum stem xylem conductance (in mmol H2O·s-1·m-2·MPa-1).
- VCstem_c, VCstem_d: Parameters of the stem xylem vulnerability curve (Weibull).
- VCroot_c, VCroot_d: Parameters of the root xylem vulnerability curve (Weibull).
- Plant_kmax: Maximum whole-plant conductance (in mmol H2O·s-1·m-2·MPa-1).
- FR_leaf, FR_stem, FR_root: Fraction of whole-plant resistance corresponding to each segment.

If control$transpirationMode = "Sureau" columns are:

- Gswmin: Minimum stomatal conductance to water vapor (in mol H2O·m-2·s-1).
- Gswmax: Maximum stomatal conductance to water vapor (in mol H2O·m-2·s-1).
- Gsw_AC_slope: Slope of the Gsw vs Ac/Cs relationship (see photo_photosynthesisBaldocchi).
- Gs_P50, Gs_slope: Parameters of the curve describing the decrease in stomatal conductance as a function of leaf water potential (sigmoid).
- Vmax298: Maximum Rubisco carboxylation rate at 25ºC (in micromol CO2·s-1·m-2).
- Jmax298: Maximum rate of electron transport at 25ºC (in micromol photons·s-1·m-2).
- Kmax_stemxylem: Sapwood-specific hydraulic conductivity of stem xylem (in kg H2O·s-1·m-1·MPa-1).
- Kmax_rootxylem: Sapwood-specific hydraulic conductivity of root xylem (in kg H2O·s-1·m-1·MPa-1).

- VCleaf_kmax: Maximum leaf hydraulic conductance (in mmol H2O·s-1·m-2·MPa-1).
- VCleaf_c, VCleaf_d: Parameters of the leaf vulnerability curve (Weibull).
- VCleaf_P50, VCleaf_slope: Parameters of the leaf vulnerability curve (sigmoid).
- VCstem_kmax: Maximum stem xylem conductance (in mmol H2O·s-1·m-2·MPa-1).
- VCstem_c, VCstem_d: Parameters of the stem xylem vulnerability curve (Weibull).
- VCstem_P50, VCstem_slope: Parameters of the stem xylem vulnerability curve (sigmoid).
- VCroot_c, VCroot_d: Parameters of the root xylem vulnerability curve (Weibull).
- VCroot_P50, VCroot_slope: Parameters of the root xylem vulnerability curve (sigmoid).
- Plant_kmax: Maximum whole-plant conductance (in mmol H2O·s-1·m-2·MPa-1).
- FR_leaf, FR_stem, FR_root: Fraction of whole-plant resistance corresponding to each segment.

• paramsWaterStorage: A data frame with plant water storage parameters for each cohort:
  - LeafPI0: Osmotic potential at full turgor of leaves (MPa).
  - LeafEPS: Modulus of elasticity (capacity of the cell wall to resist changes in volume in response to changes in turgor) of leaves (MPa).
  - LeafAF: Apoplastic fraction (proportion of water outside the living cells) in leaves.
  - Vleaf: Storage water capacity in leaves, per leaf area (L/m2).
  - StemPI0: Osmotic potential at full turgor of symplastic xylem tissue (MPa).
  - StemEPS: Modulus of elasticity (capacity of the cell wall to resist changes in volume in response to changes in turgor) of symplastic xylem tissue (Mpa).
  - StemAF: Apoplastic fraction (proportion of water outside the living cells) in stem xylem.
  - Vstem: Storage water capacity in sapwood, per leaf area (L/m2).
• internalPhenology and internalWater: data frames to store internal state variables.
• internalFCCS: A data frame with fuel characteristics, according to fuel_FCCS (only if fireHazardResults = TRUE, in the control list).

Function growthInput() returns a list of class growthInput with the same elements as spwbInput, but with additional information.

• Element above includes the following additional columns:
  - LA_live: Live leaf area per individual (m2/ind).
  - LA_dead: Dead leaf area per individual (m2/ind).
  - SA: Live sapwood area per individual (cm2/ind).
• paramsGrowth: A data frame with growth parameters for each cohort:
  - RERleaf: Maintenance respiration rates (at 20ºC) for leaves (in g gluc·g dry-1·day-1).
  - RERsapwood: Maintenance respiration rates (at 20ºC) for sapwood (in g gluc·g dry-1·day-1).
  - RERfineroot: Maintenance respiration rates (at 20ºC) for fine roots (in g gluc·g dry-1·day-1).
  - CCleaf: Leaf construction costs (in g gluc·g dry-1).
  - CCsapwood: Sapwood construction costs (in g gluc·g dry-1).
  - CCfineroot: Fine root construction costs (in g gluc·g dry-1).

- RGRleafmax: Maximum leaf relative growth rate (in m2·cm-2·day-1).
- RGRsapwoodmax: Maximum sapwood relative growth rate (in cm2·cm-2·day-1).
- RGRfinerootmax: Maximum fine root relative growth rate (in g dry·g dry-1·day-1).
- SRsapwood: Sapwood daily senescence rate (in day-1).
- SRfineroot: Fine root daily senescence rate (in day-1).
- RSSG: Minimum relative starch for sapwood growth (proportion).
- fHDmin: Minimum value of the height-to-diameter ratio (dimensionless).
- fHDmax: Maximum value of the height-to-diameter ratio (dimensionless).
- WoodC: Wood carbon content per dry weight (g C /g dry).

• paramsMortalityRegeneration: A data frame with mortality/regeneration parameters for each cohort:
  - MortalityBaselineRate: Deterministic proportion or probability specifying the baseline reduction of cohort's density occurring in a year.
  - SurvivalModelStep: Time step in years of the empirical survival model depending on stand basal area (e.g. 10).
  - SurvivalB0: Intercept of the logistic baseline survival model depending on stand basal area.
  - SurvivalB1: Slope of the logistic baseline survival model depending on stand basal area.
  - RecrTreeDensity: Density of tree recruits from seeds.
  - IngrowthTreeDensity: Density of trees reaching ingrowth DBH.
  - RecrTreeDBH: DBH for tree recruits from seeds or resprouting (e.g. 1 cm).
  - IngrowthTreeDBH: Ingrowth DBH for trees (e.g. 7.5 cm).

• paramsAllometry: A data frame with allometric parameters for each cohort:
  - Aash: Regression coefficient relating the square of shrub height with shrub area.
  - Absh, Bbsh: Allometric coefficients relating phytovolume with dry weight of shrub individuals.
  - Acr, B1cr, B2cr, B3cr, C1cr, C2cr: Regression coefficients used to calculate crown ratio of trees.
  - Acw, Bcw: Regression coefficients used to calculated crown width of trees.

• internalAllocation: A data frame with internal allocation variables for each cohort:
  - allocationTarget: Value of the allocation target variable.
  - leafAreaTarget: Target leaf area (m2) per individual.
  - sapwoodAreaTarget: Target sapwood area (cm2) per individual.
  - fineRootBiomassTarget: Target fine root biomass (g dry) per individual.
  - crownBudPercent: Percentage of the crown with buds.

• internalCarbon: A data frame with the concentration (mol·gluc·l-1) of metabolic and storage carbon compartments for leaves and sapwood.

• internalMortality: A data frame to store the cumulative mortality (density for trees and cover for shrubs) predicted during the simulation, also distinguishing mortality due to starvation or dessication.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

resetInputs, spwb, soil, forest, SpParamsMED, defaultSoilParams, plant_ID

Examples

```
#Load example plot plant data
data(exampleforest)

# Example of aboveground parameters taken from a forest
# described using LAI and crown ratio
data(exampleforest2)

#Default species parameterization
data(SpParamsMED)


# Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

# Initialize control parameters using 'Granier' transpiration mode
control <- defaultControl("Granier")

# Prepare spwb input
spwbInput(exampleforest, examplesoil, SpParamsMED, control)

# Prepare input for 'Sperry' transpiration mode
control <- defaultControl("Sperry")
spwbInput(exampleforest,examplesoil,SpParamsMED, control)

# Prepare input for 'Sureau' transpiration mode
control <- defaultControl("Sureau")
spwbInput(exampleforest,examplesoil,SpParamsMED, control)

# Example of initialization from a forest
# described using LAI and crown ratio
control <- defaultControl("Granier")
spwbInput(exampleforest2, examplesoil, SpParamsMED, control)
```

---

modifyParams *Modify parameters*

Description

Routines to modify species parameter table or model input objects

Usage

```
modifySpParams(SpParams, customParams, subsetSpecies = TRUE)

modifyCohortParams(x, customParams, verbose = TRUE)

modifyInputParams(x, customParams, verbose = TRUE)
```

Arguments

`SpParams`	A species parameter data frame, typically `SpParamsMED`.
`customParams`	A data frame or a named vector with new parameter values (see details).
`subsetSpecies`	A logical flag to indicate that the output data frame should include only those species mentioned in `customParams`.
`x`	A model input object of class `spwbInput` or `growthInput`.
`verbose`	A logical flag to indicate that messages should be printed on the console.

Details

When calling function `modifySpParams`, `customParams` should be a data frame with as many rows as species and as many columns as parameters to modify, plus a column called 'Name' or 'Species' to match species names between the two tables. In both cases, the function will match input strings with column 'Name' of `x`. Alternatively, `customParams` can contain a column 'SpIndex' for matching of species indices, but this is deprecated.

When calling `modifyCohortParams`, `customParams` can be a data frame with as many rows as cohorts and as many columns as parameters to modify, plus a column called 'Cohort' which will be matched with the cohort names given by `spwbInput` or `growthInput`. Alternatively, `customParams` can be a named list or named numeric vector as for `modifyInputParams`.

When calling `modifyInputParams`, `customParams` must be either a named list or a named numeric vector. Cohort parameters are specified using the syntax "[cohortName]/[paramName]" for names (e.g. "T2_176/Z50" to modify parameter 'Z50' of cohort 'T2_176'). Soil layer parameters are specified using the syntax "[paramName]@#layer" for names, where #layer is the layer index (e.g. "rfc@1" will modify the rock fragment content of soil layer 1). Control parameters are specified using either "[paramName]" (e.g "phloemConductanceFactor") or "[paramName]$[subParamName]" (e.g "maximumRelativeGrowthRates$leaf"). It may seem unnecessary to modify soil or control parameters via a function, but `modifyInputParams` is called from optimization functions (see `optimization`).

Value

Function `modifySpParams` returns a modified species parameter data frame.

Functions `modifyCohortParams` and `modifyInputParams` return a modified `spwbInput` or `growthInput` object. Note that modifications may affect other parameters beyond those indicated in `customParams`, as a result of parameter dependencies (see examples).

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwbInput, SpParamsMED, optimization

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x1 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

# Cohort name for Pinus halepensis
PH_coh <- paste0("T1_", SpParamsMED$SpIndex[SpParamsMED$Name=="Pinus halepensis"])
PH_coh

# Modify Z50 and Z95 of Pinus halepensis cohort
customParams <- c(200,2000)
names(customParams) <- paste0(PH_coh,c("/Z50", "/Z95"))
x1m <- modifyInputParams(x1, customParams)

# Inspect original and modified objects
x1$below
x1m$below

# Inspect dependencies: fine root distribution across soil layers
x1$belowLayers$V
x1m$belowLayers$V

# Modify rock fragment content and sand proportion of soil layer 1
x1s <- modifyInputParams(x1, c("rfc@1" = 5, "sand@1" = 10))

# Inspect original and modified soils
x1$soil
x1s$soil

# When modifying growth input objects dependencies increase
x1 <- growthInput(exampleforest,examplesoil, SpParamsMED, control)
customParams <- c(2000,2)
names(customParams) <- paste0(PH_coh,c("/Al2As", "/LAI_live"))
x1m <- modifyInputParams(x1, customParams)
```

optimization *Multiple model runs and function factories for optimization routines*

Description

Function factories to generate functions to be used in model calibration, uncertainty or sensitivity analysis.

Usage

```
multiple_runs(
  parMatrix,
  x,
  meteo,
  latitude,
  elevation = NA,
  slope = NA,
  aspect = NA,
  summary_function = NULL,
  args = NULL,
  verbose = TRUE
)

optimization_function(
  parNames,
  x,
  meteo,
  latitude,
  elevation = NA,
  slope = NA,
  aspect = NA,
  summary_function,
  args = NULL
)

optimization_evaluation_function(
  parNames,
  x,
  meteo,
  latitude,
  elevation = NA,
  slope = NA,
  aspect = NA,
  measuredData,
  type = "SWC",
```

```
  cohorts = NULL,
  temporalResolution = "day",
  metric = "loglikelihood"
)

optimization_multicohort_function(
  cohortParNames,
  cohortNames,
  x,
  meteo,
  latitude,
  otherParNames = NULL,
  elevation = NA,
  slope = NA,
  aspect = NA,
  summary_function,
  args = NULL
)

optimization_evaluation_multicohort_function(
  cohortParNames,
  cohortNames,
  x,
  meteo,
  latitude,
  otherParNames = NULL,
  elevation = NA,
  slope = NA,
  aspect = NA,
  measuredData,
  type = "SWC",
  cohorts = cohortNames,
  temporalResolution = "day",
  metric = "loglikelihood"
)
```

Arguments

`parMatrix` A matrix of parameter values with runs in rows and parameters in columns. Column names should follow parameter modification naming rules (see examples and naming rules in `modifyInputParams`).

`x` An object of class `spwbInput` or `growthInput`.

`meteo`, `latitude`, `elevation`, `slope`, `aspect`
Additional parameters to simulation functions `spwb` or `growth`.

`summary_function`
A function whose input is the result of `spwb` or `growth`. The function must return a numeric scalar in the case of `optimization_function`, but is not restricted in the case of `multiple_runs`.

`args`	A list of additional arguments of `optimization_function`.
`verbose`	A flag to indicate extra console output.
`parNames`	A string vector of parameter names (see examples and naming rules in `modifyInputParams`).
`measuredData`	A data frame with observed/measured values. Dates should be in row names, whereas columns should be named according to the type of output to be evaluated (see details).
`type`	A string with the kind of model output to be evaluated. Accepted values are `"SWC"` (soil moisture content), `"REW"` relative extractable water, `"ETR"` (total evapotranspiration), `"E"` (transpiration per leaf area), `"LFMC"` (live fuel moisture content) and `"WP"` (plant water potentials).
`cohorts`	A string or a vector of strings with the cohorts to be compared (e.g. "T1_68"). If several cohort names are provided, the function `optimization_cohorts_function` evaluates the performance for each one and provides the mean value. If `NULL` results for the first cohort will be evaluated.
`temporalResolution`	A string to indicate the temporal resolution of the model evaluation, which can be "day", "week", "month" or "year". Observed and modelled values are aggregated temporally (using either means or sums) before comparison.
`metric`	An evaluation metric (see `evaluation_metric`).
`cohortParNames`	A string vector of vegetation parameter names for cohorts (e.g. 'Z95' or 'psiExtract').
`cohortNames`	A string vector of cohort names. All cohorts will be given the same parameter values for each parameter in 'cohortParNames'.
`otherParNames`	A string vector of parameter names (see examples and naming rules in `modifyInputParams`) for non-vegetation parameters (i.e. control parameters and soil parameters).

Details

See `evaluation` for details regarding how to specify measured data.

Functions produced by these function factories should be useful for sensitivity analyses using package 'sensitivity'.

Parameter naming (i.e. `parNames`) should follow the rules specified in section details of `modifyInputParams`. The exception to the naming rules applies when multiple cohorts are to be modified to the same values with functions `optimization_multicohort_function` and `optimization_evaluation_multicohort_function`. Then, only a vector of parameter names is supplied for `cohortParNames`.

Value

Function `multiple_runs` returns a list, whose elements are either the result of calling simulation models or the result of calling `summary_function` afterwards.

Function `optimization_function` returns a function whose parameters are parameter values and whose return is a prediction scalar (e.g. total transpiration).

Function `optimization_evaluation_function` returns a function whose parameters are parameter values and whose return is an evaluation metric (e.g. loglikelihood of the data observations given model predictions). If evaluation data contains information for different cohorts (e.g. plant water potentials or transpiration rates) then the evaluation is performed for each cohort and the metrics are averaged.

Function `optimization_multicohorts_function` returns a function whose parameters are parameter values and whose return is a prediction scalar (e.g. total transpiration). The difference with `optimization_function` is that multiple cohorts are set to the same parameter values.

Function `optimization_evaluation_multicohort_function` returns a function whose parameters are parameter values and whose return is an evaluation metric (e.g. loglikelihood of the data observations given model predictions). If evaluation data contains information for different cohorts (e.g. plant water potentials or transpiration rates) then the evaluation is performed for each cohort and the metrics are averaged. The difference with `optimization_evaluation_function` is that multiple cohorts are set to the same parameter values.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`evaluation_metric`, `modifyInputParams`, `spwb`, `growth`

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x1 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

# Cohort name for Pinus halepensis
PH_coh <- paste0("T1_", SpParamsMED$SpIndex[SpParamsMED$Name=="Pinus halepensis"])
PH_coh

#Parameter names of interest
parNames <- c(paste0(PH_coh,"/Z50"), paste0(PH_coh,"/Z95"))

#Specify parameter matrix
parMatrix <- cbind(c(200,300), c(500,1000))
colnames(parMatrix) <- parNames
```

```
#Define a summary function as the total transpiration over the simulated period
sf<-function(x) {sum(x$WaterBalance$Transpiration, na.rm=TRUE)}

#Perform two runs and evaluate the summary function
multiple_runs(parMatrix,
              x1, examplemeteo, latitude = 42, elevation = 100,
              summary_function = sf)

#Load observed data (in this case the same simulation results with some added error)
# Generate a prediction function for total transpiration over the simulated period
# as a function of parameters "Z50" and "Z95" for Pinus halepensis cohort
of<-optimization_function(parNames = parNames,
                          x = x1,
                          meteo = examplemeteo,
                          latitude = 41.82592, elevation = 100,
                          summary_function = sf)

# Evaluate for the values of the parameter matrix
of(parMatrix[1, ])
of(parMatrix)


# Generate a loglikelihood function for soil water content
# as a function of parameters "Z50" and "Z95" for Pinus halepensis cohort
data(exampleobs)
oef<-optimization_evaluation_function(parNames = parNames,
                                      x = x1,
                              meteo = examplemeteo, latitude = 41.82592, elevation = 100,
                                      measuredData = exampleobs, type = "SWC",
                                      metric = "loglikelihood")

# Loglikelihood for the values of the parameter matrix
oef(parMatrix[1, ])
oef(parMatrix)
```

`Parameter means` *Parameter average values*

Description

Internal data set with parameter averages for taxonomic families. This is used by input initialization functions to provide suitable parameter values when missing from species parameter tables.

Format

Data frame `trait_family_means` has taxonomic families in rows and parameter names as columns.

Source

Same sources as SpParamsMED

See Also

SpParamsMED, spwbInput

Examples

```
medfate::trait_family_means
```

---

plot.forest *Plot forest attributes*

Description

Convenient wrappers for vertical forest profiles (see vprofile_leafAreaDensity).

Usage

```
## S3 method for class 'forest'
plot(
  x,
  SpParams,
  type = "LeafAreaDensity",
  byCohorts = FALSE,
  bySpecies = FALSE,
  includeHerbs = FALSE,
  ...
)

## S3 method for class 'forest'
shinyplot(x, SpParams, ...)
```

Arguments

x	An object of class forest.
SpParams	A data frame with species parameters (see SpParamsMED).
type	A string of the plot type: "LeafAreaDensity", "RootDistribution", "FuelBulkDensity", "PARExtinction", "SWRExtinction" or "WindExtinction".
byCohorts	A logical flag to separate profiles for each cohort.
bySpecies	A logical flag to aggregate results by species.
includeHerbs	A logical flag to include herbaceous layer in the profile.
...	Additional parameters to vertical profiles

Value

A ggplot or a shiny application, depending on the function.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

forest, summary.forest, vprofile_leafAreaDensity

Examples

```
data(exampleforest)
data(SpParamsMED)
plot(exampleforest, SpParamsMED)
```

---

plot.spwb *Plots simulation results*

Description

Function plot plots time series of the results of the soil plant water balance model (see spwb), plant water balance model (see pwb), the forest growth model (see growth) or the forest dynamics model (see fordyn).

Usage

```
## S3 method for class 'spwb'
plot(
  x,
  type = "PET_Precipitation",
  cohorts = NULL,
  bySpecies = FALSE,
  dates = NULL,
  subdaily = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  summary.freq = NULL,
  ...
)

## S3 method for class 'pwb'
plot(
```

```
  x,
  type = "PlantTranspiration",
  cohorts = NULL,
  bySpecies = FALSE,
  dates = NULL,
  subdaily = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  summary.freq = NULL,
  ...
)

## S3 method for class 'growth'
plot(
  x,
  type = "PET_Precipitation",
  cohorts = NULL,
  bySpecies = FALSE,
  dates = NULL,
  subdaily = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  summary.freq = NULL,
  ...
)

## S3 method for class 'fordyn'
plot(
  x,
  type = "StandBasalArea",
  cohorts = NULL,
  bySpecies = FALSE,
  dates = NULL,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  summary.freq = NULL,
  ...
)
```

Arguments

x An object of class spwb, pwb, growth or fordyn.

type	The information to be plotted (see details)
cohorts	An integer, boolean or character vector to select the plant cohorts to be plotted. If cohorts = "T" (resp. cohorts = "S") then all tree (resp. shrub) cohorts will be displayed.
bySpecies	Allows aggregating output by species, before drawing plots (only has an effect with some values of type). Aggregation can involve a sum (as for plant lai or transpiration) or a LAI-weighted mean (as for plant stress or plant water potential), where LAI values are those of LAIlive.
dates	A Date vector with a subset of dates to be plotted.
subdaily	Whether subdaily results should be shown, only for simulations using transpirationMode = "Sperry" and having set subdailyResults = TRUE in the simulation control object. If subdaily = TRUE, then the valid strings for type are listed in plot.spwb_day.
xlim	Range of values for x.
ylim	Range of values for y.
xlab	x-axis label.
ylab	y-axis label.
summary.freq	Frequency of summary statistics (see cut.Date).
...	Additional parameters for function plot (not used).

Details

The following plots are currently available for spwb (most of them also for pwb):

- "PET_Precipitation": Potential evapotranspiration and Precipitation.
- "PET_NetRain": Potential evapotranspiration and Net rainfall.
- "Snow": Snow precipitation and snowpack dynamics.
- "Export": Water exported through deep drainage and surface runoff.
- "Evapotranspiration": Plant transpiration and soil evaporation.
- "SoilPsi": Soil water potential.
- "SoilRWC": Soil relative water content (in percent of field capacity).
- "SoilTheta": Soil moisture water content (in percent volume).
- "SoilVol": Soil water volumetric content (in mm).
- "PlantExtraction": Water extracted by plants from each soil layer.
- "HydraulicRedistribution": Water added to each soil layer coming from other soil layers, transported through the plant hydraulic network.
- "LAI": Expanded and dead leaf area index of the whole stand.
- "PlantLAI": Plant cohort leaf area index (expanded leaves).
- "PlantLAIlive": Plant cohort leaf area index ("live" leaves).
- "PlantStress": Plant cohort average daily drought stress.
- "PlantTranspiration": Plant cohort transpiration.

- "TranspirationPerLeaf": Plant cohort transpiration per leaf area.
- "PlantGrossPhotosynthesis": Plant cohort photosynthesis.
- "GrossPhotosynthesisPerLeaf": Plant cohort photosynthesis per leaf area.
- "StemRWC": Average daily stem relative water content.
- "LeafRWC": Average daily leaf relative water content.
- "LFMC": Live fuel moisture content.

The following plots are available for spwb and pwb *only if* transpirationMode = "Granier":

- "PlantPsi": Plant cohort water potential.
- "FPAR": Fraction of PAR at the canopy level of each plant cohort.
- "AbsorbedSWRFraction": Fraction of SWR absorbed by each plant cohort.

The following plots are available for spwb and pwb *only if* transpirationMode = "Sperry":

- "SoilPlantConductance": Average instantaneous overall soil plant conductance (calculated as the derivative of the supply function).
- "LeafPsiMin": Midday leaf water potential.
- "LeafPsiMax": Pre-dawn leaf water potential.
- "LeafPsiRange": Range of leaf water potential.
- "LeafPsiMin_SL": Minimum water potential of sunlit leaves.
- "LeafPsiMax_SL": Maximum water potential of sunlit leaves.
- "LeafPsiMin_SH": Minimum water potential of shade leaves.
- "LeafPsiMax_SH": Maximum water potential of shade leaves.
- "TempMin_SL": Minimum temperature of sunlit leaves.
- "TempMax_SL": Maximum temperature of sunlit leaves.
- "TempMin_SH": Minimum temperature of shade leaves.
- "TempMax_SH": Maximum temperature of shade leaves.
- "GSWMin_SL": Minimum stomatal conductance of sunlit leaves.
- "GSWMax_SL": Maximum stomatal conductance of sunlit leaves.
- "GSWMin_SH": Minimum stomatal conductance of shade leaves.
- "GSWMax_SH": Maximum stomatal conductance of shade leaves.
- "StemPsi": Midday (upper) stem water potential.
- "RootPsi": Midday root crown water potential.
- "PlantNetPhotosynthesis": Plant cohort net photosynthesis.
- "NetPhotosynthesisPerLeaf": Plant cohort net photosynthesis per leaf area.
- "PlantWUE": Plant cohort daily water use efficiency.
- "PlantAbsorbedSWR": Plant cohort absorbed short wave radiation.
- "AbsorbedSWRPerLeaf": Plant cohort absorbed short wave radiation per leaf area.
- "PlantNetLWR": Plant cohort net long wave radiation.

- "NetLWRPerLeaf": Plant cohort net long wave radiation per leaf area.
- "AirTemperature": Minimum/maximum/mean daily temperatures above canopy.
- "CanopyTemperature": Minimum/maximum/mean daily temperatures inside canopy.
- "SoilTemperature": Minimum/maximum/mean daily temperatures inside the first soil layer.
- "CanopyEnergyBalance": Canopy energy balance components.
- "SoilEnergyBalance": Soil energy balance components.

In addition to the former, the following plots are available for objects growth or fordyn:

- "CarbonBalance": Stand-level carbon balance components.
- "BiomassBalance": Stand-level biomass balance components.
- "GrossPhotosynthesis": Gross photosynthesis rate per dry weight.
- "MaintenanceRespiration": Maintenance respiration cost per dry weight.
- "PhotosynthesisMaintenanceRatio": The ratio of gross photosynthesis over maintenance respiration.
- "RootExudation": Root exudation rate per dry weight.
- "LabileCarbonBalance": Labile carbon balance per dry weight.
- "SugarLeaf": Sugar concentration in leaves.
- "StarchLeaf": Starch concentration in leaves.
- "SugarSapwood": Sugar concentration in sapwood.
- "StarchSapwood": Starch concentration in sapwood.
- "SugarTransport": Phloem sugar transport rate.
- "StructuralBiomassBalance": Daily structural biomass balance (g dry · ind-2).
- "LabileBiomassBalance": Daily labile biomass balance (g dry · ind-2).
- "PlantBiomassBalance": Daily plant biomass balance, i.e. labile change + structural change (g dry · ind-2).
- "MortalityBiomassLoss": Biomass loss due to mortality (g dry · m-2).
- "PlantBiomassBalance": Daily cohort biomass balance (including mortality) (g dry · m-2).
- "LeafBiomass": Leaf structural dry biomass per individual.
- "SapwoodBiomass": Sapwood dry biomass per individual.
- "FineRootBiomass": Fine root dry biomass per individual.
- "SapwoodArea": Sapwood area per individual.
- "LeafArea": Leaf area per individual.
- "FineRootArea": Fine root area per individual (only for transpirationMode = "Sperry" or transpirationMode = "Sureau").
- "DBH": Diameter at breast height (in cm) for an average individual of each plant cohort.
- "Height": Height (in cm) for an average individual of each plant cohort.
- "SAgrowth": Sapwood area growth rate.
- "LAgrowth": Leaf area growth rate.

- "FRAgrowth": Fine root area growth rate (only for transpirationMode = "Sperry" or transpirationMode = "Sureau").
- "HuberValue": Ratio of leaf area to sapwood area.
- "RootAreaLeafArea": Ratio of fine root area to leaf area (only for transpirationMode = "Sperry" or transpirationMode = "Sureau").

Finally, the following plots are only available for fordyn simulation results:

- "StandBasalArea": Stand basal area of living trees.
- "StandDensity": Stand density of living trees.
- "SpeciesBasalArea": Basal area of living trees by species.
- "SpeciesDensity": Density of living trees by species.
- "CohortBasalArea": Basal area of living trees by plant cohort.
- "CohortDensity": Density of living trees by plant cohort.

Value

An ggplot object

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwb, pwb, growth, fordyn, summary.spwb

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S1 <- spwb(x, examplemeteo, latitude = 41.82592, elevation = 100)
```

```
#Plot results
plot(S1)
```

`plot.spwb_day` *Plots simulation results for one day*

Description

Functions to plot the sub-daily simulation results of `spwb_day`, `growth_day` or the transpiration calculations of `transp_transpirationSperry` or `transp_transpirationSureau`.

Usage

```
## S3 method for class 'spwb_day'
plot(
  x,
  type = "PlantTranspiration",
  bySpecies = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)

## S3 method for class 'growth_day'
plot(
  x,
  type = "PlantTranspiration",
  bySpecies = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)

## S3 method for class 'pwb_day'
plot(
  x,
  type = "PlantTranspiration",
  bySpecies = FALSE,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
```

```
    ...
)
```

Arguments

x	An object of class spwb_day, growth_day or pwb_day.
type	The information to be plotted (see details).
bySpecies	Allows aggregating output by species, before drawing plots. Aggregation can involve a sum (as for plant LAI or transpiration) or a LAI-weighted mean (as for plant stress or plant water potential).
xlim	Range of values for x.
ylim	Range of values for y.
xlab	x-axis label.
ylab	y-axis label.
...	Additional parameters for function plot.

Details

The following plots are available for spwb_day and pwb_day:

- "LeafPsi": Leaf water potential (for shade and sunlit leaves).
- "LeafPsiAverage": Average leaf water potential.
- "RootPsi": Root crown water potential.
- "StemPsi": Stem water potential.
- "StemPLC": (Average) percentage of loss conductance in the stem conduits.
- "StemRWC": (Average) relative water content in the stem.
- "LeafRWC": Relative water content in the leaf.
- "StemSympRWC": (Average) relative water content in the stem symplasm.
- "LeafSympRWC": Relative water content in the leaf symplasm.
- "SoilPlantConductance": Overall soil plant conductance (calculated as the derivative of the supply function).
- "PlantExtraction": Water extracted from each soil layer.
- "PlantTranspiration": Plant cohort transpiration per ground area.
- "TranspirationPerLeaf": Plant cohort transpiration per leaf area.
- "PlantGrossPhotosynthesis": Plant cohort gross photosynthesis per ground area.
- "GrossPhotosynthesisPerLeaf": Plant cohort gross photosynthesis per leaf area.
- "PlantNetPhotosynthesis": Plant cohort net photosynthesis per ground area.
- "NetPhotosynthesisPerLeaf": Plant cohort net photosynthesis per leaf area.
- "LeafTranspiration": Instantaneous transpiration per leaf area (differentiates sunlit and shade leaves).
- "LeafGrossPhotosynthesis": Instantaneous gross photosynthesis per leaf area (differentiates sunlit and shade leaves).

- "LeafNetPhotosynthesis": Instantaneous net photosynthesis per leaf area (differentiates sunlit and shade leaves).
- "LeafAbsorbedSWR": Absorbed short wave radiation per leaf area (differentiates sunlit and shade leaves).
- "LeafAbsorbedPAR": Absorbed photosynthetically-active radiation per leaf area (differentiates sunlit and shade leaves).
- "LeafNetLWR": Net long wave radiation per leaf area (differentiates sunlit and shade leaves).
- "LeafCi": Leaf intercellular CO2 concentration (differentiates sunlit and shade leaves).
- "LeafIntrinsicWUE": Leaf intrinsic water use efficiency, i.e. the ratio between instantaneous photosynthesis and stomatal conductance (differentiates sunlit and shade leaves).
- "LeafVPD": Leaf vapour pressure deficit (differentiates sunlit and shade leaves).
- "LeafStomatalConductance": Leaf stomatal conductance to water vapour (differentiates sunlit and shade leaves).
- "LeafTemperature": Leaf temperature (differentiates sunlit and shade leaves).
- "Temperature": Above-canopy, inside-canopy and soil temperature.
- "CanopyEnergyBalance": Canopy energy balance components.
- "SoilEnergyBalance": Soil energy balance components.
- "PlantWaterBalance": Difference between water extraction from the soil and transpired water per ground area.
- "WaterBalancePerLeaf": Difference between water extraction from the soil and transpired water per leaf area.

And the following plots are additionally available for growth_day:

- "GrossPhotosynthesis": Gross photosynthesis rate per dry weight.
- "MaintenanceRespiration": Maintenance respiration cost per dry weight.
- "RootExudation": Root exudation rate per dry weight.
- "LabileCarbonBalance": Labile carbon balance per dry weight.
- "SugarLeaf": Sugar concentration in leaves.
- "StarchLeaf": Starch concentration in leaves.
- "SugarSapwood": Sugar concentration in sapwood.
- "StarchSapwood": Starch concentration in sapwood.
- "SugarTransport": Phloem sugar transport rate.

Value

An ggplot object

Note

Only for soil plant water balance simulations using transpirationMode = "Sperry" or transpirationMode = "Sureau". This function can be used to display subdaily dynamics of corresponding to single days on spwb runs, if control option subdailyResults is set to TRUE. See also option subdaily in plot.spwb.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwb_day, growth_day, plot.spwb

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (2 layers)
examplesoil <- defaultSoilParams(4)

#Switch to 'Sperry' transpiration mode
control <- defaultControl("Sperry")

#Enable subdaily results for plotting
control["subdailyResults"] <- TRUE

#Initialize model inputs
x2 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Simulate one day only
d <- 100
date <- examplemeteo$dates[d]
meteovec <- unlist(examplemeteo[d,])
sd2 <- spwb_day(x2, date, meteovec,
              latitude = 41.82592, elevation = 100, slope= 0, aspect = 0)

#Display transpiration for subdaily steps
plot(sd2, "PlantTranspiration")
```

---

poblet_trees *Example forest inventory data*

Description

Example data to illustrate the creation of forest objects from inventory data, coming from a forest inventory survey, used to illustrate the general function forest_mapTreeTable:

- poblet_trees - Data frame with example tree plot data from Poblet, Catalonia (717 observations and 4 variables).
  - Plot.Code - Plot ID (character)
  - Indv.Ref - Tree individual (integer)
  - Species - Species name (character)
  - Diameter.cm - Tree diameter at breast height (cm)

Source

- Data table poblet_trees corresponds to field data sampled by the Catalan Forest Ownership Center (Centre de la Propietat Forestal; CPF).

See Also

forest_mapTreeTable

---

resetInputs *Reset simulation inputs*

Description

Function resetInputs() allows resetting state variables in x to their defaults.

Usage

```
resetInputs(x)
```

Arguments

x	An object of class spwbInput or growthInput.

Value

Does not return any value. Instead, it modifies input object x.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwbInput, growthInput, spwb

resistances *Soil-plant resistances*

Description

Calculates and draws rhizosphere, root, stem and leaf resistances for simulation time steps

Usage

```
resistances(
  x,
  cohort,
  relative = FALSE,
  draw = FALSE,
  cumulative = FALSE,
  xlab = NULL,
  ylab = NULL
)
```

Arguments

x	An object of class spwb, pwb, growth or fordyn. The function only works with the result of simulations with transpirationMode = "Sperry" or transpirationMode = "Sureau".
cohort	An string indicating the cohort for which resistances are desired.
relative	A boolean flag to indicate that relative percentages are desired as output.
draw	A boolean flag to indicate that a plot should be drawn (only pathway resistances, without discriminating between soil layers).
cumulative	A flag to indicate that drawn series should be cumulative.
xlab	x-axis label.
ylab	y-axis label.

Details

The function makes internal calls to hydraulics_soilPlantResistancesWeibull or hydraulics_soilPlantResistance depending on the value of transpirationMode in x.

Value

If draw = FALSE, the function returns list with three items:

- pathway: A data frame with dates in rows and resistance segments in columns (Rhizosphere, Root, Stem and Leaf).
- root: A data frame with dates in rows and root resistances for soil layers in columns.
- rhizosphere: A data frame with dates in rows and rhizosphere resistances for soil layers in columns.

Values depend on whether relative = TRUE (percentages) or relative = FALSE (absolute resistance values).

If draw = TRUE, a plot object is returned showing the time series of pathway resistances.

Author(s)

Miquel De Cáceres Ainsa, CREAF

Léa Veuillen, INRAE-URFM

See Also

waterUseEfficiency, droughtStress

SFM_metric *Standard fuel models (Albini 1976, Scott & Burgan 2005)*

Description

Standard fuel models converted to metric system. Copied from package 'Rothermel' (Giorgio Vacchiano, Davide Ascoli).

Format

A data frame including standard fuel models as in Albini (1976) and Scott and Burgan (2005), to be used as input of fire_Rothermel function. All values converted to metric format.

Fuel_Model_Type A factor with levels D (for dynamic) or S (for static).

Load_1h Loading of 1h fuel class [t/ha].

Load_10h Loading of 10h fuel class [t/ha].

Load_100h Loading of 100h fuel class [t/ha]

Load_Live_Herb Loading of herbaceous fuels [t/ha]

Load_Live_Woody Loading of woody fuels [t/ha]

'SA/V_1h' Surface area to volume ratio of 1h fuel class [m2/m3]

'SA/V_10h' Surface area to volume ratio of 10h fuel class [m2/m3]

'SA/V_100h' Surface area to volume ratio of 100h fuel class [m2/m3]

'SA/V_Live_Herb' Surface area to volume ratio of herbaceous fuels [m2/m3]

'SA/V_Live_Woody' Surface area to volume ratio of woody fuels [m2/m3]

Fuel_Bed_Depth Fuel bed depth [cm]

Mx_dead Dead fuel moisture of extinction [percent]

Heat_1h Heat content of 1h fuel class [kJ/kg]

Heat_10h Heat content of 10h fuel class [kJ/kg]

Heat_100h Heat content of 100h fuel class [kJ/kg]

Heat_Live_Herb Heat content of herbaceous fuels [kJ/kg]

Heat_Live_Woody Heat content of woody fuels [kJ/kg]

Source

Albini, F. A. (1976). Computer-based models of wildland fire behavior: A users' manual. Ogden, UT: US Department of Agriculture, Forest Service, Intermountain Forest and Range Experiment Station.

Scott, J., and Burgan, R. E. (2005). A new set of standard fire behavior fuel models for use with Rothermel's surface fire spread model. Gen. Tech. Rep. RMRSGTR-153. Fort Collins, CO: US Department of Agriculture, Forest Service, Rocky Mountain Research Station.

See Also

fire_Rothermel

Examples

```
data(SFM_metric)
```

---

shinyplot *Shiny app with interactive plots*

Description

Creates a shiny app with interactive plots for simulation results and evaluation

Usage

```
shinyplot(x, ...)

## S3 method for class 'growth'
shinyplot(x, measuredData = NULL, ...)

## S3 method for class 'spwb'
shinyplot(x, measuredData = NULL, ...)

## S3 method for class 'pwb'
shinyplot(x, measuredData = NULL, ...)

## S3 method for class 'fordyn'
shinyplot(x, measuredData = NULL, ...)

## S3 method for class 'growth_day'
shinyplot(x, ...)

## S3 method for class 'spwb_day'
shinyplot(x, ...)

## S3 method for class 'pwb_day'
shinyplot(x, ...)
```

Arguments

x	An object of the right class.
...	Additional parameters.
measuredData	A data frame with observed/measured values (see evaluation_plot).

Details

Only run this function in interactive mode. When measuredData is not NULL, an additional panel is shown for evaluation plots.

Value

An object that represents the shiny app

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

plot.spwb, evaluation_plot

---

soil — *Soil initialization*

Description

Initializes soil parameters and state variables for its use in simulations.

Usage

```
soil(x, VG_PTF = "Toth")

## S3 method for class 'soil'
summary(object, model = "SX", ...)
```

Arguments

x	A data frame of soil parameters (see an example in defaultSoilParams).
VG_PTF	Pedotransfer functions to obtain parameters for the van Genuchten-Mualem equations. Either "Carsel" (Carsel and Parrish 1988) or "Toth" (Toth et al. 2015).
object	An object of class soil.
model	Either 'SX' or 'VG' for Saxton or Van Genuchten pedotransfer models.
...	Additional parameters to summary.

Details

Function summary prompts a description of soil characteristics and state variables (water content and temperature) according to a water retention curve (either Saxton's or Van Genuchten's). Volume at field capacity is calculated assuming a soil water potential equal to -0.033 MPa. Parameter Temp is initialized as missing for all soil layers.

If available, the user can specify columns VG_alpha, VG_n, VG_theta_res, VG_theta_sat and K_sat, to override Van Genuchten parameters an saturated conductivity estimated from pedotransfer functions when calling function soil.

Value

Function soil returns a data frame of class soil with the following columns:

- widths: Width of soil layers (in mm).
- sand: Sand percentage for each layer (in percent volume).
- clay: Clay percentage for each layer (in percent volume).
- om: Organic matter percentage for each layer (in percent volume).
- nitrogen: Sum of total nitrogen (ammonia, organic and reduced nitrogen) for each layer (in g/kg).
- rfc: Percentage of rock fragment content for each layer.
- macro: Macroporosity for each layer (estimated using Stolf et al. 2011).
- Ksat: Saturated soil conductivity for each layer (estimated using function soil_saturatedConductivitySX.
- VG_alpha, VG_n, VG_theta_res, VG_theta_sat: Parameters for van Genuchten's pedotransfer functions, for each layer, corresponding to the USDA texture type.
- W: State variable with relative water content of each layer (in as proportion relative to FC).
- Temp: State variable with temperature (in ºC) of each layer.

Author(s)

Miquel De Cáceres Ainsa, CREAF

References

Carsel, R.F., and Parrish, R.S. 1988. Developing joint probability distributions of soil water retention characteristics. Water Resources Research 24: 755–769.

Tóth, B., Weynants, M., Nemes, A., Makó, A., Bilas, G., and Tóth, G. 2015. New generation of hydraulic pedotransfer functions for Europe. European Journal of Soil Science 66: 226–238.

Stolf, R., Thurler, A., Oliveira, O., Bacchi, S., Reichardt, K., 2011. Method to estimate soil macroporosity and microporosity based on sand content and bulk density. Rev. Bras. Ciencias do Solo 35, 447–459.

See Also

soil_redefineLayers, soil_psi2thetaSX, soil_psi2thetaVG, spwb, defaultSoilParams

Examples

```
# Default parameters
df_soil <- defaultSoilParams()

# Initializes soil
s = soil(df_soil)
s

# Prints soil characteristics according to Saxton's water retention curve
summary(s, model="SX")

# Prints soil characteristics according to Van Genuchten's water retention curve
summary(s, model="VG")

# Add columns 'VG_theta_sat' and 'VG_theta_res' with custom values
df_soil$VG_theta_sat <- 0.400
df_soil$VG_theta_res <- 0.040

# Reinitialize soil (should override estimations)
s2 = soil(df_soil)
s2
summary(s2, model="VG")
```

---

soil_redefineLayers   *Redefine soil layer widths*

Description

Allows redefining soil layer widths of an input data frame of soil parameters.

Usage

```
soil_redefineLayers(x, widths = c(300, 700, 1000, 2000))
```

Arguments

x	A data frame of soil parameters (see an example in defaultSoilParams) or an object of class soil.
widths	A numeric vector indicating the desired layer widths, in mm.

Details

If an initialized soil is supplied, its hydraulic parameters will be recalculated and the value of state variables will be lost.

Value

A data frame or soil object with soil parameters, depending on the class of x.

Author(s)

Víctor Granda, EMF-CREAF

Miquel De Cáceres Ainsa, EMF-CREAF

See Also

soil, defaultSoilParams

Examples

```
# Define initial soil with 5 layers
spar <- defaultSoilParams(5)
spar

# Redefine to four layers
soil_redefineLayers(spar)

# Same but after soil parameter initialization
examplesoil <- soil(spar)
examplesoil

soil_redefineLayers(examplesoil)
```

SpParams *Data tables with species parameter definitions and values*

Description

A data sets of species parameter definition and values, the latter resulting from existing databases, fit to empirical data or expert-based guesses.

Format

- Data frame `SpParamsDefinition` has parameters in rows and columns 'ParameterName', 'ParameterGroup', 'Definition', 'Type' and 'Units'.
- Data frames `SpParamsMED` has species or genus as rows and column names equal to parameter names in `SpParamsDefinition`.

Details

`SpParamsMED` was the official species parameter for package versions up to v.4.0.0, but will not be maintained in the future. Additional species parameter tables for different countries are distributed via package traits4models.

Examples

```
data(SpParamsDefinition)
data(SpParamsMED)
```

spwb *Soil-plant water balance*

Description

Function spwb() is a water balance model that determines changes in soil moisture, soil water potentials, plant transpiration and drought stress at daily steps for a given forest stand during a period specified in the input climatic data. Function pwb() performs plant water balance only (i.e. soil moisture dynamics is an input) at daily steps for a given forest stand during a period specified in the input climatic data. On both simulation functions plant transpiration and photosynthesis processes are conducted with different level of detail depending on the transpiration mode.

Usage

```
spwb(
  x,
  meteo,
  latitude,
  elevation,
  slope = NA_real_,
  aspect = NA_real_,
  CO2ByYear = numeric(0),
  waterTableDepth = NA_real_
)

pwb(
  x,
  meteo,
  W,
  latitude,
  elevation,
  slope = NA_real_,
  aspect = NA_real_,
  canopyEvaporation = numeric(0),
  snowMelt = numeric(0),
  soilEvaporation = numeric(0),
  herbTranspiration = numeric(0),
  CO2ByYear = numeric(0)
)
```

Arguments

x	An object of class spwbInput.
meteo	A data frame with daily meteorological data series. Row names of the data frame should correspond to date strings with format "yyyy-mm-dd" (see Date). Alternatively, a column called "dates" or "Dates" can contain Date or POSIXct classes. The following columns are required and cannot have missing values:

- MinTemperature: Minimum temperature (in degrees Celsius).
- MaxTemperature: Maximum temperature (in degrees Celsius).
- Precipitation: Precipitation (in mm).

The following columns are required but can contain missing values (NOTE: missing values will raise warnings):

- MinRelativeHumidity: Minimum relative humidity (in percent).
- MaxRelativeHumidity: Maximum relative humidity (in percent).
- Radiation: Solar radiation (in MJ/m2/day).

The following columns are optional:

- WindSpeed: Above-canopy wind speed (in m/s). This column may not exist, or can be left with NA values. In both cases simulations will assume a constant value specified in defaultControl.
- CO2: Atmospheric (above-canopy) CO2 concentration (in ppm). This column may not exist, or can be left with NA values. In both cases simulations will assume a constant value specified in defaultControl.
- Patm: Atmospheric pressure (in kPa). This column may not exist, or can be left with NA values. In both cases, a value is estimated from elevation.

latitude Latitude (in degrees).

elevation, slope, aspect
Elevation above sea level (in m), slope (in degrees) and aspect (in degrees from North).

CO2ByYear A named numeric vector with years as names and atmospheric CO2 concentration (in ppm) as values. Used to specify annual changes in CO2 concentration along the simulation (as an alternative to specifying daily values in meteo).

waterTableDepth
Water table depth (in mm). When not missing, capillarity rise will be allowed if lower than total soil depth.

W A matrix with the same number of rows as meteo and as many columns as soil layers, containing the soil moisture of each layer as proportion of field capacity.

canopyEvaporation
A vector of daily canopy evaporation (from interception) values (mm). The length should match the number of rows in meteo.

snowMelt A vector of daily snow melt values (mm). The length should match the number of rows in meteo.

soilEvaporation
A vector of daily bare soil evaporation values (mm). The length should match the number of rows in meteo.

herbTranspiration
A vector of daily herbaceous transpiration values (mm). The length should match the number of rows in meteo.

Details

The simulation functions allow using three different sub-models of transpiration and photosynthesis:

- The sub-model corresponding to 'Granier' transpiration mode is illustrated by function transp_transpirationGranie and was described in De Caceres et al. (2015), and implements an approach originally described in Granier et al. (1999).
- The sub-model corresponding to 'Sperry' transpiration mode is illustrated by function transp_transpirationSperry and was described in De Caceres et al. (2021), and implements a modelling approach originally described in Sperry et al. (2017).
- The sub-model corresponding to 'Sureau' transpiration mode is illustrated by function transp_transpirationSureau and was described for model SurEau-Ecos v2.0 in Ruffault et al. (2022).

Simulations using the 'Sperry' or 'Sureau' transpiration mode are computationally much more expensive than 'Granier'.

Value

Function spwb returns a list of class 'spwb' whereas function pwb returns a list of class 'pwb'. There are many elements in common in these lists, so they are listed here together:

- "latitude": Latitude (in degrees) given as input.
- "topography": Vector with elevation, slope and aspect given as input.
- "weather": A copy of the input weather data frame.
- "spwbInput": An copy of the object x of class spwbInput given as input.
- "spwbOutput": An copy of the final state of the object x of class spwbInput.
- "WaterBalance": A data frame where different variables (in columns) are given for each simulated day (in rows):
  - "PET": Potential evapotranspiration (in mm).
  - "Precipitation": Input precipitation (in mm).
  - "Rain": Precipitation as rainfall (in mm).
  - "Snow": Precipitation as snowfall (in mm).
  - "NetRain": Net rain, after accounting for interception (in mm).
  - "Infiltration": The amount of water infiltrating into the soil (in mm).
  - "InfiltrationExcess": Excess infiltration in the topmost layer leading to an increase in runoff (in mm).
  - "SaturationExcess": Excess saturation in the topmost layer leading to an increase in runoff (in mm).
  - "CapillarityRise": Water entering the soil via capillarity rise (mm) from the water table, if waterTableDepth is supplied.
  - "Runoff": The amount of water exported via surface runoff (in mm).
  - "DeepDrainage": The amount of water exported via deep drainage (in mm).
  - "Evapotranspiration": Evapotranspiration (in mm).
  - "SoilEvaporation": Bare soil evaporation (in mm).
  - "HerbTranspiration": Transpiration due to the herbaceous layer (in mm).
  - "PlantExtraction": Amount of water extracted from soil by woody plants (in mm).
  - "Transpiration": Woody plant transpiration (in mm).
  - "HydraulicRedistribution": Water redistributed among soil layers, transported through the plant hydraulic network.

- "EnergyBalance": A data frame with the daily values of energy balance components for the soil and the canopy (only for transpirationMode = "Sperry" or transpirationMode = "Sureau").
- "Temperature": A data frame with the daily values of minimum/mean/maximum temperatures for the atmosphere (input), canopy and soil (only for transpirationMode = "Sperry" or transpirationMode = "Sureau").
- "Soil": A list with the following subelements:
  - "SWC": Soil water content (percent of soil volume) in each soil layer (and overall).
  - "RWC": Relative soil moisture content (relative to field capacity) in each soil layer (and overall).
  - "REW": Relative extractable water (min. psi = -5 MPa) in each soil layer (and overall).
  - "ML": Soil water volume in each soil layer (in L/m2) (and overall).
  - "Psi": Soil water potential in each soil layer (in MPa) (and overall).
  - "PlantExt": Plant extraction from each soil layer (in mm) (and overall).
  - "HydraulicInput": Water that entered the layer coming from other layers and transported via the plant hydraulic network (in mm) (and overall).
- "Snow": A data frame where the following variable (in columns) is given for each simulated day (in rows):
  - "SWE": Snow water equivalent (mm) of the snow pack.
- "Stand": A data frame where different variables (in columns) are given for each simulated day (in rows):
  - "LAI": LAI of the stand (including the herbaceous layer and live + dead leaves of woody plants) (in m2/m2).
  - "LAIherb": LAI of the herbaceous layer (in m2/m2).
  - "LAIlive": LAI of the woody plants assuming all leaves are unfolded (in m2/m2).
  - "LAIexpanded": LAI of the woody plants with leaves actually unfolded (in m2/m2).
  - "LAIdead": LAI of the woody plants corresponding to dead leaves (in m2/m2).
  - "Cm": Water retention capacity of the canopy (in mm) (accounting for leaf phenology).
  - "LgroundPAR": The percentage of PAR that reaches the ground (accounting for leaf phenology).
  - "LgroundSWR": The percentage of SWR that reaches the ground (accounting for leaf phenology).
- "Plants": A list of daily results for plant cohorts (see below).
- "subdaily": A list of objects of class spwb_day, one per day simulated (only if required in control parameters, see defaultControl).

When transpirationMode = "Granier", element "Plants" is a list with the following subelements:

- "LAI": A data frame with the daily leaf area index for each plant cohort.
- "LAIlive": A data frame with the daily leaf area index for each plant cohort, assuming all leaves are unfolded (in m2/m2).
- "FPAR": A data frame with the fraction of PAR at the canopy level of each plant cohort.

- "AbsorbedSWRFraction": A data frame with the fraction of SWR absorbed by each plant cohort.
- "Transpiration": A data frame with the amount of daily transpiration (in mm) for each plant cohort.
- "GrossPhotosynthesis": A data frame with the amount of daily gross photosynthesis (in g C·m-2) for each plant cohort.
- "PlantPsi": A data frame with the average daily water potential of each plant (in MPa).
- "LeafPLC": A data frame with the average daily proportion of leaf conductance loss of each plant ([0-1]).
- "StemPLC": A data frame with the average daily proportion of stem conductance loss of each plant ([0-1]).
- "PlantWaterBalance": A data frame with the daily balance between transpiration and soil water extraction for each plant cohort.
- "LeafRWC": A data frame with the average daily leaf relative water content of each plant (in percent).
- "StemRWC": A data frame with the average daily stem relative water content of each plant (in percent).
- "LFMC": A data frame with the daily live fuel moisture content (in percent of dry weight).
- "PlantStress": A data frame with the amount of daily stress [0-1] suffered by each plant cohort (relative whole-plant conductance).

If transpirationMode="Sperry" or transpirationMode="Sureau", element "Plants" is a list with the following subelements:

- "LAI": A data frame with the daily leaf area index for each plant cohort.
- "AbsorbedSWR": A data frame with the daily SWR absorbed by each plant cohort.
- "NetLWR": A data frame with the daily net LWR by each plant cohort.
- "Transpiration": A data frame with the amount of daily transpiration (in mm) for each plant cohorts.
- "GrossPhotosynthesis": A data frame with the amount of daily gross photosynthesis (in g C·m-2) for each plant cohort.
- "NetPhotosynthesis": A data frame with the amount of daily net photosynthesis (in g C·m-2) for each plant cohort.
- "dEdP": A data frame with mean daily values of soil-plant conductance (derivative of the supply function) for each plant cohort.
- "PlantWaterBalance": A data frame with the daily balance between transpiration and soil water extraction for each plant cohort.
- "SunlitLeaves" and "ShadeLeaves": A list with daily results for sunlit and shade leaves:
  - "PsiMin": A data frame with the minimum (midday) daily sunlit or shade leaf water potential (in MPa).
  - "PsiMax": A data frame with the maximum (predawn) daily sunlit or shade leaf water potential (in MPa).

- "LeafPsiMin": A data frame with the minimum (midday) daily (average) leaf water potential of each plant (in MPa).
- "LeafPsiMax": A data frame with the maximum (predawn) daily (average) leaf water potential of each plant (in MPa).
- "LeafRWC": A data frame with the average daily leaf relative water content of each plant (in percent).
- "StemRWC": A data frame with the average daily stem relative water content of each plant (in percent).
- "LFMC": A data frame with the daily live fuel moisture content (in percent of dry weight).
- "StemPsi": A data frame with the minimum daily stem water potential of each plant (in MPa).
- "LeafPLC": A data frame with the average daily proportion of leaf conductance loss of each plant ([0-1]).
- "StemPLC": A data frame with the average daily proportion of stem conductance loss of each plant ([0-1]).
- "RootPsi": A data frame with the minimum daily root water potential of each plant (in MPa).
- "RhizoPsi": A list of data frames (one per plant cohort) with the minimum daily root water potential of each plant (in MPa).
- "PlantStress": A data frame with the amount of daily stress [0-1] suffered by each plant cohort (relative whole-plant conductance).

Author(s)

- Miquel De Cáceres Ainsa, CREAF
- Nicolas Martin-StPaul, URFM-INRAE

References

De Cáceres M, Martínez-Vilalta J, Coll L, Llorens P, Casals P, Poyatos R, Pausas JG, Brotons L. (2015) Coupling a water balance model with forest inventory data to predict drought stress: the role of forest structural changes vs. climate changes. Agricultural and Forest Meteorology 213: 77-90 (doi:10.1016/j.agrformet.2015.06.012).

De Cáceres M, Mencuccini M, Martin-StPaul N, Limousin JM, Coll L, Poyatos R, Cabon A, Granda V, Forner A, Valladares F, Martínez-Vilalta J (2021) Unravelling the effect of species mixing on water use and drought stress in holm oak forests: a modelling approach. Agricultural and Forest Meteorology 296 (doi:10.1016/j.agrformet.2020.108233).

Granier A, Bréda N, Biron P, Villette S (1999) A lumped water balance model to evaluate duration and intensity of drought constraints in forest stands. Ecol Modell 116:269–283. https://doi.org/10.1016/S0304-3800(98)00205-1.

Ruffault J, Pimont F, Cochard H, Dupuy JL, Martin-StPaul N (2022) SurEau-Ecos v2.0: a trait-based plant hydraulics model for simulations of plant water status and drought-induced mortality at the ecosystem level. Geoscientific Model Development 15, 5593-5626 (doi:10.5194/gmd-15-5593-2022).

Sperry, J. S., M. D. Venturas, W. R. L. Anderegg, M. Mencuccini, D. S. Mackay, Y. Wang, and D. M. Love. 2017. Predicting stomatal responses to the environment from the optimization of photosynthetic gain and hydraulic cost. Plant Cell and Environment 40, 816-830 (doi: 10.1111/pce.12852).

See Also

spwbInput, spwb_day, plot.spwb, extract, summary.spwb, forest, aspwb

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x1 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S1 <- spwb(x1, examplemeteo, latitude = 41.82592, elevation = 100)

#Switch to 'Sperry' transpiration mode
control <- defaultControl("Sperry")

#Initialize input
x2 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S2 <- spwb(x2, examplemeteo, latitude = 41.82592, elevation = 100)

#Switch to 'Sureau' transpiration mode
control <- defaultControl("Sureau")

#Initialize input
x3 <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S3 <- spwb(x3, examplemeteo, latitude = 41.82592, elevation = 100)
```

summary.forest *Summary of forest structure*

Description

Displays a summary of forest structure

Usage

```
## S3 method for class 'forest'
summary(object, SpParams, ...)

## S3 method for class 'summary.forest'
print(x, digits = getOption("digits"), ...)
```

Arguments

`object`	An object of class `forest`
`SpParams`	A data frame with species parameters (see `SpParamsMED`).
`...`	Additional parameters for functions `summary` and `print`.
`x`	The object returned by `summary.forest`.
`digits`	Minimal number of significant digits.

Details

Function `summary.forest` can be used to summarize a `forest` object in the console.

Value

Function `summary.forest` returns a list with several structural attributes, such as the basal area and LAI of the forest.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`forest`, `forest_mapWoodyTables`, `forest_mergeTrees`, `plot.forest`, `tree2forest`

Examples

```
# Summary of example forest
summary(exampleforest, SpParamsMED)
```

summary.spwb *Summarize simulation results*

Description

Function summary summarizes the model's output in different temporal steps (i.e. weekly, annual, ...).

Usage

```
## S3 method for class 'spwb'
summary(
  object,
  freq = "years",
  output = "WaterBalance",
  FUN = sum,
  bySpecies = FALSE,
  months = NULL,
  ...
)

## S3 method for class 'pwb'
summary(
  object,
  freq = "years",
  output = "WaterBalance",
  FUN = sum,
  bySpecies = FALSE,
  months = NULL,
  ...
)

## S3 method for class 'growth'
summary(
  object,
  freq = "years",
  output = "WaterBalance",
  FUN = sum,
  bySpecies = FALSE,
  months = NULL,
  ...
)

## S3 method for class 'fordyn'
summary(
  object,
  freq = "years",
```

```
  output = "WaterBalance",
  FUN = sum,
  bySpecies = FALSE,
  months = NULL,
  ...
)
```

Arguments

object	An object of class spwb, pwb, growth or fordyn.
freq	Frequency of summary statistics (see cut.Date).
output	The data table to be summarized. Accepted values are the path to data tables in object, such as 'WaterBalance', 'Soil', 'Stand' or 'Plants$LAI'. It is also possible to use strings like 'Transpiration' and the function will interpret it as 'Plants$Transpiration'.
FUN	The function to summarize results (e.g., sum, mean, ...)
bySpecies	Allows aggregating output by species before calculating summaries (only has an effect with some values of output). Aggregation can involve a sum (as for plant lai or transpiration) or a LAI-weighted mean (as for plant stress or plant water potential).
months	A vector of month numbers (1 to 12) to subset the season where summaries apply.
...	Additional parameters for function summary.

Value

A matrix with dates as row names and the desired summaries in columns

Note

When applied to fordyn objects, the summary function can be used to gather the results of different yearly steps into a single table while keeping a daily resolution (i.e. using freq = "days".

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

spwb, pwb, growth, fordyn, plot.spwb, extract

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)
```

```
#Default species parameterization
data(SpParamsMED)

#Define soil with default soil params (4 layers)
examplesoil <- defaultSoilParams(4)

#Initialize control parameters
control <- defaultControl("Granier")

#Initialize input
x <- spwbInput(exampleforest,examplesoil, SpParamsMED, control)

#Call simulation function
S1<-spwb(x, examplemeteo, latitude = 41.82592, elevation = 100)

#Queries the tables in 'Soil'
names(S1$Soil)

#Monthly summary (averages) of soil relative water content
summary(S1, freq="months",FUN=mean, output="RWC")

#Queries the tables in 'Plants'
names(S1$Plants)

#Monthly summary (averages) of plant stress
summary(S1, freq="months",FUN=mean, output="PlantStress",
        bySpecies = TRUE)
```

---

tree2forest *Single-cohort forests*

Description

Creates a forest object with a single plant cohort

Usage

```
tree2forest(
  Species,
  Height,
  LAI = NA,
  N = NA,
  DBH = NA,
  Z50 = NA,
  Z95 = NA,
  Z100 = NA,
```

```
  CrownRatio = NA,
  FoliarBiomass = NA,
  FuelLoading = NA
)

shrub2forest(
  Species,
  Height,
  LAI = NA,
  Cover = NA,
  Z50 = NA,
  Z95 = NA,
  Z100 = NA,
  CrownRatio = NA,
  FoliarBiomass = NA,
  FuelLoading = NA
)
```

Arguments

Species	String with species (taxon) name or a non-negative integer for species identity (i.e., 0,1,2,...) matching SpParams.
Height	Plant height (cm).
LAI	Leaf area index (m2/m2)
N	Tree density (ind/ha)
DBH	Tree DBH (cm).
Z50	Depth (in mm) corresponding to 50% of fine roots.
Z95	Depth (in mm) corresponding to 95% of fine roots.
Z100	Depth (in mm) corresponding to 100% of fine roots.
CrownRatio	Crown ratio (fraction of total height)
FoliarBiomass	Standing dry biomass of leaves (kg/m2)
FuelLoading	Fine fuel loading (kg/m2)
Cover	Percent cover

Value

An object of class forest

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

forest, emptyforest

Examples

```
oak_forest <-tree2forest("Quercus ilex", Height= 200, LAI = 2)
oak_forest
```

---

utils_ldrExploration *Optimization of root distribution*

Description

Functions `utils_ldrExploration` and `utils_ldrOptimization` are used to find optimum the species root distribution within `spwb`, given the arguments `x`, `meteo` and `psi_crit`.

Usage

```
utils_ldrExploration(
  x,
  meteo,
  cohorts = NULL,
  RZmin = 301,
  RZmax = 4000,
  V1min = 0.01,
  V1max = 0.94,
  resolution = 10,
  heat_stop = 0,
  transformation = "identity",
  verbose = FALSE,
  ...
)

utils_ldrOptimization(y, psi_crit, opt_mode = 1)
```

Arguments

`x`	An object of class `spwbInput`.
`meteo`	A data frame with daily meteorological data series (see `spwb`).
`cohorts`	A character string with the names of cohorts to be explored. If `NULL` then all cohorts are explored.
`RZmin`	The minimum value of RZ (the rooting depth) to be explored (in mm)
`RZmax`	The maximum value of RZ (the rooting depth) to be explored (in mm)
`V1min`	The minimum value of V1 (the root proportion in the first soil layer) to be explored
`V1max`	The maximum value of V1 (the root proportion in the first soil layer) to be explored

resolution	An integer defining the number of values to obtain by discretization of the root parameters RZ and V1. The number of parameter combinations and therefore the computation cost increases increase with the square of resolution
heat_stop	An integer defining the number of days during to discard from the calculation of the optimal root distribution. Usefull if the soil water content initialization is not certain
transformation	Function to modify the size of Z intervals to be explored (by default, bins are equal).
verbose	A logical value. Print the internal messages of the function?
...	Additional parameters to function `spwb`.
y	The result of calling `utils_ldrExploration`.
psi_crit	A numerical vector of length iqual to the number of species in the plot containing the species values of water potential inducing hydraulic failure (in MPa). Use `NA` values to skip optimization for particular plant cohorts.
opt_mode	Optimization mode:

- `opt_mode = 1` maximizes transpiration along the line of stress equal to `psi_crit` (Cabon et al. 2018). The optimization is based on the eco-hydrological equilibrium hypothesis (Eagleson, 1982), which is formulated here as the root distribution for which plant transpiration is maximized while the plant water potential is close to the species-defined critical value `psi_crit` (Cabon et al.,2018).
- `opt_mode = 2` maximizes transpiration among combinations with stress according to `psi_crit`).
- `opt_mode = 3` maximizes photosynthesis among combinations with stress according to `psi_crit`).
- `opt_mode = 4` maximizes transpiration, subject to root construction constrains, among combinations with stress according to `psi_crit`).
- `opt_mode = 5` maximizes photosynthesis, subject to root construction constrains, among combinations with stress according to `psi_crit`).

Details

For each combination of the parameters RZ and V1 the function `utils_ldrExploration` runs `spwb`, setting the total soil depth equal to RZ. The root proportion in each soil layer is derived from V1, the depth of the first soil layer and RZ using the LDR root distribution model (Schenk and Jackson, 2002) and assuming that the depth containing 95 percent of the roots is equal to RZ. Function `utils_ldrOptimization` takes the result of the exploration and tries to find optimum root distribution parameters. `psi_crit`, the species specific water potential inducing hydraulic failure, can be approached by the water potential inducing 50 percent of loss of conductance for the and gymnosperms and 88 percent for the angiosperms (Urli et al., 2013, Brodribb et al., 2010). Details of the hypothesis and limitations of the optimization method are given in Cabon et al. (2019).

Value

Function `utils_ldrExploration` returns a list containing a list containing the explored RZ and V1 combinations as well as arrays with the values of average daily plant transpiration, average daily net photosynthesis and the minimum plant water potential for each cohort and parameter combination.

Function `utils_ldrOptimization` returns a data frame with containing the species index used in medfate, `psi_crit` and the optimized values of V1 and the LDR parameters Z50 and Z95 (see `root_ldrDistribution`) and as many rows as the number of species.

Author(s)

Antoine Cabon, WSL

Miquel De Cáceres Ainsa, CREAF

References

Brodribb, T.J., Bowman, D.J.M.S., Nichols, S., Delzon, S., Burlett, R., 2010. Xylem function and growth rate interact to determine recovery rates after exposure to extreme water deficit. New Phytol. 188, 533–542. doi:10.1111/j.1469-8137.2010.03393.x

Cabon, A., Martínez-Vilalta, J., Poyatos, R., Martínez de Aragón, J., De Cáceres, M. (2018) Applying the eco-hydrological equilibrium hypothesis to estimate root ditribution in water-limited forests. Ecohydrology 11: e2015.

Eagleson, P.S., 1982. Ecological optimality in water-limited natural soil-vegetation systems: 1. Theory and hypothesis. Water Resour. Res. 18, 325–340. doi:10.1029/WR018i002p00325

Schenk, H.J., Jackson, R.B., 2002. The Global Biogeography of Roots. Ecol. Monogr. 72, 311. doi:10.2307/3100092

Urli, M., Porte, A.J., Cochard, H., Guengant, Y., Burlett, R., Delzon, S., 2013. Xylem embolism threshold for catastrophic hydraulic failure in angiosperm trees. Tree Physiol. 33, 672–683. doi:10.1093/treephys/tpt030

See Also

`utils_rockOptimization`, `spwb`, `soil`, `root_ldrDistribution`

`utils_rockOptimization` *Optimization of rock fragment content*

Description

Function `utils_rockOptimization` finds optimum rock fragment content in the soil corresponding to given vegetation, weather and target percent loss of conductance (PLC), following a modification of the method proposed by Druel et al. (2023).

Usage

```
utils_rockOptimization(
  x,
  soil,
  SpParams,
  control,
```

```
  meteo,
  PLCquantile = 0.9,
  qPLC_target = 12,
  qPLC_tol = 0.5,
  sew_min = 30,
  max_rocks = 99,
  verbose = FALSE,
  ...
)
```

Arguments

x	An object of class forest.
soil	An object of class data.frame or soil, containing soil parameters per soil layer.
SpParams	A data frame with species parameters (see SpParamsDefinition and SpParamsMED).
control	A list with default control parameters (see defaultControl).
meteo	A data frame with daily meteorological data series (see spwb).
PLCquantile	Maximum PLC quantile to be calculated across years.
qPLC_target	Target PLC to be achieved (by default 12%).
qPLC_tol	Tolerance of PLC difference to target accepted when finding solution.
sew_min	Minimum soil extractable water (mm) for rock exploration.
max_rocks	Maximum content in coarse fragments allowed for any soil layer.
verbose	A logical value. Print the internal messages of the function?
...	Additional parameters to function spwb.

Details

The function performs a model inversion based on an ecohydrological assumption, consisting in that forest leaf area index is in equilibrium with a low embolism rate under normal conditions. This is translated in that the (by default 90%) interannual quantile of the maximum annual percent loss of conductance (PLC), averaged over plant cohorts, should be close to a target PLC value (by default 12%).

The algorithm first determines the PLC corresponding to the minimum and maximum soil extractable water (SEW). The minimum SEW (SEW_min) is an input parameter, whereas the maximum SEW (SEW_max) corresponds to no rock fragments in the soil.

Then three situations are distinguished:

1. If `PLC(SEW_min)` < `qPLC_target` and `PLC(SEW_max)` < `qPLC_target`, the function will use `uniroot` to find the root of the function `f(x)` = `PLC(x)` - `qPLC_target`, where `x` is SEW, which corresponds to a factor that multiplies the original rock fragment content.
2. If both `PLC(SEW_min)` < `qPLC_target` and `PLC(SEW_max)` < `qPLC_target`, the function cannot find an optimum, because PLC is always too low, and will return the original rock fragment content
3. Analogously, if both `PLC(SEW_min)` > `qPLC_target` and `PLC(SEW_max)` > `qPLC_target`, the function cannot find an optimum, because PLC is always too large, and will return the original rock fragment content

Value

Function `utils_rockOptimization` returns a list containing:

- `RFC`: A vector with the estimated rock fragment content for each soil layer.
- `SEW`: Soil extractable water (mm).
- `runs`: Number of simulations performed.
- `message`: Text message indicating whether optimization could be done (OK) or not.

Author(s)

Arsène Druel, URFM-INRAE

Nicolas Martin-StPaul, URFM-INRAE

Miquel De Cáceres Ainsa, CREAF

References

Druel, A., Martins, N., Cochard, H., De Caceres, M., Delzon, S., Mencuccini, M., Torres-Ruiz, J., and Ruffault, J.: European forest vulnerability to hydraulic failure: an ecohydrological approach, EGU General Assembly 2023, Vienna, Austria, 24–28 Apr 2023, EGU23-17068, https://doi.org/10.5194/egusphere-egu23-17068, 2023.

See Also

`spwb`, `soil`, `utils_ldrOptimization`

Examples

```
#Load example daily meteorological data
data(examplemeteo)

#Load example plot plant data
data(exampleforest)

#Default species parameterization
data(SpParamsMED)

#Initialize soil with two layers
examplesoil <- defaultSoilParams(4)

#Rock fragment content optimization (Granier)
utils_rockOptimization(exampleforest, soil = examplesoil,
                       SpParams = SpParamsMED, meteo = examplemeteo,
                       control = defaultControl("Granier"),
                       elevation = 100, latitude = 41.82592)
```

`waterUseEfficiency` *Water use efficiency*

Description

Calculates plant water use efficiency (WUE), at different temporal scales, from simulation results.

Usage

```
waterUseEfficiency(
  x,
  type = "Plant Ag/E",
  leaves = "average",
  freq = "days",
  draw = TRUE,
  ylim = NULL
)
```

Arguments

`x`	An object of class `spwb`, `pwb`, `growth` or `fordyn`.
`type`	A string to indicate the scale of WUE calculation. Either: • `"Leaf iWUE"`: Leaf intrinsic WUE, i.e. instantaneous ratio between photosynthesis and stomatal conductance (only for simulations with `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"` and `subdailyResults = TRUE`). • `"Leaf Ci"`: Leaf intercellular CO2 concentration (only for simulations with `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"` and `subdailyResults = TRUE`). • `"Plant An/E"`: Plant (cohort) net photosynthesis over plant transpiration (only for simulations with `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"`) • `"Stand An/E"`: Stand net photosynthesis over stand transpiration (only for simulations with `transpirationMode = "Sperry"` or `transpirationMode = "Sureau"`) • `"Plant Ag/E"`: Plant (cohort) gross photosynthesis over plant transpiration • `"Stand Ag/E"`: Stand gross photosynthesis over stand transpiration
`leaves`	Either `"sunlit"`, `"shade"` or `"average"`. Refers to the WUE of different leaf types or the average (with weights according to the LAI of sunlit and shade leaves). Only relevant for `type = "iWUE"`.
`freq`	Frequency of summary statistics (see `cut.Date`).
`draw`	A boolean flag to indicate that a plot should be returned.
`ylim`	Range of values for y.

Details

Temporal aggregation of WUE values is done differently depending on the value of `type`. For `type = "Plant Ag/E"`, `type = "Stand Ag/E"`, `type = "Plant An/E"` and `type = "Stand An/E"` sums or daily photosynthesis and transpiration are first calculated at the desired temporal scale and the ratio is calculated afterwards. For `type = "Leaf iWUE"` intrinsic WUE values are first calculated at the daily scale (as averages of instantaneous An/gs ratios weighted by An) and then they are aggregated to the desired scale by calculating weighted averages, where weights are given by daily photosynthesis.

Value

If `draw=TRUE` a plot is returned. Otherwise, the function returns a matrix with WUE values, where rows are dates (at the desired temporal scale), and columns are plant cohorts. In the case of `type = "Plant Ag/E"`, `type = "Stand Ag/E"`, `type = "Plant An/E"` and `type = "Stand An/E"` values are in gC/L. In the case of `type = "Leaf iWUE"` values are in micromol of carbon per mmol of water.

Author(s)

Miquel De Cáceres Ainsa, CREAF

See Also

`droughtStress`

Index