Package ‘sitree’

October 14, 2022

Version 0.1-13
Date 2022-04-28
Title Single Tree Simulator
Author Clara Anton Fernandez [aut, cre]
Maintainer Clara Anton Fernandez <caf@nibio.no>
Depends R (>= 3.5.0), ggplot2
Suggests rmarkdown, knitr
Imports data.table, methods
VignetteBuilder knitr
Description Framework to build an individual tree simulator.
License GPL (>= 2)
Encoding UTF-8
LazyLoad yes
LazyData yes
NeedsCompilation no
Repository CRAN
Date/Publication 2022-04-28 13:40:02 UTC

R topics documented:
calculate.development.class ........................................ 2
dbh.BN2009 ................................................................. 3
dead.trees.growth ....................................................... 4
fl ................................................................. 5
fn.vars.required ..................................................... 6
growdbhinc.hgtinc .................................................... 7
height.korf ............................................................... 9
management.prob ..................................................... 10
mng.tree.removal ..................................................... 12
mort.B2007 ............................................................ 14
**calculate.development.class**

*Calculate Development Class of the Stand*

**Description**

Calculates the development class of the stand according to the Norwegian National Forest Inventory.

**Usage**

```r
calculate.development.class(SI.spp, SI.m, stand.age.years)
```

**Arguments**

- **SI.spp**: Species for which the SI is calculated (1 = spruce, 2 = pine, 3 = deciduous).
- **SI.m**: Site index (SI) in m.
- **stand.age.years**: Age of the stand in years.

**Details**

Development classes 1 (regeneration) to 5 (mature).

**Value**

It returns a vector with the development class of the stands (values 1 to 5).
**Author(s)**

Clara Anton Fernandez <caf@nibio.no>

**Examples**

```r
calculate.development.class (SI.spp = c(1, 1, 2),
   SI.m = c(23,11,11),
   stand.age.years = c(120, 80, 40))
```

---

**dbhi.BN2009**  
*DBH Increment Function by Bollandsas and Naesset (2009)*

**Description**

Implementation of the diameter at breast height function published by Bollandsås and Næsset (2009).

**Usage**

```r
dbhi.BN2009(tr, fl, common.vars, this.period, ...)
```

**Arguments**

- `tr`  
  A `trList` class object.

- `fl`  
  A list describing the plot data.

- `common.vars`  
  A list with at least variables `QMD.cm`, `i.stand`, and `SBA.m2.ha`.

- `this.period`  
  The period for which to calculate DBH increment.

- `...`  
  Ignored

**Value**

A vector with DBH increments in mm for all trees in `tr`.

**Author(s)**

Clara Anton Fernandez <caf@nibio.no>

**References**

dead.trees.growth

Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 + 1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)

tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("QMD.cm", "i.stand"),
  period.length = 5
)

dbhi.BN2009(tr.i, fl, common.vars$res, this.period = "t0",
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31))
```

dead.trees.growth

*Growth of Dead Trees*

**Description**

A very simple function to estimate the growth of dead trees between the last measurement and their death. Dead trees are assumed to die in the middle of the period. The growth before the death of the tree is assumed to be half of the growth for the whole period, both in terms of DBH and height.

**Usage**

```r
dead.trees.growth(dt, growth, mort, this.period)
```

**Arguments**

- `dt`: A list of trees with at least two elements: `dbh.mm` and `height.dm`.
- `growth`: A data frame with columns `dbh.inc.mm` and `hgt.inc.dm`.
mort A TRUE/FALSE vector with the predicted mortality for each living tree in this period.
this.period The period where the dead trees were last seen alive (last measured DBH).

Value

Returns a data frame of same length as dt with two elements:

- dbh.mm DBH in mm
- height.dm Height in dm

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

dead.trees.growth(
  dt = list(dbh.mm = data.frame (t0 = c(75, 90, 25, 24)),
           height.dm = data.frame(t0 = c(190, 210, 110, 90))),
  growth = data.frame(dbh.inc.mm = c(7, 9, 2, 1),
                      hgt.inc.dm = c(11, 12, 8, 10)),
  this.period = "t0"
)

f1

Plot Data

Description

Plot data

Usage

data(f1)

Format

The list contains the following elements:

- plot.id unique ID for each stand that corresponds to plot.id in the tr dataset
- SI.m Site index in m
- SI.spp Species code for which SI.m is measured
- prop.plot Proportion of the plot covered by this stand (0-full plot, 2-20%, 3-30%, 4-40%, ..., 8-80%)
- ha2plot multiplier to convert from "per ha" to "per plot", e.g. basal area per ha to basal area per plot
tree2ha multiplier to convert numbers "per tree" to "per ha", e.g. basal area per tree to basal area per ha
soil.depth. 1234 soil depth 1(shallow)-4(deep)
land.use land use classification
land.type land type classification
veg.type vegetation type
subplot.size.m2 size of the subplot correponding to the stand in square meters
plot.size.m2 Plot size in square meters
kom Municipality code
region region where the plot lies
skidding.distance.100m skidding distance in 100m
slope.per slope in percentatge
lat.det latitude in degrees
alt.m altitude in meters
utm.s33.ov.m UTM E-W coordinates of the plot
utm.s33.snv.m UTM S-N coordinates of the plot
stand.age.years data frame with stand age in years, only the first column, t0, is filled up
management data frame will management codes for each period

Source
This dataset is derived from the Norwegian National Forest Inventory

Examples
data(fl)
str(fl)

---

fn.vars.required Finds Variables Required on Functions

Description
Tries to find the variables to be calculated through the parameters of the functions.

Usage
fn.vars.required(my.functions, ...)

Arguments
my.functions A list of functions to be used during the simulation
... Use this to add any function that might not be included in my.functions
Value

It returns a vector with the name of the potential variables required.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

```r
fn.vars.required (my.functions = list(
    fn.growth = "grow.dbhinc.hgtinc",
    fn.dbh.inc = "dbh.BN2009",
    fn.hgt.inc = "height.korf",
    fn.mort = "mort.B2007",
    fn.recr = "recr.BBG2008",
    fn.management = "management.prob",
    fn.tree.remove = "mng.tree.remove",
    fn.modif = NULL,
    fn.prep.common.vars = "prep.common.vars.fun",
),
    "BBG2008")
```

---

grow.dbhinc.hgtinc Growth and Height Increment Function Wrapper

description

A growth and height increment function wrapper that can be used with or without a height increment function.

Usage

grow.dbhinc.hgtinc(tr, fl, common.vars, this.period, functions, ...)

Arguments

- `tr` A `trList` class object.
- `fl` A list describing the plot data.
- `common.vars` A list with at least variables spp (species classification in categories: spruce, pine, birch and other), and SBA.m2.ha.
- `this.period` The period for which to calculate growth.
- `functions` A list defining the functions to be used in sitree.
- `...` Further arguments to be passed to DBH increment and height increment functions.
Details

A growth function should calculate both dbh and height increment. This function calls two functions, defined in functions as fn.dbh.inc and fn.hgt.inc to estimate those.

Value

A data frame with two elements dbh.inc.mm DBH increment in mm hgt.inc.dm Height increment in dm

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
    plot.id = tr$plot.id,
    treeid = tr$treeid,
    dbh.mm = foo.dbh,
    height.dm = foo.height,
    yrs.sim = rep(0, nrow(tr)),
    tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
    tr = tr.i,
    fl = fl,
    i.period = 0,
    this.period = "t0",
    common.vars = "NULL",
    vars.required = c("spp", "SBA.m2.ha", "QMD.cm"),
    period.length = 5,
    species.spruce = c(1, 2, 3),
    species.pine = c(10, 11, 20, 21, 29),
    species.harw = c(30, 31)
)

grow.dbhinc.hgtinc (tr = tr.i,
    fl = fl,
    common.vars = common.vars$res,
    this.period = "t0",
    functions = list(
        fn.growth = 'grow.dbhinc.hgtinc',
        fn.mort = 'mort.B2007',
    )
)
height.korf

A simple tree height function based on tree species, and DBH. It returns the difference between the height at the current period (this.period) and the next period.

Usage

height.korf(common.vars, this.period, tr, dbh.inc.mm, ...)

Arguments

  common.vars        A list with at least an element named 'spp' with the species group composition.
  this.period        The period for which to calculate the height of the trees.
  tr                 A trList.
  dbh.inc.mm         A vector with the projected dbh increment.
  ...                Currently not used

Value

A vector with tree heights increments in dm.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  this.period = "t0",
  i.period = 0,
  common.vars = "NULL",
  mng.options = NA,
  vars.required = c("spp"),
  period.length = 5
)
height.korf(common.vars = common.vars$res, this.period = 't0',
  tr.i, fl, dbh.inc.mm = runif(nrow(tr.i$data$dbh.mm), 5, 80))
```

management.prob

Final felling and thinning functions for Norwegian forest

Description

Estimates de probability of a stand to be harvested or thinning following Anton-Fernandez et al. (20012).

Usage

```r
management.prob(tr, fl, fun.final.felling = "harv.prob",
  fun.thinning = "thin.prob", common.vars, this.period, next.period, ...)

harv.prob(region, skidding.distance.100m, AgeTo5, vuprha.m3.ha, slope.per, SI.m, SI.spp)

thin.prob(region, skidding.distance.100m, AgeTo5, vuprha.m3.ha, slope.per, SI.m, SI.spp)
```
Arguments

tr       A trList class object.
fl       A list describing the plot data.
fun.final.felling  Function to use to calculate final felling.
fun.thinning     Function to use to calculate thinning.
common.vars    A list with at least variables dev.class and vuprha.m3.ha.
this.period    The period for which to calculate final felling and thinning probability.
ext.period    The next period to the one for which final felling and thinning probability are to be calculated.
...                         
region       A vector containing the region in Norway where every plot is situated.
skidding.distance.100m   A vector containing skidding for each plot.
AgeTo5       A vector containing number of years to development class 5 for each plot.
vuprha.m3.ha  Volume per ha in cubic meters per ha for each plot.
slope.per    Slope, in percentage, for each plot.
SI.m         Site index (SI) in m.
SI.spp       Species for which the SI is calculated (1 = spruce, 2 = pine, 3 = deciduous).

Value

It returns a list with one element:

mng       a vector with the management to apply to each plot.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
    plot.id = tr$plot.id,
mng.tree.removal

Function to remove trees

treed = tr$treeid,
dbh.mm = foo.dbh,
height.dm = foo.height,
yrs.sim = rep(0, nrow(tr)),
tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("spp", "SBA.m2.ha", "QMD.cm", "vuprha.m3.ha", "AgeTo5"),
  period.length = 5
)
fl$management <- data.frame(matrix(NA, ncol = tr.i$nperiods,
  nrow = nrow(tr.i$data$dbh.mm)))
names(fl$management) <- paste0("t", 1:tr.i$nperiods)

management.prob(tr.i, fl,
  fun.final.felling = "harv.prob",
  fun.thinning = "thin.prob",
  common.vars = common.vars$res,
  this.period = "t0",
  next.period = "t1")

harv.prob(region = fl$region[1:3],
  skidding.distance.100m = fl$skidding.distance.100[1:3],
  AgeTo5 = c(50, 20, 15),
  vuprha.m3.ha = common.vars$res$vuprha.m3.ha[1:3],
  slope.per = fl$slope.per[1:3],
  SI.m = fl$SI.m[1:3],
  SI.spp = fl$SI.spp[1:3])

thin.prob(region = fl$region[1:3],
  skidding.distance.100m = fl$skidding.distance.100[1:3],
  AgeTo5 = c(50, 20, 15),
  vuprha.m3.ha = common.vars$res$vuprha.m3.ha[1:3],
  slope.per = fl$slope.per[1:3],
  SI.m = fl$SI.m[1:3],
  SI.spp = fl$SI.spp[1:3]
Description

Define how trees are going to be removed in plots with some sort of management that involves removal (e.g. final felling, thinning).

Usage

mng.tree.removal(tr, fl, common.vars, this.period, next.period, ...)

Arguments

- **tr**: A `trList` class object.
- **fl**: A list describing the plot data.
- **common.vars**: A list with at least variable `vol.wo.tr.m3.ha`
- **this.period**: The period for which to calculate final felling and thinning probability.
- **next.period**: The next period to the one for which final felling and thinning probability are to be calculated.

Value

Returns a TRUE/FALSE vector of length equal to the number of trees in `tr`.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = ")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
plot.id = tr$plot.id,
treeid = tr$treeid,
dbh.mm = foo.dbh,
height.dm = foo.height,
ys.sim = rep(0, nrow(tr)),
tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
tr = tr.i,
fl = fl,
i.period = 0,
this.period = "t0",
...
14

```r
common.vars = "NULL",
vars.required = c("vol.wo.tr.m3.ha", "vuprha.m3.h"),
period.length = 5,
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 28, 21, 29),
species.harw = c(30, 31)
)
fl$management <- data.frame(matrix(NA, ncol = tr.i$nperiods,
nrow = length(fl$plot.id)))
names(fl$management) <- paste0("t", 1:tr.i$nperiods)
management <- management.prob(tr.i,
  fl,
  fun.final.felling = "harv.prob",
  fun.thinning = "thin.prob",
  common.vars = common.vars$res,
  this.period = "t0",
  next.period = "t1")
fl$management[, "t1"] <- management$management
remove.tree <- mng.tree.removal(tr = tr.i,
  fl = fl,
  common.vars = common.vars$res,
  this.period = "t0",
  next.period = "t1",
  per.vol.harv = 0.83 )
sum(remove.tree)
```

---

**mort.B2007**

*Mortality Function by Bollandsas (2007)*

---

**Description**


**Usage**

`mort.B2007(tr, fl, common.vars, this.period, ...)`

**Arguments**

- `tr` A `trList` class object.
- `fl` A list describing the plot data.
- `common.vars` A list with at least variables `spp` (species classification in categories: spruce, pine, birch and other), and `SBA.m2.ha`.
- `this.period` The period for which to calculate mortality.
- `...` Ignored.
Value

A TRUE/FALSE vector for all trees in tr.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 + 1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  this.period = "t0",
  i.period = 0,
  common.vars = "NULL",
  mng.options = NA,
  vars.required = c("spp", "SBA.m2.ha"),
  period.length = 5
)

dead.trees <- mort.B2007(tr.i, fl, common.vars$res, this.period = 't0')
sum(dead.trees)
```
Description

These functions calculate tree volume with or without bark for the main species in Norway, that is, Norway spruce, Scots pine, Sitka spruce, birch, and ... following Braastad (1966), Brantseg (1967), and Vestjordet (1967).

Usage

```r
picea.abies.vol(dbh, trh, bark, units)
picea.abies.volV(dbh, trh, bark, units)
pinus.sylvestris.vol(dbh, trh, bark, units)
pinus.sylvestris.volV(dbh, trh, bark, units)
sitka.vol(dbh, trh, bark, units)
harw.nor.vol(tsl, dbh, trh, bark, units)
```

Arguments

- `dbh`: A vector with diameters at breast height in mm with bark.
- `trh`: A vector with tree heights in cm.
- `bark`: If tree volume should be calculated with bark, "mb", or without bark, "ub".
- `units`: "l" in liters, "c" in cubic centimeters.
- `tsl`: Tree species 30 (), 31(), 32 (), 40(), 41(), 50(), NA. This only affects the calculations of diameter without bark.

Value

Returns tree volume in the selected units.

References


Examples

```r
harw.nor.vol(tsl = c(30,31), dbh = c(45, 25), trh = c(120, 80), bark = "ub", units = "l")
```
prep.common.vars.fun  Function to Calculate Common Variables

Description

A function to calculate common variables needed to estimate growth, death, etc.

Usage

prep.common.vars.fun(tr, fl, i.period, this.period, common.vars, vars.required, period.length, n.periods, ...)

Arguments

- **tr**: A `trList` class object.
- **fl**: A list describing the plot data.
- **i.period**: The number of period (from 0 to n.periods) for which to calculate the common variables.
- **this.period**: The period for which to calculate the common variables.
- **common.vars**: A list with existing common variables.
- **vars.required**: Variables to be calculated.
- **period.length**: Length of the periods, in years.
- **n.periods**: Number of periods.
- **...**

Details

This is an example of a prep.common.vars file. A prep.common.vars (prepare common variables) is needed for the simulation. It is in this file that additional variables needed for the particular functions used in the simulation are calculated. For example, to calculate quadratic mean diameter, basal area, or volume.

Value

A list

- **res**: A list with the common variables.
- **fl**: Updated `fl`

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
 dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  i.period = 0,
  this.period = "t0",
  common.vars = "NULL",
  vars.required = c("vol.wo.tr.m3.ha", "vuprha.m3.h"),
  period.length = 5
)
str(common.vars)
```

QMD.cm.f

**Quadratic Mean Diameter**

<table>
<thead>
<tr>
<th>QMD.cm.f</th>
<th>Quadratic Mean Diameter</th>
</tr>
</thead>
</table>

**Description**

Calculates quadratic mean diameter.

**Usage**

```r
QMD.cm.f(x.mm)
```

**Arguments**

- `x.mm` DBH in mm.

**Details**

It ignores not finite values like NA, and calculates the QMD for the rest of the trees.

**Value**

The QMD in cm of x.mm.
Author(s)

Clara Anton Fernandez <caf@nibio.no>

Examples

QMD.cm.f(50)
QMD.cm.f(c(51, NA))
QMD.cm.f(c(51, 42, 120, 57))

recover.last.measurement

Recover dbh and height at time of death

Description

Removes all dbh and height data that corresponds to the live of the tree, and adds the dbh and height at time of death to the dbh.mm and height.dm in the data field.

Usage

recover.last.measurement(tr.list.dead)

Arguments

tr.list.dead a trListDead object

Details

It removes all dbh and height data from the data field, adds the dbh and height at time of death, and converts any potential NAs created in the process to 0.

Value

It returns the same object (a trListDead object) with all data on dbh and height removed. For each tree there is only one observation of dbh and height in the data field, the estimated dbh and height at time of death.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
set.seed(2017)
res <- sitree (tree.df = tr,
   stand.df = fl,
   functions = list(
      fn.growth = 'grow.dbhinc.hgtinc',
      fn.mort = 'mort.B2007',
      fn.recr = 'recr.BBG2008',
      fn.management = 'management.prob',
      fn.tree.removal = 'mng.tree.removal',
      fn.modif = NULL, '#ext.modif.fun',
      fn.prep.common.vars = 'prep.common.vars.fun'
   ),
   n.periods = 5,
   period.length = 5,
   mng.options = NA,
   print.comments = FALSE,
   fn.dbh.inc = "dbh.BN2009",
   fn.hgt.inc = "height.korf",
   species.spruce = c(1, 2, 3),
   species.pine = c(10, 11, 20, 21, 29),
   species.harw = c(30, 31),
   fun.final.felling = "harv.prob",
   fun.thinning = "thin.prob",
   'BN2009',
   'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
   per.vol.harv = 0.83)
)
dead <- recover.last.measurement(res$dead)
str(dead)
removed <- recover.last.measurement(res$removed)
str(removed)
```

recover.state

Recovers the State

Description

It adds dead and removed trees to `tr`. This is an internal function that should not be run, in general, by a regular user.

Usage

```
recover.state(tr, dead.trees, removed.trees)
```

Arguments

- `tr` A `trList` class object.
- `dead.trees` A `trListDead` class object.
- `removed.trees` A `trListDead` class object.
Details
In order to make calculations easier and lighter memory wise, at each cycle (period) of the simulation all removed (harvested) and dead trees are removed from \( tr \), which only keeps alive trees at the current period. Once the simulation is finished `recover.state` returns all dead and removed trees to the last \( tr \); so `sitree` can return a full list, including all trees alive at any point during the simulation, of trees.

Value
A `trList` class object.

Author(s)
Clara Anton Fernandez <caf@nibio.no>

---

**recr.BBG2008**  
*Recruitment function following BBG2008*

**Description**
It creates a list of new trees based on the functions from BBG2008, Bollandsås et al. (2008) doi:10.1080/02827580801995315. BBG2008 functions estimates recruitment on a plot based on stand basal area, SI, and pine, spruce, birch, and other broadleaved species abundance. The implementation of this function is stochastic, not deterministic. So, if run several times without setting a seed, the results will vary.

**Usage**
```
recr.BBG2008(tr, fl, common.vars, i.period, this.period, next.period, ...)
```

**Arguments**
- **tr**: A `trList` object
- **fl**: A list describing the plot information
- **common.vars**: The number of period (from 0 to nperiods) for which to calculate recruitment.
- **i.period**: Period to use to estimate next period recruitment.
- **this.period**: Period for which to calculate recruitment.
- **next.period**: Period for which to calculate recruitment.
- **SBA.m2.ha**: Stand basal area in m2 per ha.
- **SI.m**: Site index (SI) in m.
- **pr.spru.ba**: Percentage of spruce in terms of basal area.
- **pr.pine.ba**: Percentage of pine in terms of basal area.
- **pr.birch.ba**: Percentage of birch in terms of basal area.
- **pr.other.ba**: Percentage of other species in terms of basal area.
- **...**: Not used.
Value

Returns a data frame with two columns for each of the four species, one for the probability of positive recruitment (spruce.p, pine.p, birch.p, other.p), and one for the conditional expected recruitment (spruce.e, pine.e, birch.e, other.e).

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

```r
foo.matrix <- matrix(0, nrow = length(tr$dbh), ncol = (5 +1))
colnames(foo.matrix) <- paste("t", 0:5, sep = "")
foo.dbh <- foo.height <- foo.matrix
foo.dbh[,1] <- tr$dbh
foo.height[,1] <- tr$height

trl <- list(
  plot.id = tr$plot.id,
  treeid = tr$treeid,
  dbh.mm = foo.dbh,
  height.dm = foo.height,
  yrs.sim = rep(0, nrow(tr)),
  tree.sp = factor(tr$tree.sp)
)
tr.i <- trList$new(data = trl, nperiods = as.integer(5))

## create a fake dead trees and removed trees.
common.vars <- prep.common.vars.fun(
  tr = tr.i,
  fl = fl,
  this.period = "t0",
  i.period = 0,
  common.vars = "NULL",
  mng.options = NA,
  vars.required = c("spp", "SBA.m2.ha", "pr.spru.ba"),
  period.length = 5,
  species.spruce = c(1, 2, 3),
  species.pine = c(10, 11, 20, 21, 29),
  species.harw = c(30, 31)
)
recr <- recr.BBG2008(tr.i, fl, common.vars$res, i.period = 0, this.period = "t0",
                      next.period = "t1")
```
str(recr)

bbg <- BBG2008
  (common.vars$res$SBA.m2.ha[common.vars$res$i.tree],
   f1$SI.m,
   common.vars$res$spp.ba$spru[common.vars$res$i.tree],
   common.vars$res$spp.ba$pine[common.vars$res$i.tree],
   common.vars$res$spp.ba$birch[common.vars$res$i.tree],
   common.vars$res$spp.ba$other[common.vars$res$i.tree])

str(bbg)

---

**sitree**

*Individual Tree Simulator*

**Description**

Simulates the growth of individual trees in several plots, including management and keeping record of dead and removed trees. It can also include the effect of external modifiers such as climate change. The SiTree package is described in Antón-Fernández, C., & Astrup, R. (2022). https://doi.org/10.1016/j.softx.2021.100925

**Usage**

```r
sitree(tree.df, stand.df, functions, n.periods = 5, period.length,
  mng.options = NA, print.comments = FALSE,..., ext.mod.first = TRUE)
```

**Arguments**

- `tree.df`: A data frame with individual tree data. It should include four columns named 'plot.id' (an ID for the plot/stand that the tree belongs to), treeid (a unique ID for the tree), dbh (a measure of horizontal dimension), height (a measure of vertical dimension), tree.sp (tree species).
- `stand.df`: A data frame or list with necessary data on the stand (e.g. slope, site index). It should include a 'plot.id' that matches the one in tree.df and a 'plot.size.m2', the plot size in m2. 'plot.size.m2' is only used in the summary.sitree function to calculate figures by ha.
- `functions`: A list describing the basic functions. It should include a minimum of a function for growth (fn.growth), for diameter (or basal area) increment, for height increment (height.korf), mortality (fn.mort), recruitment (fn.recr), management (fn.management), tree removal (fn.tree.removal), and external modifiers (fn.modif). fn.management, fn.tree.removal, and fn.modif, can be missing.
- `n.periods`: Number of periods to simulate (numeric).
- `period.length`: The length of the periods (e.g. 5 years).
- `mng.options`: Management options and definitions to be passed further.
- `print.comments`: TRUE/FALSE Print comments. It is meant to help debugging new functions.
- `...`: Further arguments to pass to functions used during simulation.
- `ext.mod.first`: If TRUE (default) the external modifier functions will be run before fn.prep.common.vars.
Value

Returns a list with all live, dead, removed trees, and some plot information like stand age in years at each period, and management applied at each plot at each period.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

References


Examples

```r
result.sitree <- sitree (tree.df = tr, 
                      stand.df = fl, 
                      functions = list(
                        fn.growth = 'growdbhinc.hgtinc',
                        fn.mort = 'mort.B2007',
                        fn.recr = 'recr.BBG2008',
                        fn.management = 'management.prob',
                        fn.tree.removal = 'mng.tree.removal',
                        fn.modif = NULL, 
                        fn.prep.common.vars = 'prep.common.vars.fun'
                      ),
                      n.periods = 5,
                      period.length = 5,
                      mng.options = NA, 
                      print.comments = FALSE, 
                      fn.dbh.inc = "dbhinc.BN2009", 
                      fn.hgt.inc = "height.korf", 
                      fun.final.felling = "harv.prob", 
                      fun.thinning = "thin.prob", 
                      per.vol.harv = 0.83 
                      
                      
                      )
str(result.sitree)
set.seed(2017)
result.sitree <- sitree (tree.df = stand.west.tr, 
                      stand.df = stand.west.st, 
                      functions = list(
                        fn.growth = 'growdbhinc.hgtinc',
                        fn.mort = 'mort.B2007',
                        fn.recr = 'recr.BBG2008',
                        fn_management = NULL,
                        fn.tree.removal = NULL,
                        fn.modif = NULL,
                        fn.prep.common.vars = 'prep.common.vars.fun'
                      ),
                      n.periods = 20,
                      period.length = 5,
                      mng.options = NA,
                      )
```
sitree.summary

print.comments = FALSE,
fn.dbh.inc = "dbhi.BN2009",
fn.hgt.inc = "height.korf"
)

sitree.summary  Summary plots

Description

It produces 5 summary plots (stand basal area, number of stems, average height of tallest 10 trees, number of dead trees, and number of harvested trees)

Usage

sitree.summary(sitrees.res, plots, by.stand = TRUE, plot = FALSE,
plot.all.together = FALSE, with.legend = FALSE)

Arguments

sitrees.res  An object result of running sitree().
plots        Plots to be printed (1: stand basal area, 2: number of stems, 3: average height of tallest 10 trees, 4: number of dead trees, 5: number of harvested trees)
by.stand     TRUE/FALSE TRUE prints plot by stand (figures by ha), if FALSE it prints plots with totals (divides values per ha by 'plot.size' and multiplies by 10.000 to obtain total values)
plot         TRUE/FALSE if plots are printed
plot.all.together  if TRUE all plots are plotted in a single page
with.legend   if TRUE a legend will be added to the plot when by.stand == TRUE

Value

It passes the required plots in a named list. It plot is TRUE it also prints the required plots.

Author(s)

Clara Anton Fernandez <caf@nibio.no>

See Also

sitree
Examples

```r
require(sitree)
res <- sitree (tree.df = tr,
stand.df = fl,
functions = list(
    fn.growth = "grow.dbhinc.hgtinc",
    fn.mort = "mort.B2007",
    fn.recr = "recr.BBG2008",
    fn.management = "management.prob",
    fn.tree.removal = "mng.tree.removal",
    fn.modif = NULL,
    fn.prep.common.vars = "prep.common.vars.fun"
),
n.periods = 5,
period.length = 5,
mng.options = NA,
print.comments = FALSE,
fn.dbh.inc = "dbhi.BN2009",
fn.hgt.inc = "height.korf",
species.spruce = c(1, 2, 3),
species.pine = c(10, 11, 20, 21, 29),
species.harw = c(30, 31),
fun.final.felling = "harv.prob",
fun.thinning = "thin.prob",
per.vol.harv = 0.83
)

summary.plots <- sitree.summary(res, plots = 1:6, by.stand = FALSE, plot = TRUE)
```

Description

sitree2dataframe converts trList and trListDead class objects to data.frames. sitree2dataframe converts all output of sitree to a data.frame

Usage

```r
sitree2dataframe(tr.dt)
sitree2dataframe.all(sitree.res)
```

Arguments

- `tr.dt`: A trList or trListDead object.
- `sitree.res`: An object resulting from running sitree.
Details

As implemented, it expects to have a 'removed' element on sitree2dataframe.all. For an alternative when the simulation has no harvest see the examples below.

Value

sitree2dataframe returns a data.frame with dbh, height, and species by period. sitree2dataframe.all returns a list

- **live**: A data.frame with dbh, height, and species of the trees alive at each period.
- **dead**: A data.frame with dbh, height, and species of the trees found dead at each period.
- **removed**: A data.frame with dbh, height, and species of the trees removed at each period.

Author(s)

Clara Anton Fernandez

Examples

```r
n.periods <- 10

# Run a simulation with the example functions
res <- sitree (tree.df = tr,
               stand.df = fl,
               functions = list(
                 fn.growth = "Var.grow.dbhinc.hgtinc",
                 fn.mort = "mort.B2007",
                 fn.recr = "recr.BBG2008",
                 fn.management = "management.prob",
                 fn.tree.removal = "mng.tree.removal",
                 fn.modif = NULL,
                 fn.prep.common.vars = 'prep.common.vars.fun'
               ),
               n.periods = n.periods,
               period.length = 5,
               mng.options = NA,
               print.comments = FALSE,
               fn.dbh.inc = "dbh.BN2009",
               fn.hgt.inc = "height.korf",
               species.spruce = c(1, 2, 3),
               species.pine = c(10, 11, 20, 21, 29),
               species.harw = c(30, 31),
               fn.final.felling = "harv.prob",
               fn.thinning = "thin.prob",
               'BN2009',
               'BBG2008', 'SBA.m2.ha', 'spp', 'pr.spru.ba', 'QMD.cm',
               per.vol.harv = 0.83
             )

dbh.mm <- sitree2dataframe(res$live)
```
res.dt <- sitree2dataframe.all(res)

## The function is currently defined as

---

**sp.classification**  
Species classification

### Description

Classifies the tree species into four species groups

### Usage

sp.classification(tree.sp, species.spruce, species.pine, species.harw)

### Arguments

- **tree.sp**: Tree species code.
- **species.spruce**: A vector with the codes used to define 'spruce' group.
- **species.pine**: A vector with the codes used to define 'pine' group.
- **species.harw**: A vector with the codes used to define the 'birch' group.

### Value

Returns a vector with the classification of tree.sp into one of these four groups: spruce, pine, birch, and other. spruce group is defined by species.spruce, pine group is defined by species.pine, and birch is defined by species.harw. If a species does not fall within any of these three groups is classified as 'other'.

### Author(s)

Clara Anton Fernandez <caf@nibio.no>

### Examples

sp.classification(tr$tree.sp, species.spruce = c(1, 15),
                  species.pine = c(2, 34),
                  species.harw = c(3, 4))
**stand.west.st**  
*Stand and plot characteristics for stand.west.tr*

**Description**
Dataset with stand and plot data for stand.west.tr

**Usage**
stand.west.st

**Format**
A data frame with 4 observations on the following 7 variables.

- `plot.id` unique ID for each plot
- `SI.m` site index in meters
- `kom` municipality code
- `SI.spp` species for which site index is measured
- `plot.size.m2` plot size in m²
- `tree2ha` multiplier to convert values per tree to per hectare
- `lat.deg` latitude of the plots in degrees
- `subplot.size.m2` plot size in m², this is a dummy variable needed for some of the calculations

**Examples**

```r
stand.west.st
str(stand.west.st)
```

---

**stand.west.tr**  
*A whole stand dataset*

**Description**
This dataset describes a whole stand of 1019 m² and spacing 1.25x1.25 m. The stand is divided in 4 plots of approximately 250 m². All trees are Norway spruce.

**Usage**

stand.west.tr
Format

A data frame with 651 observations on the following 5 variables.

- **dbh**  diameter at breast height in mm
- **height**  tree height in dm
- **plot.id**  a unique ID for each plot
- **treeid**  a unique ID for each tree
- **tree.sp**  tree species, a factor with levels 1 2 3 10 11 20 21 29 30 31 32 40 41 42 43 44 48 49 50 51 52 53 54 55 56 57 58 59 70

Examples

```r
stand.west.tr
str(stand.west.tr)
```

---

**toBindLists**

**Binding lists**

Description

This function is only used internally to bind vectors, data frames, and lists.

Usage

```r
toBindLists(x, y)
```

Arguments

- **x**  A vector, data frame or list.
- **y**  Object of the same type as x.

Value

It returns a vector, a data frame or a list depending on x.

Author(s)

Clara Anton Fernandez <caf@nibio.no>
Examples

```r
## Example of how toBindLists works with data frames
x <- data.frame(1:10, ncol = 2)
names(x) <- c('a', 'b')
y <- data.frame(1:20, ncol = 2)
names(y) <- names(x)
j <- toBindLists(x, y)
j

## with vectors
x <- c(1:3)
y <- (8:15)
j <- toBindLists(x, y)
j

## with lists
x <- data.frame(1:10, ncol = 2)
names(x) <- c('a', 'b')
y <- data.frame(1:20, ncol = 2)
names(y) <- names(x)
x <- list(x = x, y = y)

xx <- data.frame(1:10, ncol = 3)
names(xx) <- c('a', 'b')
yy <- data.frame(1:20, ncol = 3)
names(yy) <- names(xx)
y <- list(xx = xx, yy = yy)
j <- toBindLists(x, y)
j
```

---

**tr**  
**Individual Tree Data**

**Description**  
A data frame with data to create the trList object needed to run `sitree`.

**Usage**  
data("tr")

**Format**  
The data frame contains the following columns

- **plot.id** unique ID for each stand that corresponds to plot.id in the `fl` dataset
trList-class

- **treeid**: unique ID for each tree
- **dbh**: dbh in mm
- **height**: height of the tree in dm
- **tree.sp**: tree species code

**Source**

This dataset is derived from the Norwegian National Forest Inventory.

**Examples**

```r
data(tr)
hist(tr$dbh)
```

---

**Description**

Describes the Reference Class `trList`, which is used to store data on live trees.

**Extends**

All reference classes extend and inherit methods from "envRefClass".

**Fields**

- **data**: Object of class list containing the unique stand ID (plot.id), the tree ID (treeid), and dbh (dbh.mm) and height (height.dm) for all the periods.
- **nperiods**: Object of class integer containing the number of period that should be simulated.

**Methods**

- **extractTrees(i)**: Removes trees from the objectd, for example when trees die or are harvested.
- **as.list()**: converts this class to a list
- **show()**: shows the first 20 trees
- **getTrees(i, j)**: get the information for i trees and j periods without deleting the data
- **addTrees(value)**: adds trees to the object

**Note**

`trList` objects should not be modified inside any user function. Any change should be made by the `sitree` function. Reference class objects are mutable, they don’t use R’s usual copy-on-modify semantics, but are modified in place. So, if you want to modify, for example for a posterior analysis of the results, any `trList` object I recommend to make a copy of the object first, using `$copy()`, e.g. `my.tr.list$copy()`.
Author(s)
Clara Anton Fernandez <caf@nibio.no>

See Also
trListDead

Examples
showClass("trList")

data: Object of class list containing the unique stand ID, the tree ID, and dbh and height for all
the periods. Only data from the periods were the tree was alive is stored in this field

nperiods: Object of class integer containing the number of period that should be simulated

last.measurement: Object of class data.frame containing tree ID, dbh, height, and period when
the tree was "found" dead

Methods

last.time.alive(): It estimates when the trees where last "seen" alive

addTrees(value): It add trees to the trlistDead object

remove.next.period(next.period): It removed the data from the period when the tree was seen
"dead". The next period is calculated so the dbh and height at time of death can be estimated

The following methods are inherited (from the corresponding class): extractTrees ("trList"), as.list
("trList"), show ("trList"), getTrees ("trList"), addTrees ("trList")

Note
This class is meant to keep the data of the dead trees in a similar way as the live trees, but with some
extra information such as diameter and height at death.
Author(s)
Clara Anton Fernandez <caf@nibio.no>

See Also
trList.

Examples
showClass("trListDead")

table

volume.norway

Tree volume for Norwegian conditions

Description
Calculates tree volume following the equations used in the Norwegian national forest inventory

Usage
volume.norway(dbh.mm, height.dm, tree.sp, kom, vol.reduksjon = NULL,
vol.w = TRUE, vol.wo = TRUE)

Arguments
dbh.mm tree dbh in mm
height.dm tree height in dm
tree.sp tree species following the same codification as the Norwegian NFI
kom municipality code
vol.reduksjon volume reduction in 100
vol.w TRUE/FALSE if volume with bark needs to be calculated
vol.wo TRUE/FALSE if volume without bark needs to be calculated

Value
It returns a list with up to two elements:
vol.w.tr.m3 volume with bark per tree in m3
vol.wo.tr.m3 volume without bark per tree in m3

Examples
volume.norway(dbh.mm = c(50, 70), height.dm = c(17, 20), tree.sp = c(1, 10), kom = c(623, 623))
Index

* classes
  trList-class, 32
  trListDead-class, 33

* datasets
  fl, 5
  stand.west.st, 29
  stand.west.tr, 29
  tr, 31

BBG2008 (recr.BBG2008), 21

calculate.development.class, 2

dbhi.BN2009, 3
dead.trees.growth, 4

eenvRefClass, 32, 33

f1, 5
fn.vars.required, 6

grow.dbhinc.hgtinc, 7

harv.prob (management.prob), 10

harw.nor.vol (picea.abies.vol), 15

height.korf, 9

management.prob, 10

mng.tree.removal, 12

mort.B2007, 14

picea.abies.vol, 15
picea.abies.volV (picea.abies.vol), 15
pinus.sylvestris.vol (picea.abies.vol), 15

pinus.sylvestris.volV (picea.abies.vol), 15

prep.common.vars.fun, 17

QMD.cm.f, 18

recover.last.measurement, 19

recover.state, 20

recr.BBG2008, 21

sitka.vol (picea.abies.vol), 15

sitree, 23, 25

sitree.summary, 25

sitree2dataframe, 26

sp.classification, 28

stand.west.st, 29

stand.west.tr, 29

thin.prob (management.prob), 10

toBindLists, 30

tr, 31

trList, 33, 34

trList (trList-class), 32

trList-class, 32

trListDead, 33

trListDead (trListDead-class), 33

trListDead-class, 33

volume.norway, 34