

Package: LandR (via r-universe)

March 12, 2025

Type Package

Title Landscape Ecosystem Modelling in R

Description Utilities for 'LandR' suite of landscape simulation models. These models simulate forest vegetation dynamics based on LANDIS-II, and incorporate fire and insect disturbance, as well as other important ecological processes. Models are implemented as 'SpaDES' modules.

URL <https://landr.predictiveecology.org>,
<https://github.com/PredictiveEcology/LandR>

Date 2025-03-12

Version 1.1.5.9038

Depends R (>= 4.2)

Imports crayon, data.table, fpCompare, ggplot2, ggpubr, ggspatial, grDevices, httr2, lme4, methods, MuMIn (>= 1.42.5), parallel, pemisc (>= 0.0.4.9011), quickPlot (>= 1.0.2.9001), raster, RColorBrewer, reproducible, sf, sp, SpaDES.tools (>= 2.0.4), stats, terra, tidyterra, tools, utils, viridis

Suggests bbmle, biomod2 (>= 4.2-2), BioSIM, covr, curl, dplyr, fasterize, future, future.apply, ggalluvial, ggrepel, googledrive (>= 2.0.0), httr, knitr, lqmm, map, MASS, memuse, parallelly, qs, rasterVis, RCurl, RhpcBLASctl, rlang, rmarkdown, roxygen2, R.utils, SpaDES.core, spelling, testthat, withr, XML

Remotes CeresBarros/biomod2@dev_currentCeres, RNCAN/BioSimClient_R, PredictiveEcology/map@development, PredictiveEcology/pemisc@development, PredictiveEcology/quickPlot@development, PredictiveEcology/reproducible@development, PredictiveEcology/SpaDES.core@development, PredictiveEcology/SpaDES.tools@development

Encoding UTF-8

Language en-CA

License GPL-3

VignetteBuilder knitr, rmarkdown

BugReports <https://github.com/PredictiveEcology/LandR/issues>

ByteCompile yes

LazyData true

Collate 'LANDIS-II-tables.R' 'LandR-package.R' 'assertions.R'
 'biomodWrappers.R' 'biosim.R' 'chooseName.R' 'cohorts.R'
 'convertUnwantedLCC2.R' 'dem.R' 'disturbances.R' 'ecoregions.R'
 'ecozones.R' 'fireSeverity.R' 'lccMapGenerator.R'
 'light-shade.R' 'makeDummyInputs.R' 'maps.R' 'maxBestimation.R'
 'misc.R' 'nullClimate.R' 'overlayLCCs.R' 'partitionBiomass.R'
 'plot-transitions.R' 'plot_summaries.R' 'plotting.R'
 'prepEcoregions.R' 'prepInputObjects.R' 'prepInputs_NTEMS.R'
 'prepSpeciesLayers.R' 'seedDispersalLANDIS.R'
 'serotiny-resprouting.R' 'speciesPresentFromKNN.R'
 'speciesTable.R' 'sppEquivalencies_CA.R' 'sppHarmonize.R'
 'studyArea.R' 'terra-raster-utils.R' 'wildfire.R' 'zzz.R'

RoxygenNote 7.3.2

Roxygen list(markdown = TRUE)

Config/testthat/edition 3

Config/testthat/parallel true

Config/testthat/start-first prep*

Config/pak/sysreqs cmake libgdal-dev gdal-bin libgeos-dev make
 default-jdk libicu-dev libjpeg-dev libpng-dev libssl-dev
 libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://predictiveecology.r-universe.dev>

RemoteUrl <https://github.com/PredictiveEcology/LandR>

RemoteRef development

RemoteSha 42107da7b47972ea8589600534c337009c8bc929

Contents

LandR-package	5
.compareRas	6
.fitNLMwCovariates	7
.getMaxBCoefs	8
.interpolateLightProb	9
.MLLMaxBPartialPlotData	9
.MLLMaxBplotData	10
.NvalueFlag	12
addNoPixel2CohortData	13
addPixels2CohortData	13
aggregateRasByDT	14

asInteger	15
assert1	15
assertPostPartialDist	20
assignLightProb	21
biomodEnsembleFrcstWrapper	21
biomodEnsembleProjMaps	22
biomodEnsembleWrapper	23
biomodModelingWrapper	23
biomodProjWrapper	24
BioSIM_extractPoints	25
BioSIM_getMPBSLR	25
BioSIM_getWindAnnual	26
BioSIM_getWindMonthly	26
calcSeverityB	27
calcSiteShade	28
calculateClimateEffect	28
CASFRItoSpRasts	29
checkSpeciesTraits	30
cohortDefinitionCols	31
Colors	32
columnsForPixelGroups	33
convertUnwantedLCC	33
convertUnwantedLCC2	35
defineFlammable	36
describeCohortData	36
doResprouting	37
doSerotiny	39
dropTerm	41
ecoregionProducer	41
equivalentName	42
extractMaxB	42
fasterizeFromSp	43
FireDisturbance	43
fitNLMMModels	50
generatePixelGroups	52
genericExtract	53
genPGsPostDisturbance	53
getSpeciesTable	55
getWildfire_NFI	56
ggplotMLL_maxB	57
installBioSIM	58
LANDISDisp	59
lccMapGenerator	62
loadCASFRI	64
loadkNNSpeciesLayers	65
makeAndCleanInitialCohortData	66
makeBiomassMap	67
makeCohortDataFiles	68

makeDummyEcoregionMap	69
makeEcoregionDT	70
makeEcoregionMap	71
makeMinRelativeB	72
makePickellStack	72
makePixelGroupMap	73
makePixelTable	73
makeSpeciesEcoregion	74
makeVegTypeMap	76
minRelativeBDefaults	78
modifySpeciesAndSpeciesEcoregionTable	78
NAcover2zero	79
nonForestedPixels	80
overlayLCCs	80
overlayStacks	82
partialggplotMLL_maxB	82
partitionBiomass	84
pixelFate	85
plantNewCohorts	85
plotLeadingSpecies	86
plotSpatial	87
plotVTM	88
prepEcoregions	89
prepEcozonesRst	89
prepInputsCanDEM	90
prepInputsEcoregion	91
prepInputsFireYear	91
prepInputsLCC	92
prepInputsSpecies	93
prepInputsSpeciesEcoregion	94
prepInputsStandAgeMap	95
prepInputs_NTEMS_LCC_FAO	97
prepInputs_NTEMS_Nonforest	98
prepRasterToMatch	98
prepRawBiomassMap	99
prepSpeciesLayers_KNN	100
randomStudyArea	102
replaceAgeInFires	103
rescale	104
rmMissingCohorts	104
scheduleDisturbance	105
speciesEcoregionLatestYear	105
speciesEcoregionStack	106
speciesInStudyArea	107
speciesPresentFromKNN	108
speciesTableUpdate	109
sppColors	112
sppEquivalencies_CA	112

sppEquivCheck	114
sppHarmonize	114
standAgeMapGenerator	115
statsModel	116
subsetDT	117
sumRastersBySpecies	117
uniqueCohortDefinition	118
updateCohortData	118
updateCohortDataPostHarvest	120
updateSpeciesTable	122
vegTypeGenerator	123
vegTypeMapGenerator	124
vtm2conifdecid	126
WardKernel	127

Index	128
--------------	------------

LandR-package	LandR <i>package</i>
---------------	----------------------

Description

Utilities for 'LandR' suite of landscape simulation models. These models simulate forest vegetation dynamics based on LANDIS-II, and incorporate fire and insect disturbance, as well as other important ecological processes. Models are implemented as 'SpaDES' modules.

Package options

LandR packages use the following `options()` to configure behaviour:

- `LandR.assertions`: If TRUE, additional code checks are run during function calls. Default FALSE.

Author(s)

Maintainer: Eliot J B McIntire <eliot.mcintire@nrcan-rncan.gc.ca> ([ORCID](#))

Authors:

- Alex M Chubaty <achubaty@for-cast.ca> ([ORCID](#))
- Ceres Barros <ceres.barros@ubc.ca> ([ORCID](#))
- Ian M S Eddy <ian.eddy@nrcan-rncan.gc.ca> ([ORCID](#))

Other contributors:

- His Majesty the King in Right of Canada, as represented by the Minister of Natural Resources Canada [copyright holder]

See Also

Useful links:

- <https://landr.predictiveecology.org>
- <https://github.com/PredictiveEcology/LandR>
- Report bugs at <https://github.com/PredictiveEcology/LandR/issues>

.compareRas

Compare raster properties

Description

TODO: Move to reproducible

TODO: Move to reproducible TODO: expand to multiple objects

Usage

```
.compareRas(x, ...)
```

```
.compareCRS(x, y)
```

Arguments

x, y a Raster, SpatRaster, sf, SpatVector, or Spatial object

... additional Raster or SpatRaster objects, and arguments passed to `terra::compareGeom()`.

Value

the projected extent

Note

this function internally converts Raster to SpatRaster to allow using `compareGeom()`, and benefit from its complexity.

.fitNLMwCovariates *Fit non-linear growth model under various starting conditions*

Description

Uses likelihood parameter estimation to fit non linear models while attempting several starting values.

Usage

```
.fitNLMwCovariates(  
  data,  
  nonLinModelQuoted,  
  linModelQuoted,  
  mllsOuterPrev,  
  model = c("CR", "Logistic"),  
  maxCover = 1L,  
  starts = NULL,  
  lower = NULL,  
  upper = NULL,  
  nbWorkers = 1L  
)
```

Arguments

data	a data.table or data.frame with all covariates and the response variable. Note that incomplete lines are removed.
nonLinModelQuoted	The non-linear equation as a call (quoted expression) passed to <code>bbmle::mle2(minuslog1)</code> . See <code>?mle</code> . Accepts equations with three parameters 'A', 'p' and 'k'.
linModelQuoted	A list of linear equations/modes relating each parameter ('A', 'p' and 'k') with a set of covariates. A call (quoted expression) passed to <code>mle2(..., parameters)</code> . Note that for the purpose of tree growth, the linear equation determining 'A' should include a 'cover' predictor indicating the tree cover or dominance in the stand. Should be scaled between 0 and <code>maxCover</code> .
mllsOuterPrev	the output of a previous <code>fitNLMwCovariates</code> run which is used to extract last best AIC and maximum biomass estimate and judge if new iterations are better.
model	character. Non-linear model form used to estimate average maximum biomass. One of "CR" (Chapman-Richards) or "Logistic". In both cases, maximum biomass is equivalent to the 'A' asymptote parameter, which is estimated using observed mean values of predictors entering its linear equation and <code>cover == maxCover</code> , if this predictor is included (as it should). Passed to <code>extractMaxB</code>
maxCover	numeric. Value indicating maximum cover/dominance.
starts	data.table or data.frame of parameter starting values. Will be coerced to named list with names being parameter names.

lower	passed to bbmle::mle2
upper	passed to bbmle::mle2
nbWorkers	integer. If > 1, the number of workers to use in <code>parLapply::makeClusterPSOCK(nbWorkers = .)</code> , otherwise no parallelisation is done.

Value

a list with entries `m11` (the maximum likelihood-estimated coefficients) and `AICbest` (the AIC of the best models generating these coefficients)

See Also

[bbmle::mle2\(\)](#)

<code>.getMaxBCoefs</code>	<i>Get maximum biomass coefficient names</i>
----------------------------	--

Description

Extracts the names of linear coefficients for the maximum-biomass-equivalent parameter in the non-linear growth equations

Usage

```
.getMaxBCoefs(m11, model = c("CR", "Logistic"))
```

Arguments

<code>m11</code>	the output of an <code>bbmle::mle2</code> call (the fitted non-linear model), from which coefficient values will be extracted
<code>model</code>	character. Non-linear model form used to estimate average maximum biomass. One of "CR" (Chapman-Richards) or "Logistic".

Value

a list of two vectors of parameter names one following coefficient names in `m11` (`'m11CoefNames'`), the other using the original names as in the data used for model fitting (`'origCoefNames'`)

.interpolateLightProb *Find interpolated value of light probability*

Description

Find interpolated value of light probability

Usage

```
.interpolateLightProb(x, x0, x1, y0, y1)
```

Arguments

x	the species shade tolerance trait value for which we want to find the interpolated probability.
x0	the floor of x corresponding to a class of shade tolerance in the sufficientLight table.
x1	the ceiling of x corresponding to a class of shade tolerance in the sufficientLight table.
y0	the probability of germination in the sufficientLight table corresponding to x0.
y1	the probability of germination in the sufficientLight table corresponding to x1.

Value

vector of the interpolated value

.MLLMaxBPartialPlotData
Prepare data for model plotting

Description

Prepare data for model plotting

Usage

```
.MLLMaxBPartialPlotData(  
  mll,  
  nonLinModelQuoted,  
  linModelQuoted,  
  targetCovar = "cover",  
  fixMaxCover = TRUE,
```

```

maxCover = 1,
data,
fun = "mean",
plotCIs = TRUE
)

```

Arguments

<code>mll</code>	outputs of an <code>bbmle::mle2</code> call (the fitted non-linear model), from which coefficient values will be extracted.
<code>nonLinModelQuoted</code>	The non-linear equation as a call (quoted expression) passed to <code>mle2(minuslog1)</code> . See <code>?mle</code> . Accepts equations with three parameters 'A', 'p' and 'k'.
<code>linModelQuoted</code>	A list of linear equations/modes relating each parameter ('A', 'p' and 'k') with a set of covariates. A call (quoted expression) passed to <code>mle2(..., parameters)</code> . Note that for the purpose of tree growth, the linear equation determining 'A' should include a 'cover' predictor indicating the tree cover or dominance in the stand. Should be scaled between 0 and <code>maxCover</code> .
<code>targetCovar</code>	the covariate for which variation in <code>maxB</code> values will be shown. Defaults to showing how <code>maxB</code> values change with "cover". All other covariates except "age" are averaged. Age values are generated as <code>round(seq(min(age), max(age)*1.5, length.out = 100), 0)</code> . When <code>targetCovar != "cover"</code> , "cover" will be fixed at <code>maxCover</code> . See <code>fixMaxCover</code> .
<code>fixMaxCover</code>	logical. If TRUE and <code>targetCovar != "cover"</code> , cover is not averaged and is fixed to <code>maxCover</code> .
<code>maxCover</code>	numeric. Value indicating maximum cover/dominance.
<code>data</code>	data for estimation of maximum biomass. Should contain at least an 'age' column. Note that other covariates will be averaged and 'cover' values will be replaced with the maximum cover value (<code>maxCover</code>).
<code>fun</code>	The function to apply when summarizing other variables. By default, the all other variables except age are averaged ("mean"). Other options are: "median", "min", "max".
<code>plotCIs</code>	should confidence intervals be calculated and plotted?

See Also

[bbmle::mle2\(\)](#)

.MLLMaxBplotData *Prepare data for model plotting*

Description

Prepare data for model plotting

Usage

```
.MLLMaxBplotData(  
  mll,  
  nonLinModelQuoted,  
  linModelQuoted,  
  maxCover,  
  data,  
  averageCovariates = TRUE,  
  observedAge = FALSE,  
  plotCIs = TRUE  
)
```

Arguments

- mll** outputs of an `bbmle::mle2` call (the fitted non-linear model), from which coefficient values will be extracted.
- nonLinModelQuoted** The non-linear equation as a call (quoted expression) passed to `mle2(minuslog1)`. See `?mle`. Accepts equations with three parameters 'A', 'p' and 'k'.
- linModelQuoted** A list of linear equations/modes relating each parameter ('A', 'p' and 'k') with a set of covariates. A call (quoted expression) passed to `mle2(..., parameters)`. Note that for the purpose of tree growth, the linear equation determining 'A' should include a 'cover' predictor indicating the tree cover or dominance in the stand. Should be scaled between 0 and `maxCover`.
- maxCover** numeric. Value indicating maximum cover/dominance.
- data** data for estimation of maximum biomass. Should contain at least an 'age' column. Note that other covariates will be averaged and 'cover' values will be replaced with the maximum cover value (`maxCover`).
- averageCovariates** should covariates other than age/cover be averaged for biomass predictions? If not, for each age (at maximum cover) there will be as many predictions as other covariate values. If `observedAge == TRUE` and `averageCovariates == FALSE` then the original data is used, with cover changed to `maxCover`.
- observedAge** should observed age values be used, or should these be generated as `round(seq(min(age), max(age)*1.5, length.out = 100), 0)`? If `observedAge == TRUE` and `averageCovariates == FALSE` then the original data is used, with cover changed to `maxCover`.
- plotCIs** should confidence intervals be calculated and plotted?

See Also

[bbmle::mle2\(\)](#)

.NAvalueFlag *Set NA values in Raster or SpatRaster*

Description

These all create a single function that can be used for either Raster or SpatRaster objects.

Usage

```
.NAvalueFlag(ras, NAval)
.stack(rasList)
.projectExtent(ras, crs)
asInt(ras)
isInt(ras)
reclass(ras, tab)
```

Arguments

ras	a Raster, or SpatRaster object
NAval	the value to use as NA
rasList	a list of Raster or SpatRaster objects
crs	passed to raster::projectRaster(..., crs = crs) and terra::project(..., y = crs)
tab	matrix of values to reclassify. See terra::classify and raster::reclassify.

Value

a raster with attributed NA values
a stacked raster
the projected extent
asInt returns a *Raster with values converted to integer, if they weren't already.
isInt returns a logical as per is.integer.
reclass returns a *Raster with values reclassified as per terra::classify and raster::reclassify.

addNoPixel2CohortData *Add number of pixels per pixelGroup and add it has a new column to cohortData*

Description

Add number of pixels per pixelGroup and add it has a new column to cohortData

Usage

```
addNoPixel2CohortData(
  cohortData,
  pixelGroupMap,
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

cohortData A `data.table` with columns: `pixelGroup`, `ecoregionGroup`, `speciesCode`, and optionally `age`, `B`, `mortality`, `aNPPAct`, and `sumB`.

pixelGroupMap A `RasterLayer` with pixel values equal to a pixel group number that corresponds exactly to `pixelGroup` column in `cohortData`.

cohortDefinitionCols the columns in `cohortData` that define unique cohorts (default: `c("pixelGroup", "speciesCode", "age")`).

doAssertion A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: `getOption("LandR.assertions", TRUE)`.

Value

An `cohortData` `dat.table` with a new `noPixels` column

addPixels2CohortData *Generate cohortData table per pixel:*

Description

Generate `cohortData` table per pixel:

Usage

```
addPixels2CohortData(
  cohortData,
  pixelGroupMap,
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

cohortData A `data.table` with columns: `pixelGroup`, `ecoregionGroup`, `speciesCode`, and optionally `age`, `B`, `mortality`, `aNPPAct`, and `sumB`.

pixelGroupMap A `RasterLayer` with pixel values equal to a pixel group number that corresponds exactly to `pixelGroup` column in `cohortData`.

cohortDefinitionCols the columns in `cohortData` that define unique cohorts (default: `c("pixelGroup", "speciesCode", "age")`).

doAssertion A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: `getOption("LandR.assertions", TRUE)`.

Value

An expanded `cohortData` `data.table` with a new `pixelIndex` column.

aggregateRasByDT	<i>Aggregate a raster</i>
------------------	---------------------------

Description

Uses **data.table** to perform aggregation calculations, which is faster than `raster::aggregate`.

Usage

```
aggregateRasByDT(ras, newRas, fn = sum)
```

Arguments

ras `RasterLayer` to aggregate

newRas `RasterLayer` to match

fn function to use to aggregate pixel values

Value

`RasterLayer`

asInteger	<i>Convert numeric values to rounded integers</i>
-----------	---

Description

Essentially a wrapper around round, rather than truncate, which is what `as.integer` does. Internally, this is simply `as.integer(floor(x + 0.5))`.

Usage

```
asInteger(x)
```

Arguments

`x` A numeric vector

Value

An integer vector of length `x`, rounded to zero decimal places prior to `as.integer`

Note

Values ending in `.5` will be rounded up, whether positive or negative. This is different than `round`.

Examples

```
x <- seq(-2, 2, 0.25)
data.frame(dbl = x, int = asInteger(x))
```

assert1	<i>Assertions</i>
---------	-------------------

Description

- `assert1`: Assert that `ecoregionCodes` that were replaced, were correctly identified; This is the full `pixelCohortData`, not the collapsed one.

Usage

```
assert1(
  cohortData34to36,
  cohortData,
  rmZeroBiomassQuote,
  classesToReplace = 34:36,
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

```
assert2(  
  cohortDataNo34to36,  
  classesToReplace = 34:36,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertSppMaxBMaxANPP(  
  speciesEcoregion,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertUniqueCohortData(  
  cohortData,  
  columns,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertERGs(  
  ecoregionMap,  
  cohortData,  
  speciesEcoregion,  
  minRelativeB,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertColumns(  
  obj,  
  colClasses,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertCohortData(  
  cohortData,  
  pixelGroupMap,  
  maxExpectedNumDiverge = 1,  
  message = "",  
  doAssertion = getOption("LandR.assertions", TRUE),  
  verbose = getOption("LandR.verbose", TRUE),  
  cohortDefinitionCols = LandR::cohortDefinitionCols()  
)  
  
assertCohortDataERG(  
  cohortData,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)  
  
assertPixelCohortData(  
  cohortData,  
  pixelGroupMap,  
  doAssertion = getOption("LandR.assertions", TRUE)  
)
```



```
    pixelCohortData,  
    pixelGroupMap,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertSpeciesPlotLabels(  
    speciesNames,  
    sppEquiv,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertFireToleranceDif(  
    burnedPixelCohortData,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertSpeciesLayers(  
    speciesLayers,  
    thresh,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertRstLCChange(  
    rstLCChange,  
    rasterToMatch,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertSpeciesEcoregionCohortDataMatch(  
    cohortData,  
    speciesEcoregion,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertPixelCohortDataValid(  
    standCohortData,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertRepsAllCohortData(  
    allCohortData,  
    reps,  
    years,  
    doAssertion = getOption("LandR.assertions", TRUE)  
  )  
  
  assertStandAgeMapAttr(  
    standAgeMap,
```

```

doAssertion = getOption("LandR.assertions", TRUE)
)

assertCohortDataAttr(
  cohortData,
  doAssertion = getOption("LandR.assertions", TRUE)
)

assertSppVectors(
  sppEquiv = NULL,
  sppNameVector = NULL,
  sppColorVect = NULL,
  sppEquivCol = NULL,
  doAssertion = getOption("LandR.assertions", TRUE)
)

assertSpeciesTableRaw(
  speciesTableRaw,
  doAssertion = getOption("LandR.assertions", TRUE)
)

assertSpeciesTable(
  speciesTable,
  doAssertion = getOption("LandR.assertions", TRUE)
)

```

Arguments

cohortData34to36
A cohortData data.table with only the pixels what were LCC 34:36

cohortData
A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.

rmZeroBiomassQuote
An expression to evaluate, in the form of quote(B>0), used to select cohorts with biomass.

classesToReplace
Integer vector of classes that are to be replaced, e.g., 34, 35, 36 on LCC2005, which are burned young, burned 10 year, and cities.

doAssertion
A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

cohortDataNo34to36
A cohortData data.table with only the pixels what were LCC 34:36

speciesEcoregion
A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species

	in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).
columns	Vector of column names on which to test for unique cohortData
ecoregionMap	The ecoregionMap, a raster of all the unique ecoregion groupings, which group together pixels of similar biophysical characteristics.
minRelativeB	a data.frame with the cut points to classify stand shadiness.
obj	A data.frame- or data.table-like object
colClasses	A named vector of column classes, where the names are the column names
pixelGroupMap	A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to pixelGroup column in cohortData.
maxExpectedNumDiverge	A numeric, length 1, indicating by how many they can diverge. Default 1.
message	An optional message to print. This may help identify where this function was called.
verbose	Controls message output. Defaults to getOption("LandR.verbose")
cohortDefinitionCols	the columns in cohortData that define unique cohorts (default: c("pixelGroup", "speciesCode", "age")).
pixelCohortData	The full cohortData data.table
speciesNames	A vector of species names for which the labels will be checked
sppEquiv	an table with a column containing species names
burnedPixelCohortData	An expanded cohortData data.table with pixel-level cohort information on burnt pixels and the following (optional) columns: severity - fire severity in that pixel calculated based on fire behaviour properties; firetolerance - species-level fire tolerance; severityToleranceDif - the difference between severity and firetolerance.
speciesLayers	A RasterStack or RasterLayer that should contain species cover data in the study area
thresh	the minimum number of pixels where the species must have biomass > 0 to be considered present in the study area. Defaults to 1.
rstLCChange	a raster layer indicating pixels were land-use change occurred as 1s
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
standCohortData	A data.table with simulated and observed stand data for validation
allCohortData	A data.table with all simulated cohortData to use for validation
reps	repetition ids
years	years

standAgeMap	A rasterLayer of stand ages with attribute "imputedPixID"
sppNameVector	A character vector of species to use. These species must all be from one naming convention, i.e., from one column in the sppEquiv.
sppColorVect	A named vector of colours to use for plotting. The names must conform with species name convention used (see sppEquivalencies_CA for possible naming conventions) and should also contain a colour for 'Mixed', when a mixed forest type is supported (see vegLeadingProportion parameter in vegTypeGenerator() for details on mixed forest types).
sppEquivCol	the column name to use from sppEquiv.
speciesTableRaw	raw species traits data. table (see getSpeciesTable())
speciesTable	species traits data. table (see prepSpeciesTable())

assertPostPartialDist *Assert post-fire disturbance mortality and regeneration*

Description

Assert post-fire disturbance mortality and regeneration

Usage

```
assertPostPartialDist(
  cohortDataOrig,
  pixelGroupMapOrig,
  cohortDataNew,
  pixelGroupMapNew,
  postDistPixelCohortData,
  distrbdPixelCohortData,
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

cohortDataOrig original cohortData (prior to any modification)
 pixelGroupMapOrig original pixelGroupMap (prior to any modification)
 cohortDataNew modified cohortData output from updateCohortData()
 pixelGroupMapNew modified pixelGroupMap output from updateCohortData()
 postDistPixelCohortData modified cohortData output from updateCohortData()
 distrbdPixelCohortData cohortData-like object containing all dead, surviving and new cohorts (i.e. activated by serotiny/resprouting)
 doAssertion A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

assignLightProb	<i>Assign light probability</i>
-----------------	---------------------------------

Description

Assign light probability

Usage

```
assignLightProb(
  sufficientLight,
  newCohortData,
  interpolate = TRUE,
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

sufficientLight	a data.frame (not data.table!) containing probability of establishment, given a site's light conditions (X0-X5) for each level of a species shade tolerance (1-5).
newCohortData	a modified version of cohortData that contains new cohorts.
interpolate	Logical. Activates interpolation of probabilities of establishment between any two values of shade tolerance in the sufficient light table, allowing species shade tolerance trait values to take any decimal value between 1 and 5 (inclusively). If FALSE, species shade tolerances can only take integer values between 1 and 5 (inclusively).
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Value

newCohortData with a lightProb column

biomodEnsembleFrcstWrapper	<i>Simple wrapper around</i> <code>biomod2::BIOMOD_EnsembleForecasting()</code>
----------------------------	---

Description

Simple wrapper around `biomod2::BIOMOD_EnsembleForecasting()`

Usage

```

biomodEnsembleFrcstWrapper(
  bm.em,
  bm.proj = NULL,
  proj.name = NULL,
  new.env = NULL,
  new.env.xy = NULL,
  keep.in.memory = TRUE,
  ...
)

```

Arguments

bm.em	output of biomod2::BIOMOD_EnsembleModeling
bm.proj	output of biomod2::BIOMOD_Projection
proj.name	passed to biomod2::BIOMOD_Projection
new.env	passed to biomod2::BIOMOD_EnsembleForecasting
new.env.xy	passed to biomod2::BIOMOD_EnsembleForecasting
keep.in.memory	passed to biomod2::BIOMOD_EnsembleForecasting
...	passed to reproducible::Cache

biomodEnsembleProjMaps

Make maps from BIOMOD_EnsembleForecasting() results

Description

Make maps from BIOMOD_EnsembleForecasting() results

Usage

```

biomodEnsembleProjMaps(bm.em.proj, predModel, rasTemplate, origCRS)

```

Arguments

bm.em.proj	output of biomod2::BIOMOD_EnsembleForecasting() . Note that x and y coordinates extracted from <code>bm.em.proj@coord</code> must be in the same projection as <code>rasTemplate</code> .
predModel	character. Which model predictions should be used. Choose one of <code>bm.em.proj@models.projected</code> .
rasTemplate	a template <code>RasterLayer</code> or <code>SpatRaster</code> that can be used to map the projections
origCRS	character. The original CRS projection of <code>bm.em.proj@coord</code> .

Value

a `SpatRaster` object

biomodEnsembleWrapper *BIOMOD_EnsembleModeling wrapper*

Description

BIOMOD_EnsembleModeling wrapper

Usage

```
biomodEnsembleWrapper(bm.mod, metric.select.thresh = NULL, ...)
```

Arguments

bm.mod	output of biomod2::BIOMOD_Modeling
metric.select.thresh	passed biomod2::BIOMOD_EnsembleModeling . By default no thresholding is applied.
...	passed to reproducible::Cache

biomodModelingWrapper *BIOMOD_Modeling wrapper*

Description

BIOMOD_Modeling wrapper

Usage

```
biomodModelingWrapper(
  sp,
  responseVar,
  responseVarData,
  predictorVars,
  predictorVarsData,
  dir.name,
  BIOMOD_ModelingArgs = list(models = c("GLM", "MARS"), bm.options = NULL, CV.k = 5,
    CV.perc = 100, metric.eval = c("TSS", "ROC"), modeling.id = "test"),
  ...
)
```

Arguments

sp	character of species name to subset responseVarData table
responseVar	character vector, or list of character vectors, of response variable (column) to subset responseVarData table. If a list, it should be named according to sp, for subsetting.
responseVarData	a data.table or list of data.tables with environmental data. If a list, it should be named according to sp, for subsetting.#'
predictorVars	character vector, or list of character vectors, of environmental variables (columns) to subset predictorVarsData table. If a list, it should be named according to sp, for subsetting.
predictorVarsData	environmental data.
dir.name	passed to biomod2::BIOMOD_FormatingData
BIOMOD_ModelingArgs	a named list of arguments passed to biomod2::BIOMOD_Modeling
...	further arguments passed to reproducible::Cache

biomodProjWrapper *BIOMOD_Projection wrapper*

Description

BIOMOD_Projection wrapper

Usage

```
biomodProjWrapper(
  bm.mod,
  proj.name = "testProj",
  new.env = NULL,
  new.env.xy = NULL,
  ...
)
```

Arguments

bm.mod	passed to biomod2::BIOMOD_Projection . If not supplied the data used to fit the model will be used.
proj.name	passed to biomod2::BIOMOD_Projection
new.env	passed to biomod2::BIOMOD_Projection
new.env.xy	passed to biomod2::BIOMOD_Projection
...	passed to reproducible::Cache

BioSIM_extractPoints *Extract point locations from DEM raster to pass to BioSIM functions*

Description

Extract point locations from DEM raster to pass to BioSIM functions

Usage

```
BioSIM_extractPoints(x)
```

Arguments

x A digital elevation model (DEM) RasterLayer.

Value

data.table with columns Name, Long, Lat, Elev.

See Also

[prepInputsCanDEM\(\)](#)

BioSIM_getMPBSLR *Get annual historic and projected MPB climate suitability maps from BioSIM*

Description

Raster stacks for all 9 MPB climate indices. See `BioSIM::getModelHelp("MPB_SLR")`.

Usage

```
BioSIM_getMPBSLR(dem, years, SLR = "R", climModel = "GCM4", rcp = "RCP45")
```

Arguments

dem RasterLayer of elevation data (m).
years numeric vector corresponding to the years to retrieve.
SLR character. Specifies which climate suitability index to extract. Currently, one of "S", "L", "R", or "G", corresponding to Safranyik-P3P4, Logan-2b, Régnière Cold Tolerance Survival, or their Geometric product (S\L\R), respectively.
climModel climate model to use. one of "GCM4" or "RCM4".
rcp RCP scenario to use. one of "RCP45" or "RCP85".

Value

RasterStack

Note

Although the BioSIM MPB_SLR model provides several other indices (see `BioSIM::getModelHelp("MPB_SLR")`), only 4 are currently used here.

`BioSIM_getWindAnnual` *Get annual historic and projected wind maps from BioSIM*

Description

Get annual historic and projected wind maps from BioSIM

Usage

```
BioSIM_getWindAnnual(dem, years, climModel = "GCM4", rcp = "RCP45")
```

Arguments

<code>dem</code>	RasterLayer of elevation data (m).
<code>years</code>	numeric vector corresponding to the years to retrieve.
<code>climModel</code>	climate model to use. one of "GCM4" or "RCM4".
<code>rcp</code>	RCP scenario to use. one of "RCP45" or "RCP85".

Value

RasterStack

`BioSIM_getWindMonthly` *Get monthly historic and projected wind maps from BioSIM*

Description

Get monthly historic and projected wind maps from BioSIM

Usage

```
BioSIM_getWindMonthly(dem, years, months, climModel = "GCM4", rcp = "RCP45")
```

Arguments

dem	RasterLayer of elevation data (m).
years	numeric vector corresponding to the years to retrieve.
months	numeric vector corresponding to the months to retrieve (e.g., 6 : 8 for June through August).
climModel	climate model to use. one of "GCM4" or "RCM4".
rcp	RCP scenario to use. one of "RCP45" or "RCP85".

Value

RasterStack

calcSeverityB	<i>Calculate fire severity</i>
---------------	--------------------------------

Description

Calculates fire severity as the loss of pre-fire to post-fire biomass (in absolute and percentual terms).

Usage

```
calcSeverityB(cohortData, burnedPixelCohortData)
```

Arguments

cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
burnedPixelCohortData	An expanded cohortData data.table with pixel-level cohort information on burnt pixels and the following (optional) columns: severity - fire severity in that pixel calculated based on fire behaviour properties; firetolerance - species-level fire tolerance; severityToleranceDif - the difference between severity and firetolerance.

Value

data.table with columns pixelIndex, pixelGroup and severityB (absolute biomass lost) and severityPropB (proportion of biomass lost)

Note

if burnedPixelCohortData does not have a B column, the fire is assumed to be stand replacing (i.e. we assume B to be 0 across all pixels/cohorts in burnedPixelCohortData)

 calcSiteShade

Calculate site shade

Description

Calculate site shade

Usage

```
calcSiteShade(currentTime, cohortData, speciesEcoregion, minRelativeB)
```

Arguments

`currentTime` The current simulation time e.g., `time(sim)`.

`cohortData` A `data.table` with columns: `pixelGroup`, `ecoregionGroup`, `speciesCode`, and optionally `age`, `B`, `mortality`, `aNPPAct`, and `sumB`.

`speciesEcoregion` A `data.table` with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: `speciesCode` and `ecoregionGroup`, character representation of species and ecoregion groups respectively, `maxB` the maximum biomass for the species in a given 'ecoregion', `maxANPP` the maximum aboveground net primary productivity and `SEP` the species establishment probability. May contain columns `inflationFactor` (used to adjust `maxB`) and `maxANPPproportion` (used to calculate `maxANPP`).

`minRelativeB` a `data.frame` with the cut points to classify stand shadiness.

Value

`cohortData` table with a `siteShade` column

See Also

[makeMinRelativeB\(\)](#)

 calculateClimateEffect

Null climate effect

Description

Default climate effects function in the case where no climate effect is simulated

Usage

```
calculateClimateEffect(cohortData, ...)
```

Arguments

cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
...	additional arguments that are passed to LandR.CS

Details

the cohortData object is used to calculate the \ reduction/increase of mortality and growth biomasses per cohort.

Value

data.table with pixelGroup, age and speciesCode, as well as mortPred and growthPred columns with \ growth biomasses resulting from a climate effect. These percentages are later multiplied by the by baseline biomasses of mortality and growth (e.g. 0\ This default, no climate effect, function outputs 100\ mortPred and growthPred.

CASFRIToSpRasts	CASFRIToSpRasts
-----------------	-----------------

Description

TODO: description and title needed

Usage

```
CASFRIToSpRasts(  
  CASFRIRas,  
  CASFRIattrLong,  
  CASFRIdt,  
  sppEquiv,  
  sppEquivCol,  
  destinationPath  
)
```

Arguments

CASFRIRas	TODO: description needed
CASFRIattrLong	TODO: description needed
CASFRIdt	TODO: description needed

sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
destinationPath	path to data directory where objects will be downloaded or saved to

Value

TODO: description needed

checkSpeciesTraits	<i>Check if all species in have trait values</i>
--------------------	--

Description

Check if all species in have trait values

Usage

```
checkSpeciesTraits(speciesLayers, species, sppColorVect)
```

Arguments

speciesLayers	A RasterStack or RasterLayer that should contain species cover data in the study area
species	a data.table with species traits such as longevity, shade tolerance, etc. Must have column speciesCode, with species names/IDs. The following is a list of default trait columns: <ul style="list-style-type: none"> • "species" same as "speciesCode" – species ID name • "speciesCode" • "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits where derived. Not used during the simulation • "firetolerance" – modulerelative (to other species) fire tolerance • "growthcurve" and "mortalityshape" – growth curve shape parameters. • "longevity" – maximum species age • "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none") • "resproutprob" – probability of resprouting • "resproutage_min" – minimum age at which species is capable of resprouting

- "resproutage_max" – maximum age at which species is capable of resprouting
 - "seeddistance_eff" – effective dispersal distance
 - "seeddistance_max" – maximum dispersal distance
 - "shadetolerance" – *relative* (to other species) shade tolerance
 - "sexualmature" – age at sexual maturity Known optional parameters added/needed by some modules (the user may add others for their own modules):
 - "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB
 - "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated
 - "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB
 - "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermocarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment:
 - "leaflongevity"
 - "wooddecayrate"
 - "leafLignin"
 - "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.
- sppColorVect A named vector of colours to use for plotting. The names must conform with species name convention used (see [sppEquivalencies_CA](#) for possible naming conventions) and should also contain a colour for 'Mixed', when a mixed forest type is supported (see vegLeadingProportion parameter in [vegTypeGenerator\(\)](#) for details on mixed forest types).

Value

A list with the speciesLayers and sppColorVect containing only the species that have trait values in species

cohortDefinitionCols *Default columns that define cohorts*

Description

Default columns that define cohorts

Usage

```
cohortDefinitionCols()
```

Note

because the name `cohortDefinitionCols` is also used as a function argument, be sure to use `LandR::cohortDefinitionCols()` in those functions or you'll get a "promise already under evaluation" error.

Colors

Helper for setting Raster or SpatRaster colors

Description

This is a wrapper to help with migration to **terra**. Currently can only be used for a single layer `SpatRaster` or a `RasterLayer`.

Usage

```
Colors(ras, cols, n = NULL)
```

Arguments

<code>ras</code>	A <code>Raster*</code> or <code>SpatRaster</code> class object.
<code>cols</code>	a character vector of colours. See examples. Can also be a <code>data.frame</code> , see <code>terra::coltab</code>
<code>n</code>	A numeric scalar giving the number of colours to create. Passed to <code>quickPlot::setColors(ras, n = n)</code> If missing, then <code>n</code> will be <code>length(cols)</code>

Examples

```
cols <- colorRampPalette(c("blue", "red"))(12)
ras <- terra::rast(matrix(1:100, 10, 10))
ras <- Colors(ras, cols)
terra::plot(ras)
```

```
ras <- raster::raster(matrix(1:100, 10, 10))
ras <- Colors(ras, cols)
raster::plot(ras)
```

columnsForPixelGroups *Default columns that define pixel groups*

Description

Default columns that define pixel groups

Usage

```
columnsForPixelGroups()
```

Note

because the name columnsForPixelGroups is also used as a function argument, be sure to use LandR::columnsForPixelGroups() in those functions or you'll get a "promise already under evaluation" error.

convertUnwantedLCC *Convert Land Cover Classes (LCC) to another value in its neighbourhood*

Description

This will search around the pixels on rstLCC that have classesToReplace, and search in iteratively increasing radii outwards for other Land Cover Classes than the those indicated in classesToReplace. This will constrain It will then take the cohorts that were in pixels with classesToReplace and assign them new values in the output object. This function will also check that it must be an ecoregionCode that already exists in cohortData, i.e., not create new ecoregionCode values. See Details.

Usage

```
convertUnwantedLCC(
  classesToReplace = 34:36,
  rstLCC,
  availableERC_by_Sp,
  theUnwantedPixels,
  ecoregionGroupVec,
  speciesEcoregion,
  pixelClassesToReplace,
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

classesToReplace	Integer vector of classes that are to be replaced, e.g., 34, 35, 36 on LCC2005, which are burned young, burned 10 year, and cities.
rstLCC	LCC raster, e.g., LCC2005
availableERC_by_Sp	A <code>data.table</code> or <code>data.frame</code> with 3 columns: <code>speciesCode</code> , <code>initialEcoregionCode</code> and <code>pixelIndex</code> . <code>pixelIndex</code> is the pixel id for each line in the <code>data.table</code> ; <code>speciesCode</code> is the species name in the pixel (can have more than one species per pixel, so multiple rows per pixel); and, <code>initialEcoregionCode</code> is the unique codes that are "available" to be used as a replacement for <code>classesToReplace</code> . <code>initialEcoregionCode</code> must be a character vector, with one or no "" used as a separator, with the last component being the Land Cover Class that matches <code>classesToReplace</code> , e.g., "242_18". If there is no "" in this code, then the codes must match the <code>classesToReplace</code> exactly, e.g., "11". If <code>pixelIndex</code> is missing, the function will fill it with <code>seq(ncell(rstLCC))</code> . If <code>speciesCode</code> is missing, the function will replace it with a dummy value ("allSpecies").
theUnwantedPixels	An optional vector of pixel IDs that need to be changed. If not provided, then pixels to change will be taken from the match between <code>availableERC_by_Sp</code> and <code>classesToReplace</code> . Supplying this allows the user to only replace some of the pixels with a given class.
ecoregionGroupVec	Deprecated. Use <code>availableERC_by_Sp</code>
speciesEcoregion	Deprecated. Use <code>availableERC_by_Sp</code>
pixelClassesToReplace	Deprecated. Use <code>classesToReplace</code>
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: <code>getOption("LandR.assertions", TRUE)</code> .

Details

This function is designed to be used in highly constrained situations, where it is not just replacing a Land Cover Class by a neighbouring Land Cover Class. But it can be used for the simpler cases of simply replacing a Land Cover Class.

Value

A `data.table` with two columns, `pixelIndex` and `ecoregionGroup`. This represents the new codes to used in the `pixelIndex` locations. These should have no values overlapping with `classesToReplace`.

Author(s)

Eliot McIntire

convertUnwantedLCC2 *Convert Land Cover Classes (LCC) to another value in its neighbourhood*

Description

This will search around the pixels on rstLCC that have classesToReplace, and search in iteratively increasing radii outwards for other Land Cover Classes than the those indicated in classesToReplace.

Usage

```
convertUnwantedLCC2(
  classesToReplace = 34:36,
  rstLCC,
  nIterations = 6,
  defaultNewValue = NA,
  invalidClasses = NA
)
```

Arguments

classesToReplace Integer vector of classes that are to be replaced, e.g., 34, 35, 36 on LCC2005, which are burned young, burned 10 year, and cities.

rstLCC raster of land-cover class, e.g., LCC2005

nIterations the number of iterations to perform

defaultNewValue the value to assign a pixel in classesToReplace if no valid pixel is closer after nIterations

invalidClasses classes that are not valid options

Value

A rasterLayer with values in classesToReplace converted to adjacent values or NA.

Author(s)

Eliot McIntire Ian Eddy

defineFlammable *Define flammability map*

Description

Define flammability map

Usage

```
defineFlammable(
  LandCoverClassifiedMap = NULL,
  nonFlammClasses = c(0L, 25L, 30L, 33L, 36L, 37L, 38L, 39L),
  mask = NULL,
  to = NULL,
  writeTo = NULL,
  ...
)
```

Arguments

LandCoverClassifiedMap	A Raster that represents land cover (e.g., Land Cover Classified map from 2005 or 2010 from the Canadian Forest Service).
nonFlammClasses	numeric vector defining which classes in LandCoverClassifiedMap.
mask	A raster to use as a mask (see raster::mask()).
to	Passed to <code>postProcessTo(..., to = to)</code> and to the mask arg here, if mask is not supplied.
writeTo	See reproducible::postProcess() . Default NULL.
...	additional args (not used)

describeCohortData *Summary for cohortData*

Description

Summary for cohortData

Usage

```
describeCohortData(cohortData)
```

Arguments

cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
------------	--

doResprouting	<i>Activate resprouting after a (fire) disturbance</i>
---------------	--

Description

Activate resprouting after a (fire) disturbance

Usage

```
doResprouting(
  burnedPixelCohortData,
  postFirePixelCohortData,
  postFireRegenSummary = NULL,
  serotinyPixel,
  treedFirePixelTableSinceLastDisp,
  currentTime,
  species,
  sufficientLight,
  calibrate = FALSE
)
```

Arguments

burnedPixelCohortData
An expanded cohortData data.table with pixel-level cohort information on burnt pixels and the following (optional) columns: severity - fire severity in that pixel calculated based on fire behaviour properties; firetolerance - species-level fire tolerance; severityToleranceDif - the difference between severity and firetolerance.

postFirePixelCohortData
an empty cohortData-like table with columns: age, B, mortality, aNPPAct, and sumB removed; and pixelIndex added.

postFireRegenSummary
a data.table summarizing for which species serotiny/resprouting were activated and in how many pixels, for each year. Only necessary if calibrate = TRUE.

serotinyPixel a vector of pixels where serotiny was activated;

treedFirePixelTableSinceLastDisp
a vector of pixels that burnt and were forested in the previous time step.

currentTime integer. The current simulation time obtained with time(sim)

species a data.table with species traits such as longevity, shade tolerance, etc. Must have column speciesCode, with species names/IDs. The following is a list of default trait columns:

- "species" same as "speciesCode" – species ID name
- "speciesCode"

- "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits were derived. Not used during the simulation
- "firetolerance" – *modulere* (to other species) fire tolerance
- "growthcurve" and "mortalityshape" – growth curve shape parameters.
- "longevity" – maximum species age
- "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- "resproutprob" – probability of resprouting
- "resproutage_min" – minimum age at which species is capable of resprouting
- "resproutage_max" – maximum age at which species is capable of resprouting
- "seeddistance_eff" – effective dispersal distance
- "seeddistance_max" – maximum dispersal distance
- "shadetolerance" – *relative* (to other species) shade tolerance
- "sexualmature" – age at sexual maturity Known optional parameters added/needed by some modules (the user may add others for their own modules):
- "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB
- "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated
- "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB
- "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermocarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment:
- "leaflongevity"
- "wooddecayrate"
- "leafLignin"
- "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

sufficientLight

a data.frame (not data.table!) containing probability of establishment, given a site's light conditions (X0-X5) for each level of a species shade tolerance (1-5).

calibrate

logical. Determines whether to output postFirePixelCohortData. Defaults to FALSE

Value

A list of objects: postFirePixelCohortData, a data.table with the cohorts that undergo serotiny; serotinyPixel, a vector of pixels where serotiny was activated; postFireRegenSummary, the updated postFireRegenSummary, if calibrate = TRUE.

doSerotiny	<i>Activate serotiny after a (fire) disturbance</i>
------------	---

Description

Activate serotiny after a (fire) disturbance

Usage

```
doSerotiny(
  burnedPixelCohortData,
  postFirePixelCohortData,
  postFireRegenSummary = NULL,
  species,
  sufficientLight,
  speciesEcoregion,
  currentTime,
  treedFirePixelTableSinceLastDisp,
  calibrate = FALSE
)
```

Arguments

- | | |
|-------------------------|---|
| burnedPixelCohortData | An expanded cohortData data.table with pixel-level cohort information on burnt pixels and the following (optional) columns: severity - fire severity in that pixel calculated based on fire behaviour properties; firetolerance - species-level fire tolerance; severityToleranceDif - the difference between severity and firetolerance. |
| postFirePixelCohortData | an empty cohortData-like table with columns: age, B, mortality, aNPPAct, and sumB removed; and pixelIndex added. |
| postFireRegenSummary | a data.table summarizing for which species serotiny/resprouting were activated and in how many pixels, for each year. Only necessary if calibrate = TRUE. |
| species | a data.table with species traits such as longevity, shade tolerance, etc. Must have column speciesCode, with species names/IDs. The following is a list of default trait columns: <ul style="list-style-type: none"> • "species" same as "speciesCode" – species ID name • "speciesCode" • "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits were derived. Not used during the simulation • "firetolerance" – modular <i>relative</i> (to other species) fire tolerance • "growthcurve" and "mortalityshape" – growth curve shape parameters. • "longevity" – maximum species age |

- "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- "resproutprob" – probability of resprouting
- "resproutage_min" – minimum age at which species is capable of resprouting
- "resproutage_max" – maximum age at which species is capable of resprouting
- "seeddistance_eff" – effective dispersal distance
- "seeddistance_max" – maximum dispersal distance
- "shadetolerance" – *relative* (to other species) shade tolerance
- "sexualmature" – age at sexual maturity Known optional parameters added/needed by some modules (the user may add others for their own modules):
- "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB
- "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated
- "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB
- "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermokarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment:
- "leaflongevity"
- "wooddecayrate"
- "leafLignin"
- "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

sufficientLight

a data.frame (not data.table!) containing probability of establishment, given a site's light conditions (X0-X5) for each level of a species shade tolerance (1-5).

speciesEcoregion

A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).

currentTime integer. The current simulation time obtained with time(sim)

treedFirePixelTableSinceLastDisp

a vector of pixels that burnt and were forested in the previous time step.

calibrate logical. Determines whether to output postFirePixelCohortData. Defaults to FALSE

Value

A list of objects: postFirePixelCohortData, a data.table with the cohorts that undergo serotiny; serotinyPixel, a vector of pixels where serotiny was activated; postFireRegenSummary, the updated postFireRegenSummary, if calibrate = TRUE.

dropTerm	<i>Drop factor term including interactions from a model formula</i>
----------	---

Description

Based on <https://stackoverflow.com/a/23382097/1380598>.

Usage

```
dropTerm(form, term)
```

Arguments

form	A model formula.
term	Character vector giving the name of the term to drop.

Value

An updated model formula.

ecoregionProducer	<i>Make ecoregionMap and ecoregion table</i>
-------------------	--

Description

This function combines an ecoregion map and a land cover map (e.g. ecodistricts and LCC) and creates a map and table of containing their combined values and pixel IDs. Used internally in LandR modules to prepare maps for to make cohortData.

Usage

```
ecoregionProducer(ecoregionMaps, ecoregionName = NULL, rasterToMatch)
```

Arguments

ecoregionMaps	a list with two rasters, one with ecoregions (e.g. ecodistricts) and another with land cover (e.g. LCC).
ecoregionName	the name describing the type of ecoregions in first map (e.g. "ecoDistrict") if passing a polygon file.
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

A list with two objects: the ecoregionMap and a table summarizing its information per pixelID

equivalentName	<i>Return equivalent name from a data.frame of equivalencies</i>
----------------	--

Description

This is simply a wrapper around match or \%in\% for a specific data.frame of values.

Usage

```
equivalentName(value, df, column, multi = FALSE, searchColumn = NULL)
```

```
equivalentNameColumn(value, df, column, multi = FALSE, searchColumn = NULL)
```

Arguments

value	Vector of values to match in df.
df	A data.frame where every row is a set of equivalent names.
column	A character string or numeric of length 1, indicating the column in df to return names from.
multi	Logical. If TRUE, then all matches will be returned. Default FALSE for backwards compatibility. This may result in more elements than were in value, as each value may be matched by more than one entry, returning more than one result each from column. If a species is found in the df, but there is no corresponding value in column, it will return NA.
searchColumn	Optionally, provide the name of a column in df that results must be found in. The return value will still be from column

extractMaxB	<i>Maximum biomass estimator</i>
-------------	----------------------------------

Description

Estimation of maximum biomass as the A parameter in the Chapman-Richards and Logistic growth equations. Since A is modelled as linear term, it is a matrix product of its linear coefficients. It is assumed that all coefficients are related additively.

Usage

```
extractMaxB(mll, newdata, average = FALSE, model = c("CR", "Logistic"))
```

Arguments

m11	the output of an <code>bbmle::mle2</code> call (the fitted non-linear model), from which coefficient values will be extracted
newdata	data for estimation of 'A'
average	should 'A' be estimated for average values of its predictors.
model	character. Non-linear model form used to estimate average maximum biomass. One of "CR" (Chapman-Richards) or "Logistic".

fasterizeFromSp	<i>Rasterize polygons using fasterize</i>
-----------------	---

Description

Rasterize polygons using fasterize

Usage

```
fasterizeFromSp(sp, raster, fieldName)
```

Arguments

sp	a shapefile to rasterize
raster	the template raster to use
fieldName	the field to use (will be ignored if the shapefile has no fields)

Value

RasterLayer

FireDisturbance	<i>Disturbance functions</i>
-----------------	------------------------------

Description

Disturbance functions

Usage

```

FireDisturbance(
  cohortData = copy(sim$cohortData),
  cohortDefinitionCols = c("pixelGroup", "age", "speciesCode"),
  calibrate = FALSE,
  postFireRegenSummary = copy(sim$postFireRegenSummary),
  treedFirePixelTableSinceLastDisp = sim$treedFirePixelTableSinceLastDisp,
  rstCurrentBurn = sim$rstCurrentBurn,
  inactivePixelIndex = sim$inactivePixelIndex,
  pixelGroupMap = sim$pixelGroupMap,
  currentTime = NULL,
  rasterToMatch = sim$rasterToMatch,
  species = copy(sim$species),
  sufficientLight = copy(sim$sufficientLight),
  speciesEcoregion = copy(sim$speciesEcoregion),
  initialB = 10,
  successionTimestep = 10L,
  doAssertion = getOption("LandR.assertions", TRUE),
  verbose = getOption("LandR.verbose", TRUE)
)

FireDisturbancePM(
  cohortData = copy(sim$cohortData),
  cohortDefinitionCols = c("pixelGroup", "age", "speciesCode"),
  colsForPixelGroups = LandR::columnsForPixelGroups(),
  calibrate = FALSE,
  LANDISPM = TRUE,
  postFireRegenSummary = copy(sim$postFireRegenSummary),
  treedFirePixelTableSinceLastDisp = copy(sim$treedFirePixelTableSinceLastDisp),
  rstCurrentBurn = sim$rstCurrentBurn,
  inactivePixelIndex = sim$inactivePixelIndex,
  pixelGroupMap = sim$pixelGroupMap,
  currentTime = NULL,
  rasterToMatch = sim$rasterToMatch,
  fireDamageTable = copy(sim$fireDamageTable),
  fireRSORas = sim$fireRSORas,
  fireROSRas = sim$fireROSRas,
  fireCFBRas = sim$fireCFBRas,
  species = copy(sim$species),
  sufficientLight = copy(sim$sufficientLight),
  speciesEcoregion = copy(sim$speciesEcoregion),
  initialB = 10,
  minRelativeB = copy(sim$minRelativeB),
  successionTimestep = 10L,
  doAssertion = getOption("LandR.assertions", TRUE),
  verbose = getOption("LandR.verbose", TRUE)
)

```

```

PeatlandThermokarst(
  thawedPixIDs = copy(sim$thawedPixIDs),
  treedThawedPixelTableSinceLastDisp = copy(sim$treedThawedPixelTableSinceLastDisp),
  wetlands = sim$wetlands,
  cohortData = copy(sim$cohortData),
  pixelGroupMap = sim$pixelGroupMap,
  rasterToMatch = sim$rasterToMatch,
  species = copy(sim$species),
  speciesEcoregion = copy(sim$speciesEcoregion),
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  initialB = 10L,
  inactivePixelIndex = sim$inactivePixelIndex,
  currentTime = NULL,
  successionTimestep = 10L
)

```

Arguments

cohortData A `data.table` with columns: `pixelGroup`, `ecoregionGroup`, `speciesCode`, and optionally `age`, `B`, `mortality`, `aNPPAct`, and `sumB`.

cohortDefinitionCols the columns in `cohortData` that define unique cohorts (default: `c("pixelGroup", "speciesCode", "age")`).

calibrate logical. Determines whether to output `postFirePixelCohortData`. Defaults to `FALSE`

postFireRegenSummary a `data.table` summarizing for which species serotiny/resprouting were activated and in how many pixels, for each year. Only necessary if `calibrate = TRUE`.

treedFirePixelTableSinceLastDisp `data.table` with 3 columns: `pixelIndex`, `pixelGroup`, and `burnTime`. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding `pixelGroup` and time it occurred. Pixel group IDs correspond to the last year's `pixelGroupMap` and not necessarily the `pixelGroupMap` of the `burnTime` year.

rstCurrentBurn raster layer with burnt pixels in `currentTime`. Usually generated by a fire spread module (or collection of modules) like `fireSense`. Values in the raster should contain 0's for unburnt pixels and values > 0 for burn pixels. NA's are assumed to be outside of the study area.

inactivePixelIndex an internal object to `Biomass_core` that track of inactive pixels.

pixelGroupMap A `RasterLayer` with pixel values equal to a pixel group number that corresponds exactly to `pixelGroup` column in `cohortData`.

currentTime numeric. Current simulation time step (as in `time(sim)`)

rasterToMatch A `RasterLayer` objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

species	<p>a <code>data.table</code> with species traits such as longevity, shade tolerance, etc. Must have column <code>speciesCode</code>, with species names/IDs. The following is a list of default trait columns:</p> <ul style="list-style-type: none"> • "species" same as "speciesCode" – species ID name • "speciesCode" • "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits were derived. Not used during the simulation • "firetolerance" – modular <i>relative</i> (to other species) fire tolerance • "growthcurve" and "mortalityshape" – growth curve shape parameters. • "longevity" – maximum species age • "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none") • "resproutprob" – probability of resprouting • "resproutage_min" – minimum age at which species is capable of resprouting • "resproutage_max" – maximum age at which species is capable of resprouting • "seeddistance_eff" – effective dispersal distance • "seeddistance_max" – maximum dispersal distance • "shadetolerance" – <i>relative</i> (to other species) shade tolerance • "sexualmature" – age at sexual maturity Known optional parameters added/needed by some modules (the user may add others for their own modules): • "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB • "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated • "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB • "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermokarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment: • "leaflongevity" • "wooddecayrate" • "leafLignin" • "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.
sufficientLight	<p>a <code>data.frame</code> (not <code>data.table</code>!) containing probability of establishment, given a site's light conditions (X0-X5) for each level of a species shade tolerance (1-5).</p>
speciesEcoregion	<p>A <code>data.table</code> with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: <code>speciesCode</code> and <code>ecoregionGroup</code>, character representation of species</p>

and ecoregion groups respectively, `maxB` the maximum biomass for the species in a given 'ecoregion', `maxANPP` the maximum aboveground net primary productivity and `SEP` the species establishment probability. May contain columns `inflationFactor` (used to adjust `maxB`) and `maxANPPproportion` (used to calculate `maxANPP`).

`initialB` the initial biomass of new cohorts. Defaults to 10. If NULL or NA, initial cohort biomass is calculated as in LANDIS-II Biomass Succession Extension v3.2.1 (Scheller & Miranda, 2015):

$$\text{initialB} = \text{asInteger}(\text{pmin}(\text{maxANPP}, \text{asInteger}(\text{pmax}(1, \text{maxANPP} \times \exp(-1.6 \times \text{sumB} / \text{maxB}))))$$

where `maxANPP` and `maxB_eco` are the maximum ANPP and B parameters of the species in question within the pixel's ecolocation, and `sumB` is the total stand biomass excluding cohorts with ages less than `successionTimestep`.

`successionTimestep` The time between successive seed dispersal events. In LANDIS-II, this is called "Succession Timestep".

`doAssertion` A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: `getOption("LandR.assertions", TRUE)`.

`verbose` Controls message output. Defaults to `getOption("LandR.verbose")`

`colsForPixelGroups` character. cohortData columns used to find identical pixels in terms of cohort composition, and group them into `pixelGroups` (default: `c("ecoregionGroup", "speciesCode", "age", "B")`).

`LANDISPM` logical. Should partial mortality be calculated as in LANDIS-II Dynamic Fire System v3.0. Must be TRUE for the time being.

`fireDamageTable` data.table defining upper age limit of cohorts killed by fire. See LANDIS-II Dynamic Fire System v3.0 Manual for an example.

`fireRSORas` raster layer of critical spread rate for crowning (m/min) for each burned pixel. If not provided dummy values are used.

`fireROSas` raster layer of equilibrium rate of spread (m/min) for each burned pixel. If not provided dummy values are used.

`fireCFBRas` raster layer of crown fraction burnt for each burned pixel. If not provided dummy values are used.

`minRelativeB` a data.frame with the cut points to classify stand shadiness.

`thawedPixIDs` integer. Vector of pixel IDs than underwent thermokarst (and were converted to wetlands) in the current year (not the last thermokarst event).

`treedThawedPixelTableSinceLastDisp` data.table with 3 columns: `pixelIndex`, `pixelGroup`, and `burnTime`. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding `pixelGroup` and time it occurred. Pixel group IDs correspond to the last year's `pixelGroupMap` and not necessarily the `pixelGroupMap` of the `burnTime` year.

`wetlands` binary SpatRaster with current wetland pixels.

Value

a list of objects to be exported to sim:

- cohortData
- pixelGroupMap
- lastFireYear
- treedFirePixelTableSinceLastDisp
- serotinyResproutSuccessPixels
- severityBMap
- severityData
- postFireRegenSummary (if `calibrate == TRUE` and `!is.null(postFireRegenSummary)`)

Stand-replacing fire disturbances

`FireDisturbance()` simulates post-fire mortality, serotiny and regeneration sequentially occurring after a fire. Post-fire mortality is assumed to be 100% (stand-replacement). The serotiny and regeneration algorithms are based on those in LANDIS-II Biomass Succession extension, v3.2.1, with modifications. Requires the following objects in `sim` (passed as `sim$*`): @template `FireDisturbance`. For any given burnt pixel, the function begins by killing all cohorts (i.e. removing them from `cohortData`). Then it activates serotiny for serotinous species that had been present pre-fire, and reprotouting for reprotouter species. Whether a species successfully regenerates via serotiny or resprouting depends on its traits (whether it is serotinous, resprouter, both or none; species table), pre-fire cohort age (in `cohortData` table), its light/shade requirements (species table) and its local `maxB` (speciesEcoregion) table (see Scheller & Miranda, 2015 and Scheller & Mladenoff, 2004 for details). Unlike in LANDIS-II Biomass Succession extension, v3.2.1, here we allow serotiny and resprouting to occur in the same pixel to reflect the competitive advantage of reprotouters. However, for a given species only serotiny (takes precedence) or resprouting can be activated. The species table must contain the columns:

- `sexualmature` – age at sexual maturity
- `postfireregen` – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- `shadetolerance` – shade tolerance value relative to *other* species.
- `resproutage_min`, `resproutage_max` – minimum and maximum age at which species can resprout
- `resproutprob` – probability of resprouting success (before light/shade suitability is assessed)

Partial severity (i.e. mortality) fire disturbances

`FireDisturbancePM()` simulates partial post-fire mortality, serotiny and regeneration sequentially after a fire. The level of mortality depends of fire severity, and, by default, follows the mechanisms in LANDIS-II Dynamic Fire System v3.0. Serotiny and regeneration algorithms are based on those in LANDIS-II Biomass Succession extension, v3.2.1, with modifications (see `FireDisturbance()`). Requires the following objects in `sim` (and passed as `sim$*`): @template `FireDisturbance`

- `fireDamageTable`

- fireRSORas (Critical spread rate for crowning)
- fireROSRas (Equilibrium Head Fire Rate of Spread)
- fireCFBRas (Crown Fraction Burnt)
- minRelativeB Rasters of fire behaviour properties (fireRSORas, fireROSRas and fireCFBRas) can be calculated using the `cffdrs` package. The species table must contain the columns:
- firetolerance – fire tolerance value relative to *other* species.
- longevity – maximum species age
- sexualmature – age at sexual maturity
- postfireregen – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- shadetolerance – shade tolerance value relative to *other* species.
- resproutage_min, resproutage_max – minimum and maximum age at which species can resprout
- resproutprob – probability of resprouting success (before light/shade suitability is assessed)

Peatland permafrost degradation (thermokarst) disturbances

`PeatlandThermokarst()` simulates tree cohort survival/mortality after peatland permafrost thermokarst. The level of mortality depends on species tolerance to thermokarst, determined by the `thermokarsttol` trait column in the species traits table. At the moment, this level of tolerance is used as the proportion of a cohort biomass that survives (is kept) when a pixel undergoes thermokarst. This is similar to the partial disturbance effects used in LANDIS-II Biomass Harvest v4.0. Requires the following objects in `sim` (and passed as `sim$*`):

- thawedPixIDs
- treedThawedPixelTableSinceLastDisp
- wetlands
- cohortData
- pixelGroupMap
- rasterToMatch
- species
- speciesEcoregion
- inactivePixelIndex The species table must contain the columns:
- thermokarsttol – proportion of cohort biomass (B) that survives thermokarst

References

- Scheller, R.M. & Miranda, B.R. (2015). LANDIS-II Biomass Succession v3.2 Extension – User Guide.
- Scheller, R.M. & Mladenoff, D.J. (2004). A forest growth and biomass module for a landscape simulation model, LANDIS: design, validation, and application. *Ecological Modelling*, 180, 211–229.
- Sturtevant, B.R., Miranda, B.R., Scheller, R.M. & Shinneman, D. (2018). LANDIS-II Dynamic Fire System Extension v3.0 – User Guide.
- Scheller, R.M. & Domingo, J.B. (2021). LANDIS-II Biomass Harvest v4.4 Extension – User Guide

fitNLMModels

FUNCTIONS TO FIT NON-LINEAR MODELS TO ESTIMATE MAXB
Wrapper function to fit non-linear growth model per species.

Description

The function will create sensible parameter ranges for A, k and p parameters of the Chapman-Richards and Logistic growth curves and attempt to run a forward step-AIC procedure to add covariates to the linear component of the model (on the A parameter – the asymptote). The maximum number of covariates to add is determined by maxNoCoefs.

Usage

```
fitNLMModels(
  sp = NULL,
  predictorVarsData,
  sppVarsB,
  predictorVars,
  predictorVarsCombos = NULL,
  maxNoCoefs = 4,
  doFwdSelection = FALSE,
  sampleSize = 3000,
  Ntries = 2000,
  maxCover = 1L,
  models = c("CR", "Logistic"),
  modelOutputsPrev = NULL,
  randomStarts = FALSE,
  lowerBounds = TRUE,
  upperBounds = TRUE,
  nbWorkers = 1L
)
```

Arguments

sp	species name – only used for messaging.
predictorVarsData	a data.table of predictor variables including those in predictorVars and age, as well as pixelIndex. Note that age should be in the original scale (e.g., not logged).
sppVarsB	s data.table of species biomass (B) and pixelIndex.
predictorVars	character vector of predictor variables to be included in the linear component of the model affecting the asymptote (need to correspond to names(predictorVarsData[[sp]])) the same predictors will be considered for all species.
predictorVarsCombos	a list of sets of covariates in predictorVars to add to the fitted models. If this list has several entries with sets of covariates, each will be fitted as part of the model selection process.

maxNoCoefs	how many covariates from predictorVars should be added to the linear component of the model affecting the asymptote? Note that the more covariates are added the longer the model takes to fit, as all combinations are attempted. For 2 or more covariates, only combinations with "cover" are attempted.
doFwdSelection	should covariates be added one at a time to the linear component of the model? If TRUE, and is.null(predictorVarsCombos), then each entry in predictorVarsCombos is used as the set of covariates to test. Otherwise, predictorVarsCombos will be created from combinations of predictorVars, with maxNoCoefs determining the maximum number of covariates to add. If FALSE the full model is fitted.
sampleSize	how many data points should be randomly sampled to fit the model? If NA the full dataset will be used. Note that this may result in long computation times. Biomass data will be binned into 10 regular bins before sampling points in number equal to sampleSize.
Ntries	how many times should the models be fit with new randomly generated starting values? Only used if randomStarts == TRUE.
maxCover	numeric. Value indicating maximum cover/dominance.
models	character vector of models to fit. Only Chapman-Richards ('CR') and 'Logistic' can be chosen at the moment.
modelOutputsPrev	previous outputs of fitNLMmodels. The model will try refitting and comparing AIC with the last results.
randomStarts	logical. Should random starting values of A, k and p non-linear parameters be picked from a range sensible values, or should all combinations of values within this range be used? If FALSE, the default, the starting values are spaced at regular intervals within an acceptable range for each parameter – 20 values for A, 10 for k and p – and all combinations are used (2000 starting values in total). Parameter ranges are estimated from data following Fekedulegn et al. (1999) as follows: <ul style="list-style-type: none"> • range of A starting values (B0 parameter in Fekedulegn et al. 1999) varies between $ObsMaxB \times 0.3$ and $ObsMaxB \times 0.9$, where $ObsMaxB$ is the maximum observed B across the full dataset (not the sampled data for fitting) • k (CR model) and p (Logistic model; both are B2 parameter in Fekedulegn et al. 1999) are estimated as a constant rate to get to $ObsMaxB$, calculated as $\frac{B_{obs2} - B_{obs1}}{ObsMaxB \cdot (\frac{age2 - age1}{age2 \cdot age1})}$, where B1/B2 and age1/age2 are observed values at two points in time. We draw 100 samples of two age values, and corresponding B, to calculate a sample of rates. After excluding rates ≤ 0, we take the minimum and maximum values as the range of k (CR model) or p (Logistic model) parameters of the growth model • the p parameter (CR model) (related to B3 parameter in Fekedulegn et al. 1999) should be > 1. Here we use a range of values between 1.1 and 80 which provided suitable fitting using data from the Northwest Territories, Canada • the k parameter (Logistic model; B1 parameter in Fekedulegn et al. 1999) is estimated as $B_0 = \frac{ObsMaxB}{1+k}$, using a small positive number for B_0, e.g.

2. Here, we estimate k values for B_0 values 1 to 5, and use the minimum and maximum to determine the range from where to draw starting values.

lowerBounds	a named vector of lower parameter boundaries. If FALSE, no lower boundaries are applied. If TRUE, coefficients of the linear model on the A parameter (intercept, cover, k and p) are bound (intercept = observed maximum $B * 0.5$, cover = 0, $k = 0.05$ and $p = 1$). Alternatively, pass a named vector of parameter boundaries.
upperBounds	a named vector of upper parameter boundaries. If FALSE, no lower boundaries are applied. If TRUE, coefficient of the linear model on the A parameter (intercept and k) are bound (intercept = observed maximum $B * 1.5$, $k = 0.2$). Alternatively, pass a named vector of parameter boundaries.
nbWorkers	integer. If > 1, the number of workers to use in <code>future::future_apply</code> , otherwise no parallelisation is done.

See Also

[bbmle::mle2\(\)](#)

generatePixelGroups *Add the correct pixelGroups to a pixelDataTable object*

Description

Generates unique groupings of a `data.table` object where one or more rows can all belong to the same `pixelIndex`. Pixel groups will be identical pixels based on unique combinations of columns.

Usage

```
generatePixelGroups(
  pixelDataTable,
  maxPixelGroup,
  columns = c("ecoregionGroup", "speciesCode", "age", "B")
)
```

Arguments

pixelDataTable	A <code>data.table</code> with column-based descriptions. Must have a column called <code>pixelIndex</code> , which allows for multiple rows to be associated with a single pixel.
maxPixelGroup	A length 1 numeric indicating the current maximum <code>pixelGroup</code> value; the <code>pixelGroup</code> numbers returned will start at <code>maxPixelGroup + 1</code> .
columns	A character vector of column names to use as part of the generation of unique combinations of features. Default is <code>c("ecoregionGroup", "speciesCode", "age", "B")</code>

Value

Returns a vector of `pixelGroup` in the original order of the input `pixelDataTable`. This should likely be added to the `pixelDataTable` object immediately.

genericExtract	<i>Generic extract points</i>
----------------	-------------------------------

Description

Extracts points from raster layers using the original raster layer projection.

Usage

```
genericExtract(x, y, field = NULL, ...)
```

Arguments

x	a raster or polygon object (sp, raster or terra)
y	a points or polygons spatial object (sp, sf, or terra)
field	character. The field(s) to extract when x is a polygon. If NULL, all fields are extracted and returned. IDs of y are always returned (ID column).
...	passed to <code>terra::extract()</code>

Details

If x and y are both polygons, extract often outputs NA due to misalignments (this can happen even when $x == y$), even after `snap(y, x)`. To circumvent this problem, `intersect` is used internally and, if the `extract` argument `fun` is passed, it is applied to values of y per polygon ID of x.

Value

a `data.table` with extracted values and an ID column of y point IDs

genPGsPostDisturbance	<i>Re-generate new pixelGroups in partially disturbed pixels.</i>
-----------------------	---

Description

Re-generate new pixelGroups in partially disturbed pixels.

Usage

```
genPGsPostDisturbance(
  cohortData,
  pixelGroupMap,
  disturbedPixelTable,
  disturbedPixelCohortData,
  colsForPixelGroups = LandR::columnsForPixelGroups(),
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

cohortData	data.table. The pre-disturbance cohortData table
pixelGroupMap	SpatRaster. The pre-disturbance pixelGroupMap.
disturbedPixelTable	data.table. A table with at least the pixelIndex of all disturbed pixels. Additional columns are ignored.
disturbedPixelCohortData	a cohortData-like table with information of dead, and surviving, but <i>NOT</i> regenerating cohorts (cohorts for whom regeneration via, e.g., serotiny or resprouting was successfully activated), in <i>disturbed pixels only</i> . Dead cohorts should age B == 0, surviving cohorts B > 0.
colsForPixelGroups	character. cohortData columns used to find identical pixels in terms of cohort composition, and group them into pixelGroups (default: c("ecoregionGroup", "speciesCode", "age", "B")).
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Details

This function regenerates pixelGroups in situations where disturbances are not stand-replacing and create survivor and dead cohorts in some, but potentially not all, pixels of a pixelGroup. This is necessary to prevent reintroducing dead cohorts that were not affected in the other pixels of the same, original, pixelGroup. **ATTENTION** This function alone will not generate final pixelGroups, and will likely need to be followed by an updateCohortData run. The function:

1. creates a table of undisturbed pixels, and disturbed pixels with dead and surviving cohorts but not new cohorts (e.g. from serotiny/resprouting) – these are added by updateCohortData;
2. generates pixelGroups considering these dead and surviving cohorts – this ensures that we account for cohorts that died in some but not all pixels of a given pixelGroup;
3. then removes dead cohorts (as they should not be in tables for downstream functions like updateCohortData) and recalculates pixelGroups – this ensures that pixels that became similar/dissimilar after the death of some cohorts can form new pixelGroups.

Value

a named list with:

- a cohortData table with the updated pixelGroups, as well as survivor cohorts, but not dead cohorts.
- a pixelGroupMap with the updated pixelGroups in disturbed pixels

getSpeciesTable	<i>Download and prepare a species traits table for use with Biomass_core module</i>
-----------------	---

Description

prepSpeciesTable

Usage

```
getSpeciesTable(url = NULL, dPath = tempdir(), cacheTags = NULL)
```

```
prepSpeciesTable(
  speciesTable,
  speciesLayers = NULL,
  sppEquiv = NULL,
  sppEquivCol = "LandR",
  areas = c("BSW", "BP", "MC")
)
```

Arguments

url	If NULL (the default), uses one from D. Cyr's LANDIS-II files: https://github.com/dcyr/LANDIS-II_IA_generalUseFiles/master/speciesTraits.csv .
dPath	The destination path.
cacheTags	User tags to pass to Cache.
speciesTable	A species traits table, with at least the following columns: <ul style="list-style-type: none"> • speciesCode an character representation of species; • Area a character of geographical area within a species range for which trait values are relevant; • longevity species longevity in years, sexualmature age in years at sexual maturity; • shadetolerance numeric value between 1-5 of relative shade tolerance (with respect to other species); • seeddistance_eff the "effective" seed dispersal distance; • seeddistance_max a numeric with the maximum seed dispersal distance; • mortalityshape and growthcurve: growth curve shape parameters. Other columns (e.g. fire-related traits) can also be included depending on LandR modules in use. Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.
speciesLayers	Deprecated.

sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
areas	A character vector of areas to use. Can be one or more of <code>c("Acadian", "AM", "NorthShore", "BP", "BSE", "BSW", "LSJ", "MC", "PM", "WestON")</code> . If it is more than one, this function will take the minimum value, within a species. These are short versions of the Canada Ecoprovinces. Currently defaults to <code>c("BSW", "BP", "MC")</code> for historical reasons.

Value

`getSpeciesTable()` returns a `data.table` with columns `LandisCode`, `Area`, `Longevity`, `Maturity`, `Shade`, `Fire`, `SeedEffDist`, `SeedMaxDist`, `VegProb`, `MinAgeVeg`, `MaxAgeVeg`, `PostFireRegen`, `LeafLongevity`, `WoodDecayRate`, `MortalityCurve`, `GrowthCurve`, `LeafLignin`, `HardSoft`. See [assertSpeciesTableRaw\(\)](#) for expected column data types.

`prepSpeciesTable()` returns a `data.table` with columns `species`, `Area`, `longevity`, `sexualmature`, `shadetolerance`, `firetolerance`, `seeddistance_eff`, `seeddistance_max`, `resproutprob`, `resproutage_min`, `resproutage_max`, `postfireregen`, `leaflongevity`, `wooddecayrate`, `mortalityshape`, `growthcurve`, `leafLignin`, `hardsoft`. See [assertSpeciesTable\(\)](#) for expected column data types.

Note

This one is tailored to Canadian forests

See Also

[assertSpeciesTableRaw\(\)](#)

[assertSpeciesTable\(\)](#)

getWildfire_NFI

Download and prepare raster fire data from NFI

Description

Download and prepare raster fire data from NFI

Usage

```
getWildfire_NFI(dPath, rasterToMatch, url = NULL)
```


Arguments

dPath	destination path for archive
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
url	location from which to download the wildfire raster layer(s)

Value

a raster with values representing fire year 1985-2015

ggplotMLL_maxB	<i>Plot estimated maximum biomass by age</i>
----------------	--

Description

Plots a maximum biomass estimated at maximum 'cover' (or dominance) levels as a function of age.

Usage

```
ggplotMLL_maxB(
  m11,
  data,
  maxCover = 1L,
  xCovar = "age",
  plotTitle = NULL,
  nonLinModelQuoted,
  linModelQuoted,
  averageCovariates = TRUE,
  observedAge = FALSE,
  plotCIs = TRUE
)
```

Arguments

m11	a named list with outputs of an <code>bbmle::mle2</code> call (the fitted non-linear model), from which coefficient values will be extracted. If several model outputs are provided all fitted models will be plotted, with plot labels corresponding to list names.
data	data for estimation of maximum biomass. Should contain at least an 'age' column. Note that other covariates will be averaged and 'cover' values will be replaced with the maximum cover value (<code>maxCover</code>). If <code>m11</code> is a list data is assumed to be the same for the two models.
maxCover	numeric. Value indicating maximum cover/dominance.
xCovar	the variable shown in the x axis. Defaults to <code>age</code> .

plotTitle	character. Passed to title in <code>ggplot2::labs()</code> .
nonLinModelQuoted	a named list of non-linear equations as a call (quoted expression) passed to <code>mle2(minuslog1)</code> . See <code>bbmle::mle2()</code> . Accepts equations with three parameters 'A', 'p' and 'k'. List names and length must be the same as in <code>m11</code> .
linModelQuoted	A named list of lists of linear equations/modes relating each parameter ('A', 'p' and 'k') with a set of covariates. A call (quoted expression) passed to <code>mle2(..., parameters)</code> . Note that for the purpose of tree growth, the linear equation determining 'A' should include a 'cover' predictor indicating the tree cover or dominance in the stand. Should be scaled between 0 and <code>maxCover</code> . List names and length must be the same as in <code>m11</code> .
averageCovariates	should covariates other than age/cover be averaged for biomass predictions? If not, for each age (at maximum cover) there will be as many predictions as other covariate values. If <code>observedAge == TRUE</code> and <code>averageCovariates == FALSE</code> then the original data is used, with cover changed to <code>maxCover</code> .
observedAge	should observed age values be used, or should these be generated as <code>round(seq(min(age), max(age)*1.5, length.out = 100), 0)</code> ? If <code>observedAge == TRUE</code> and <code>averageCovariates == FALSE</code> then the original data is used, with cover changed to <code>maxCover</code> .
plotCIs	should confidence intervals be calculated and plotted?

installBioSIM

*Install **BioSIM** to retrieve climate and other projections using BioSIM*

Description

Install **BioSIM** to retrieve climate and other projections using BioSIM

Usage

```
installBioSIM(lib)
```

Arguments

`lib` character vector giving the library directories where to install the packages. Recycled as needed. If missing, defaults to the first element of `.libPaths()`.

LANDISDisp

Simulate a LANDIS-II dispersal process on a landscape.

Description

Simulate seed dispersal using user defined function. This is a "receiving pixel" focused dispersal approach. It is the "potentially receiving" cell that looks around itself for potential seed sources. If it finds a single seed source, that passes the probability function described by the `dispersalFn`. If this passes a comparison to a uniform random draw, then the receiving cell is deemed to have a "successful" dispersal for that species. This function can therefore only be used for a relatively specific situation where there is a yes/no returned for each potential receiving cell, i.e., not abundance. This function is also not cumulative, i.e., there is no higher abundance of seeds received if a receiving cell has lots of seed sources around it vs. a single seed source. The difference will come with a higher probability of successfully receiving a "seed".

Usage

```
LANDISDisp(
  dtSrc,
  dtRcv,
  pixelGroupMap,
  speciesTable,
  dispersalFn = Ward,
  b = 0.01,
  k = 0.95,
  plot.it = FALSE,
  successionTimestep,
  verbose = getOption("LandR.verbose", TRUE),
  ...
)
```

Arguments

<code>dtSrc</code>	data.table
<code>dtRcv</code>	data.table
<code>pixelGroupMap</code>	A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to <code>pixelGroup</code> column in <code>cohortData</code> .
<code>speciesTable</code>	A species traits table, with at least the following columns: <ul style="list-style-type: none"> • <code>speciesCode</code> an character representation of species; • <code>Area</code> a character of geographical area within a species range for which trait values are relevant; • <code>longevity</code> species longevity in years, <code>sexualmature</code> age in years at sexual maturity; • <code>shadetolerance</code> numeric value between 1-5 of relative shade tolerance (with respect to other species);

- `seeddistance_eff` the "effective" seed dispersal distance;
- `seeddistance_max` a numeric with the maximum seed dispersal distance;
- `mortalityshape` and `growthcurve`: growth curve shape parameters. Other columns (e.g. fire-related traits) can also be included depending on LandR modules in use. Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

<code>dispersalFn</code>	An expression that can take a "dis" argument. See details. Default is "Ward" (temporarily unused, as it is hard coded inside Rcpp function)
<code>b</code>	LANDIS Ward seed dispersal calibration coefficient (set to 0.01 in LANDIS)
<code>k</code>	LANDIS Ward seed dispersal the probability that seed will disperse within the effective distance (e.g., 0.95)
<code>plot.it</code>	Deprecated. If TRUE, then plot the raster at every interaction, so one can watch the LANDISDisp event grow.
<code>successionTimestep</code>	The time between successive seed dispersal events. In LANDIS-II, this is called "Succession Timestep".
<code>verbose</code>	Controls message output. Defaults to <code>getOption("LandR.verbose")</code>
<code>...</code>	Additional parameters. Currently none.

Details

`dispersalFn` (temporarily unused as code is converted to Rcpp – the default `dispersalFn` is hard coded within the `spiralSeedDispersal` function that uses C++) must be an expression that returns a probability distribution. Because it is a dispersal kernel, it must be a probability distribution. The expression that can take an argument named "dis" (without quotes) as this will be calculated internally and represents the distance from the initial (receiving) pixel and all active pixels within that cluster of active pixels. **LandR** includes the `Ward()` kernel as defined in the LANDIS-II documentation.

Value

A numeric vector of raster pixel indices, in the same resolution and extent as `seedSrc` raster.

Author(s)

Eliot McIntire

Examples

```
if (require("googledrive")) {
  seed <- sample(1e6, 1)
  set.seed(seed)
  library(data.table)

  # keep this here for interactive testing with a larger raster
  rasterTemplate <- reproducible::rasterRead(terra::ext(0, 2500, 0, 2500), res = 100)

  # make a pixelGroupMap
```

```

pgs <- 4 # make even just because of approach below requires even
pixelGroupMap <- SpaDES.tools::randomPolygons(rasterTemplate, numTypes = pgs)
pixelGroupMap[1:100] <- NA # emulate a mask at the start

# Make a receive pixels table -- need pixelGroup and species
nSpecies <- 3
maxNSpeciesPerPixel <- min(5, nSpecies)
rcvSpByPG <- lapply(seq_len(pgs / 2), function(pg) {
  data.table(speciesCode = sample(nSpecies, size = sample(maxNSpeciesPerPixel, 1)))
})
seedReceive <- rbindlist(rcvSpByPG, idcol = "pixelGroup")

# Make a source pixels table -- need pixelGroup and species
srcSpByPG <- lapply(seq_len(pgs / 2), function(pg) {
  data.table(speciesCode = sample(nSpecies, size = sample(maxNSpeciesPerPixel, 1)))
})
seedSource <- rbindlist(srcSpByPG, idcol = "pixelGroup")
# make source pixels not same pixelGroups as receive
seedSource[, pixelGroup := pixelGroup + pgs / 2]

# Get a species table -- if using in Canada, can use this
destPath <- file.path(tempdir(), "ex_LANDISDisp")
speciesTable <- getSpeciesTable(dPath = destPath)
speciesTable <- speciesTable[Area == "BSW"]
speciesTable[, speciesCode := as.factor(LandisCode)]
speciesTable[, seeddistance_eff := SeedEffDist]
speciesTable[, seeddistance_max := SeedMaxDist]

speciesTable <- speciesTable
speciesTable <- data.table(speciesTable)[, speciesCode := seq_along(LandisCode)]
seedReceiveFull <- speciesTable[seedReceive, on = "speciesCode"]
output <- LANDISDisp(
  dtRcv = seedReceiveFull, plot.it = interactive(),
  dtSrc = seedSource,
  speciesTable = speciesTable,
  pixelGroupMap,
  verbose = TRUE,
  successionTimestep = 10
)
# Summarize
output[, .N, by = speciesCode]

## Plot the maps
if (interactive()) {
  library(quickPlot)
  clearPlot()
  spMap <- list()
  spMap$pixelGroupMap <- pixelGroupMap
  for (sppp in unique(output$speciesCode)) {
    spppChar <- paste0("Sp_", sppp)
    spMap[[spppChar]] <- reproducibile::rasterRead(pixelGroupMap)
    ss <- unique(seedSource[speciesCode == sppp], on = c("pixelGroup", "speciesCode"))
    spMap[[spppChar]][pixelGroupMap[] %in% ss$pixelGroup] <- 1
  }
}

```

```

    receivable <- reproducible::rasterRead(pixelGroupMap)
    srf <- unique(seedReceiveFull[speciesCode == sppp], on = c("pixelGroup", "speciesCode"))
    receivable[pixelGroupMap[] %in% srf$pixelGroup] <- 1

    forest <- which(!is.na(pixelGroupMap[]))
    src <- which(!is.na(spMap[[spppChar]][]))
    recvable <- which(!is.na(receivable[]))
    rcvd <- output[speciesCode == sppp][["pixelIndex"]]

    spMap[[spppChar]][forest] <- 0
    spMap[[spppChar]][recvable] <- 2
    spMap[[spppChar]][src] <- 1
    spMap[[spppChar]][rcvd] <- 3
    spMap[[spppChar]][intersect(src, rcvd)] <- 4

    levels(spMap[[spppChar]]) <- data.frame(
      ID = 0:4,
      type = c(
        "OtherForest", "Source", "Didn't receive",
        "Received", "Src&Rcvd"
      )
    )
  }
  Plot(spMap, cols = "Set2")

  # A summary
  rr <- apply(rast(spMap)[[-1]][] + 1, 2, tabulate)
  rownames(rr) <- raster::levels(spMap[[2]])[[1]][, "type"][1:NROW(rr)]
  # next line only works if there are some places that are both source and potential to receive
  # rr <- rbind(rr, propSrcRcvd = round(rr[5,]/ (rr[5,]+rr[2,]), 2))
}
}

```

lccMapGenerator

Calculate landcover classes based on species cohorts

Description

Convert cohort tData into landcover class based on leading species and density.

1. Classify each pixel as "broadleaf", "mixedwood", or "conifer";
2. Define the openness of each pixel as "dense", "open", or "sparse";
3. Combine these into cover classes for forested cells.

Usage

```
lccMapGenerator(
```

```

    cohortData,
    pixelGroupMap,
    lccTable = NULL,
    deciduousCoverDiscount = 0.8418911,
    vegLeadingProportion = 0.75,
    deciduousSpp = NULL,
    rstLCC
  )

```

Arguments

cohortData	A <code>data.table</code> with columns: <code>pixelGroup</code> , <code>ecoregionGroup</code> , <code>speciesCode</code> , and optionally <code>age</code> , <code>B</code> , <code>mortality</code> , <code>aNPPAct</code> , and <code>sumB</code> .
pixelGroupMap	A <code>RasterLayer</code> with pixel values equal to a pixel group number that corresponds exactly to <code>pixelGroup</code> column in <code>cohortData</code> .
lccTable	<code>data.table</code> (with columns <code>leading</code> and <code>LCC</code>) defining the mappings between leading species and landcover classes.
deciduousCoverDiscount	numeric between 0 and 1 that translates <code>%%</code> cover to <code>%%</code> biomass. It assumes all hardwoods are equivalent; all softwoods are equivalent; and that <code>%%</code> cover of hardwoods will be an overestimate of the <code>%%</code> biomass of hardwoods. Hardwoods in Canada have a much wider canopy than softwoods. E.g., 30% cover of hardwoods might translate to 20% biomass of hardwoods. NB: the default <code>deciduousCoverDiscount</code> value was estimated from NWT data in March 2020, and may not be useful for other study areas.
vegLeadingProportion	Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading".
deciduousSpp	character vector of deciduous species names. If not supplied, will attempt to match deciduous species names in sppEquivalencies_CA to those in <code>cohortData</code> .
rstLCC	landcover raster

Details

By default, landcover classes are coded to match the Land Cover Map of Canada 2005 at 250m (Latifovic & Pouliot, 2005).

Code modified from https://github.com/tati-micheletti/caribouRSF_NT/blob/main/R/makeLCCfromCohortData.R.

Value

raster of the same type as `pixelGroupMap`, containing LCC values

Note

Sparse classes: Even though the default LCC2005 input does *not* have "mixed sparse" or "broadleaf sparse", these are likely to be produced in simulations, starting from "conifer sparse" sites (i.e.

broadleaf trees might grow on these sites, and eventually convert these into mixed or broadleaf sparse sites).

Author(s)

Tati Micheletti and Alex Chubaty

References

Latifovic, R. & Pouliot, D. (2005). Multitemporal land cover mapping for Canada: methodology and products. *Canadian Journal of Remote Sensing*, 31, 347–363.

loadCASFRI

Load CASFRI data

Description

TODO: description needed

Usage

```
loadCASFRI(
  CASFRIRas,
  attrFile,
  headerFile,
  sppEquiv,
  sppEquivCol,
  type = c("cover", "age")
)
```

Arguments

CASFRIRas	TODO: description needed
attrFile	TODO: description needed
headerFile	TODO: description needed
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
type	Character string. Either "cover" or "age".

Value

TODO: description needed

loadkNNSpeciesLayers *Load kNN species layers from online data repository*

Description

TODO: description needed

Usage

```
loadkNNSpeciesLayers(
  dPath,
  rasterToMatch = NULL,
  studyArea = NULL,
  sppEquiv,
  year = 2001,
  knnNamesCol = "KNN",
  sppEquivCol = "Boreal",
  thresh = 10,
  url = NULL,
  ...
)
```

Arguments

dPath	path to the data directory
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
studyArea	A SpatialPolygons* object used as the principle study region, passed to <code>reproducible::prepInputs()</code>
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
year	which year's layers should be retrieved? One of 2001 (default) or 2011.
knnNamesCol	character string indicating the column in <code>sppEquiv</code> containing kNN species names. Default "KNN" for when <code>sppEquivalencies_CA</code> is used.
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
thresh	the minimum percent cover a species must have (per pixel) to be considered present in the study area. Defaults to 10.
url	the source url for the data, default is KNN 2011 dataset (https://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributes-forests-canada/2011-attributes_attributs-2011/)
...	Additional arguments passed to <code>reproducible::Cache()</code> and <code>equivalentName()</code> . Also valid: <code>outputPath</code> , and <code>studyAreaName</code> .

Value

A raster stack of percent cover layers by species.

```
makeAndCleanInitialCohortData
```

Generate initial cohortData table

Description

Takes a single `data.table` input, which has the following columns in addition to others that will be labelled with species name, and contain percent cover of each:

Usage

```
makeAndCleanInitialCohortData(
  inputDataTable,
  sppColumns,
  imputeBadAgeModel = quote(lme4::lmer(age ~ B * speciesCode + cover * speciesCode + (1 |
    initialEcoregionCode))),
  minCoverThreshold,
  doAssertion = getOption("LandR.assertions", TRUE),
  doSubset = TRUE
)
```

Arguments

`inputDataTable` A `data.table` with columns described above.

`sppColumns` A vector of the names of the columns in `inputDataTable` that represent percent cover by species, rescaled to sum up to 100%%.

`imputeBadAgeModel` statistical model used to impute ages in pixels with missing data or with cover == 0. If set to NULL no imputation will be attempted, and pixels with missing age are excluded.

`minCoverThreshold` minimum total cover percentage necessary to consider the pixel vegetated, or a cohort present in a pixel.

`doAssertion` A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: `getOption("LandR.assertions", TRUE)`.

`doSubset` Turns on/off subsetting. Defaults to TRUE.

Details

- `pixelIndex` (integer)
- `age` (integer)
- `logAge` (numeric)

- initialEcoregionCode (factor)
- totalBiomass (integer)
- lcc (integer)
- rasterToMatch (integer)
- speciesCode (factor)
- cover (integer)
- coverOrig (integer)
- B (integer)

Several data correction/imputation operations are also performed. Namely, age is imputed in pixels where age data is missing (but not cover) and where cover == 0 but age > 0, total biomass is zeroed if age == 0, and age is zeroed if biomass == 0.

Value

a cohortData data.table with attribute "imputedPixID" (a vector of pixel IDs that suffered imputation).

Author(s)

Eliot McIntire

makeBiomassMap	<i>Create biomassMap</i>
----------------	--------------------------

Description

This is a function that creates the biomassMap raster used for simulations in Biomass_core module, using estimated data based on rawBiomassMap contained in pixelCohortData.

Usage

```
makeBiomassMap(pixelCohortData, rasterToMatch)
```

Arguments

pixelCohortData	The full cohortData data.table
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

The biomassMap, a raster of total stand biomass per pixel.

makeCohortDataFiles *Make the cohortData table, while modifying the temporary pixelCohortData that will be used to prepare other files.*

Description

Takes a pixelCohortData table (see makeAndCleanInitialCohortData), the speciesEcoregion list and returns a modified pixelCohortData and the cohortData tables to be used in the simulation. This function mainly removes unnecessary columns from pixelCohortData, subsets pixels with biomass > 0, generates pixelGroups, and adds ecoregionGroup and totalBiomass columns to pixelCohortData. cohortData is then created by subsetting unique combinations of pixelGroup and whatever columns are listed in columnsForPixelGroups. The resulting cohortData table has the following columns:

- speciesCode (factor)
- ecoregionGroup (factor)
- pixelGroup (integer)
- age (integer)
- B (integer)

Usage

```
makeCohortDataFiles(
  pixelCohortData,
  columnsForPixelGroups,
  speciesEcoregion,
  pixelGroupBiomassClass,
  pixelGroupAgeClass,
  minAgeForGrouping = 0,
  rmImputedPix = FALSE,
  imputedPixID,
  pixelFateDT
)
```

Arguments

pixelCohortData The full cohortData data.table

columnsForPixelGroups Default columns that define pixel groups

speciesEcoregion A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species

in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).

pixelGroupBiomassClass

Integer. When assigning pixelGroup membership, this defines the resolution of biomass that will be considered 'the same pixelGroup', e.g., if it is 100, then 5160 and 5240 will be the same

pixelGroupAgeClass

Integer. When assigning pixelGroup membership, this defines the resolution of ages that will be considered 'the same pixelGroup', e.g., if it is 10, then 6 and 14 will be the same.

minAgeForGrouping

Minimum age for regrouping. This may be because there is a source of ages for young stands/trees that is very reliable, such as a fire database. Ages below this will not be grouped together. Defaults to -1, meaning treat all ages equally. If this is related to known ages from a high quality database, then use age of the oldest trees in that database.

rmImputedPix Should imputed pixels be removed

imputedPixID a vector of IDs of pixels that suffered data imputation.

pixelFateDT A data.table of pixelFateDT; if none provided, will make an empty one.

Value

A list with a modified pixelCohortData, cohortData, and pixelFateDT data.tables.

makeDummyEcoregionMap *Create dummy inputs for test simulations*

Description

ecoregionMap is a raster of all the unique groupings.

Usage

```
makeDummyEcoregionMap(rasterToMatch)
```

```
makeDummyRawBiomassMap(rasterToMatch)
```

```
makeDummyStandAgeMap(rawBiomassMap)
```

```
makeDummyRstLCC(rasterToMatch)
```

```
makeDummyEcoregionFiles(ecoregionMap, rstLCC, rasterToMatch)
```

Arguments

rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
rawBiomassMap	a rawBiomassMap (e.g. the one used throughout the simulation)
ecoregionMap	The ecoregionMap, a raster of all the unique ecoregion groupings, which group together pixels of similar biophysical characteristics.
rstLCC	raster of land-cover class, e.g., LCC2005

Details

rawBiomassMap is a raster of "raw" total stand biomass per pixel, with values between 100 and 20000 g/m².

standAgeMap is a raster of stand age per pixel (where biomass exists) with values between 1 and 300 years.

rstLCC is a raster land-cover class per pixel, with values between 1 and 5 that have no correspondence to any real land-cover classes.

ecoregionFiles uses dummy versions of ecoregionMap and rstLCC to create a list with two objects: the ecoregionMap and a table summarizing its information per pixelID. See ecoregionProducer (it uses ecoregionProducer internally).

Value

a RasterLayer object or, in the case of makeDummyEcoregionFiles, a list.

makeEcoregionDT	<i>Make the ecoregion table</i>
-----------------	---------------------------------

Description

This function creates a table containing pixel-wise ecoregion codes and whether they are "active" (have biomass > 0) or not for simulation. Unlike ecoregionProducer, this function creates the ecoregion table from pixel information contained in pixelCohortData

Usage

```
makeEcoregionDT(pixelCohortData, speciesEcoregion)
```

Arguments

pixelCohortData	The full cohortData data.table
-----------------	--------------------------------

speciesEcoregion

A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).

Value

A data.table with ecoregion codes and their active status per pixelID.

makeEcoregionMap	<i>Make the ecoregionMap raster</i>
------------------	-------------------------------------

Description

Creates a raster of ecoregion codes per pixel. Unlike ecoregionProducer, this fills the raster with pixel information contained in pixelCohortData.

Usage

```
makeEcoregionMap(ecoregionFiles, pixelCohortData)
```

Arguments

ecoregionFiles A list with two objects: the ecoregionMap and a table summarizing its information per pixelID.

pixelCohortData
The full cohortData data.table

Value

A raster with ecoregion codes.

makeMinRelativeB	Create minRelativeB table
------------------	---------------------------

Description

The table contains expert-based values for minimum relative biomass of each shade tolerance class (the minimum relative biomass a cohort with a given shade tolerance should have to be able to germinate), in each unique ecoregion group. All ecoregion groups currently have the same values.

Usage

```
makeMinRelativeB(pixelCohortData)
```

Arguments

pixelCohortData
The full cohortData data.table

Value

a data.frame of min relative biomass values per ecoregion group.

makePickellStack	makePickellStack
------------------	------------------

Description

TODO: description and title needed

Usage

```
makePickellStack(PickellRaster, sppEquiv, sppEquivCol, destinationPath)
```

Arguments

PickellRaster TODO: description needed

sppEquiv table with species name equivalencies between the kNN and final naming formats. See data("sppEquivalencies_CA", "LandR"). For functions that have mixedType, this only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use data("sppEquivalencies_CA", "LandR").

sppEquivCol the column name to use from sppEquiv. For functions that have mixedType, only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use "Boreal".

destinationPath
path to data directory where objects will be downloaded or saved to

Value

TODO: description needed

makePixelGroupMap	<i>Create makePixelGroupMap</i>
-------------------	---------------------------------

Description

Create the makePixelGroupMap raster containing pixelGroups in pixelCohortData.

Usage

```
makePixelGroupMap(pixelCohortData, rasterToMatch)
```

Arguments

pixelCohortData	The full cohortData data.table
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

a raster with pixel groups

makePixelTable	<i>Make pixelTable from biomass, age, land-cover and species cover data</i>
----------------	---

Description

Make pixelTable from biomass, age, land-cover and species cover data

Usage

```
makePixelTable(
  speciesLayers,
  standAgeMap,
  ecoregionFiles,
  biomassMap,
  rasterToMatch,
  rstLCC,
  printSummary = TRUE,
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

speciesLayers	A RasterStack or RasterLayer that should contain species cover data in the study area
standAgeMap	A rasterLayer of stand ages with attribute "imputedPixID"
ecoregionFiles	A list with two objects: the ecoregionMap and a table summarizing its information per pixelID. See ecoregionProducer.
biomassMap	raster of total stand biomass in t/ha. Biomass units are converted to g/m ² .
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
rstLCC	raster of land-cover class, e.g., LCC2005
printSummary	Logical. If TRUE, the default, a print out of the summary(pixelTable) will occur.
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Value

A data.table as many rows as non-NA pixels in rasterToMatch and the columns containing pixel data from the input raster layers, with biomass in g/m².

makeSpeciesEcoregion *Create speciesEcoregion*

Description

Use statistically estimated maxB, maxANPP and establishment probabilities to generate speciesEcoregion table.

Usage

```
makeSpeciesEcoregion(
  cohortDataBiomass,
  cohortDataShort,
  cohortDataShortNoCover,
  species,
  modelCover,
  modelBiomass,
  successionTimestep,
  currentYear
)
```

Arguments

cohortDataBiomass

a subset of cohortData object

cohortDataShort

a subset of cohortData

cohortDataShortNoCover

a subset of cohortData

species

a data.table with species traits such as longevity, shade tolerance, etc. Must have column speciesCode, with species names/IDs. The following is a list of default trait columns:

- "species" same as "speciesCode" – species ID name
- "speciesCode"
- "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits were derived. Not used during the simulation
- "firetolerance" – modular *relative* (to other species) fire tolerance
- "growthcurve" and "mortalityshape" – growth curve shape parameters.
- "longevity" – maximum species age
- "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- "resproutprob" – probability of resprouting
- "resproutage_min" – minimum age at which species is capable of resprouting
- "resproutage_max" – maximum age at which species is capable of resprouting
- "seeddistance_eff" – effective dispersal distance
- "seeddistance_max" – maximum dispersal distance
- "shadetolerance" – *relative* (to other species) shade tolerance
- "sexualmature" – age at sexual maturity Known optional parameters added/needed by some modules (the user may add others for their own modules):
- "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB
- "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated
- "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB
- "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermokarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment:
- "leaflongevity"
- "wooddecayrate"
- "leafLignin"
- "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

modelCover statistical model of species presence/absence
 modelBiomass statistical model of species biomass
 successionTimestep
 The time between successive seed dispersal events. In LANDIS-II, this is called
 "Succession Timestep".
 currentYear time(sim)

Details

See Details.

establishprob

This section takes the cover as estimated from the mature tree cover and partitions it between resprouting and seeds. Unfortunately, establishment by seed is not independent of resprouting, i.e., some pixels would have both. Since we don't know the level of independence, we can't correctly assess how much to discount the two. If there is resprouting > 0, then this is the partitioning: $establishprob = f(establishprob + resproutprob + jointEstablishProbResproutProb)$. If $jointEstablishProbResproutProb$ is 0, then these are independent events and the total cover probability can be partitioned easily between seeds and resprout. This is unlikely ever to be the case. We are picking 50% overlap as a number that is better than 0 (totally independent probabilities, meaning no pixel has both seeds and resprout potential) and 100% overlap (totally dependent probabilities, i.e., every pixel where there is seeds will also be a pixel with resprouting). This is expressed with the `"* 0.5"` in the code.

#' @return A speciesEcoregion data.table with added columns for parameters maxB, maxANPP and establishprob

<code>makeVegTypeMap</code>	<i>Make a vegetation type map from a stack of species abundances</i>
-----------------------------	--

Description

`makeVegTypeMap` is a wrapper around `vegTypeMapGenerator` that works from a species stack of percent cover. These do not have to sum to 100%.

Downloads the 2011 kNN species cover layers from the Canadian Forestry Service, National Inventory System, for validation.

Usage

```
makeVegTypeMap(speciesStack, vegLeadingProportion, mixed, ...)
```

```
loadkNNSpeciesLayersValidation(
  dPath,
  rasterToMatch,
  studyArea,
  sppEquiv,
```

```

    knnNamesCol = "KNN",
    sppEquivCol,
    thresh = 1,
    url,
    ...
)

prepSpeciesLayers_KNN2011(
  destinationPath,
  outputPath,
  url = NULL,
  studyArea,
  rasterToMatch,
  sppEquiv,
  sppEquivCol,
  thresh = 10,
  ...
)

```

Arguments

speciesStack	A RasterStack of species abundances. This must be one RasterLayer per species.
vegLeadingProportion	Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading".
mixed	Deprecated. See mixedType argument to vegTypeMapGenerator .
...	Additional arguments passed to reproducible::Cache() and equivalentName() .
dPath	path to the data directory
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
studyArea	A SpatialPolygons* object used as the principle study region, passed to reproducible::prepInputs()
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have mixedType, this only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
knnNamesCol	character string indicating the column in sppEquiv containing kNN species names. Default "KNN" for when sppEquivalencies_CA is used.
sppEquivCol	the column name to use from sppEquiv. For functions that have mixedType, only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use "Boreal".
thresh	the minimum number of pixels where the species must have biomass > 0 to be considered present in the study area. Defaults to 1.
url	the source url for the data, passed to reproducible::prepInputs()
destinationPath	path to data directory where objects will be downloaded or saved to
outputPath	path to output directory

Value

A factor raster

A raster stack of percent cover layers by species.

minRelativeBDefaults *minRelativeB defaults for Western Boreal Forest Canada*

Description

minRelativeB defaults for Western Boreal Forest Canada

Usage

minRelativeBDefaults()

modifySpeciesAndSpeciesEcoregionTable
Update species and speciesEcoregion tables

Description

Update species and speciesEcoregion tables

Usage

modifySpeciesAndSpeciesEcoregionTable(speciesEcoregion, speciesTable)

Arguments

speciesEcoregion

A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).

speciesTable

A species traits table, with **at least** the following columns:

- speciesCode an character representation of species;
- Area a character of geographical area within a species range for which trait values are relevant;

- `longevity` species longevity in years, sexualmature age in years at sexual maturity;
- `shadetolerance` numeric value between 1-5 of relative shade tolerance (with respect to other species);
- `seeddistance_eff` the "effective" seed dispersal distance;
- `seeddistance_max` a numeric with the maximum seed dispersal distance;
- `mortalityshape` and `growthcurve`: growth curve shape parameters. Other columns (e.g. fire-related traits) can also be included depending on LandR modules in use. Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

Value

a list with modified `speciesEcoregion` and `speciesTable`, under the names "newSpeciesEcoregion" and "newSpeciesTable", respectively.

NAcover2zero	<i>Convert NA values in speciesLayers to zeros</i>
--------------	--

Description

Pixels that are NA but are inside `rasterToMatch` may need to be converted to 0, as they can could still potentially be forested

Usage

```
NAcover2zero(speciesLayers, rasterToMatch)
```

Arguments

<code>speciesLayers</code>	A <code>RasterStack</code> or <code>RasterLayer</code> that should contain species cover data in the study area
<code>rasterToMatch</code>	A <code>RasterLayer</code> objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

the `speciesLayers` with 0 in pixels that had NA

nonForestedPixels	<i>Assess non-forested pixels based on species cover data and land-cover</i>
-------------------	--

Description

Assess non-forested pixels based on species cover data and land-cover

Usage

```
nonForestedPixels(
  speciesLayers,
  omitNonTreedPixels,
  forestedLCCClasses,
  rstLCC
)
```

Arguments

speciesLayers	A RasterStack or RasterLayer that should contain species cover data in the study area
omitNonTreedPixels	logical. Should pixels with classes in forestedLCCClasses be included as non-forested?
forestedLCCClasses	vector of non-forested land-cover classes in rstLCC
rstLCC	raster of land-cover class, e.g., LCC2005

Value

a logical vector of length ncell(rstLCC) where TRUE indicates non-forested pixels where there is no species cover data, or a non-forested land-cover class

overlayLCCs	<i>Overlay different LCC data sources</i>
-------------	---

Description

Overlay different LCC data sources

Usage

```

overlayLCCs(
  LCCs,
  forestedList,
  outputLayer,
  NAcondition,
  NNcondition,
  remapTable = NULL,
  classesToReplace,
  availableERC_by_Sp,
  forestEquivalencies = NULL
)

```

Arguments

LCCs	A named list or named RasterStack of layers whose content is Land Cover Class.
forestedList	A named list of same length and names as LCCs indicating which classes in each LCC raster are 'forested', either permanent or transient
outputLayer	A character string that matches one of the named elements in LCCs. This will be the classification system returned.
NAcondition	The condition when a pixel is deemed to be NA. Given as a character string of a vectorized logical statement that will be within the forestEquivalencies table. It should be a set of conditions with == 0, i.e., non-forested. Examples:, e.g., "LCC2005 == 0" or "CC == 0 LCC2005 == 0", where 0 is the non-forested pixels based on converting LCCs and forestedList to 1 and 0.
NNcondition	The 'nearest-neighbour' condition; i.e., the condition when a nearest-neighbour search is done to fill in the pixel with forested type. Given as a character string of a vectorized logical statement that will be parsed within the forestEquivalencies table. It should be a set of conditions with == 0, i.e., non-forested. Examples:, e.g., "LCC2005 == 0" or "CC == 0 LCC2005 == 0", where 0 is the non-forested pixels based on converting LCCs and forestedList to 1 and 0.
remapTable	data.table. This would be for a situation where 2 LCC layers are provided, one has information in a pixel, but not the one which is outputLayer, so this needs a reclassify or remap.
classesToReplace	Passed to <code>convertUnwantedLCC()</code> , for the pixels where NNcondition is TRUE
availableERC_by_Sp	Passed to <code>convertUnwantedLCC()</code> , for the pixels where NNcondition is TRUE. If this is NULL, then it will be created internally with all pixels with: <code>data.table(initialEcoregionCode = LCCs[[outputLayer]][[]])</code>
forestEquivalencies	A data.frame or NULL. If NULL, this function will derive this table automatically from the other arguments. Otherwise, the user must provide a data.frame with <code>length(LCCs) + 1</code> columns, and <code>2 ^ length(LCCs)</code> rows. Currently not used.

Author(s)

Eliot McIntire and Alex Chubaty

overlayStacks	<i>Overlay layers within raster stacks</i>
---------------	--

Description

Overlays rasters of different data resolution by filling in gaps in the highest resolution raster with data available in lowest resolution one. If only high or low resolution data are available, it will use it without attempting to overlay.

Usage

```
overlayStacks(
  highQualityStack,
  lowQualityStack,
  outputFilenameSuffix = "overlay",
  destinationPath
)
```

Arguments

highQualityStack
high quality list/stack of rasters (will be used preferentially)

lowQualityStack
low quality list/stack of rasters (will be used to fill NAs in highQualityStack)

outputFilenameSuffix
file suffix to save raster if there was overlaying. Defaults to "overlay".

destinationPath
path to data directory where objects will be downloaded or saved to

partialggplotMLL_maxB	<i>Partial effect plots of maximum biomass estimates by age</i>
-----------------------	---

Description

Plots the maximum biomass estimates along an age gradient as a function of another target covariate, with all others held at average values.

Usage

```
partialggplotMLL_maxB(
  mll,
  data,
  targetCovar = "cover",
  maxCover = 1,
  fixMaxCover = TRUE,
  xCovar = "age",
  showQuantiles = "allQuantiles",
  plotTitle = NULL,
  nonLinModelQuoted,
  linModelQuoted,
  fun = "mean",
  plotCIs = TRUE
)
```

Arguments

mll	a named list with outputs of an <code>bbmle::mle2</code> call (the fitted non-linear model), from which coefficient values will be extracted. If several model outputs are provided all fitted models will be plotted, with plot labels corresponding to list names.
data	data for estimation of maximum biomass. Should contain at least an 'age' column. Note that other covariates will be averaged and 'cover' values will be replaced with the maximum cover value (<code>maxCover</code>). If <code>mll</code> is a list, <code>data</code> is assumed to be the same for the two models.
targetCovar	the covariate for which variation in <code>maxB</code> values will be shown per age value. Defaults to showing how <code>maxB</code> values change with "cover" at any given age. Age values are generated as <code>round(seq(min(age), max(age)*1.5, length.out = 100), 0)</code> . When <code>targetCovar != "cover"</code> , "cover" may be fixed at <code>maxCover</code> . See <code>fixMaxCover</code> .
maxCover	numeric. Value indicating maximum cover/dominance.
fixMaxCover	logical. If <code>TRUE</code> and <code>targetCovar != "cover"</code> , cover is not averaged and is fixed to <code>maxCover</code> .
xCovar	the variable shown in the x axis. Defaults to "age". When <code>xCovar == "age"</code> the output plots are not true "Partial Effects" plots. Instead, they show the variation in <code>B</code> values across values of <code>targetCovar</code> for each value of age.
showQuantiles	controls whether quantile predictions will be shown. If "allQuantiles", quantile values (5%, 25%, 50%, 75%, 95%) of <code>B</code> will be plotted as blue lines, with their respective asymptote values (quantile values at maximum age) as dashed lines. If "maximum" the 100% quantile will be plotted. If "none", only the 50% quantile line (average prediction) is plotted. Quantiles are always calculated at <code>max(age)</code> .
plotTitle	character. Passed to <code>title</code> in <code>ggplot2::labs()</code>

nonLinModelQuoted	a named list of non-linear equations as a call (quoted expression) passed to <code>mle2(minuslog1)</code> . See <code>?mle</code> . Accepts equations with three parameters 'A', 'p' and 'k'. List names and length must the same as in <code>mll</code> .
linModelQuoted	A named list of lists of linear equations/modes relating each parameter ('A', 'p' and 'k') with a set of covariates. A call (quoted expression) passed to <code>mle2(..., parameters)</code> . Note that for the purpose of tree growth, the linear equation determining 'A' should include a 'cover' predictor indicating the tree cover or dominance in the stand. Should be scaled between 0 and <code>maxCover</code> . List names and length must the same as in <code>mll</code> .
fun	passed to <code>.MLLMaxBPartialPlotData</code> .
plotCIs	should confidence intervals be calculated and plotted?

Details

Note that the original data, not the predicted values is shown.

See Also

[bbmle:mle2\(\)](#)

partitionBiomass	<i>Partition biomass according to cover estimates</i>
------------------	---

Description

This function will partition `totalBiomass` into each cohort. It will discount deciduous cover, if $x < 1$.

Usage

```
partitionBiomass(x = 1, pixelCohortData)
```

Arguments

x	The ratio for deciduous cover:biomass, where conifer cover:biomass = 1
pixelCohortData	A full <code>pixelCohortData</code> object (i.e., not <code>cohortData</code>)

pixelFate	<i>Create or amend data to a pixelFateDT object</i>
-----------	---

Description

Create or amend data to a pixelFateDT object

Usage

```
pixelFate(
  pixelFateDT,
  fate = NA_character_,
  pixelsRemoved = 0,
  runningPixelTotal = NA_integer_
)
```

Arguments

pixelFateDT	A pixelFateDT data. table with 3 columns: fate, pixelsRemoted, and runningPixelTotal.
fate	A character string (length 1) describing in words the change
pixelsRemoved	A numeric indicating how many pixels were removed due to the fate.
runningPixelTotal	an optional numeric with new, running total. If not supplied, it will be calculated from the last row of pixelFateDT runningTotal minus the pixelsRemoved

Value

A pixelFateDT object, updated with one extra row.

plantNewCohorts	<i>Create new cohorts based on provenance table with unique pixelGroup and add to cohortData</i>
-----------------	--

Description

Create new cohorts based on provenance table with unique pixelGroup and add to cohortData

Usage

```
plantNewCohorts(
  newPixelCohortData,
  cohortData,
  pixelGroupMap,
  initialB = 10,
  currentTime,
```

```

    successionTimestep,
    trackPlanting = FALSE
  )

```

Arguments

newPixelCohortData	the cohorts that were harvested
cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
pixelGroupMap	A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to pixelGroup column in cohortData.
initialB	the initial biomass of new cohorts. Defaults to ten.
currentTime	The current simulation time e.g., time(sim).
successionTimestep	The time between successive seed dispersal events. In LANDIS-II, this is called "Succession Timestep".
trackPlanting	adds column that tracks planted cohorts if TRUE

Value

A data.table with a new cohortData

plotLeadingSpecies *Leading species plots*

Description

Plot effects on conifer-to-deciduous or deciduous-to-conifer conversions.

Usage

```

plotLeadingSpecies(
  studyAreaName,
  climateScenario,
  Nreps,
  years,
  outputDir,
  treeSpecies,
  defineLeading = .defineLeading,
  leadingPercentage = 0.8,
  treeType = NULL,
  rasterToMatch
)

```

Arguments

studyAreaName	character string giving the study area name
climateScenario	character string specifying the name of a CIMP6 climate scenario, including SSP, formatted as in ClimateNA, using underscores as separator (e.g., 'CanESM5_SSP370').
Nreps	the number of simulation replicates/run used to produce summary figures. NOTE: mclapply is used internally, so you should set options(mc.cores = nReps) to take advantage of parallel processing.
years	TODO
outputDir	Path specifying the directory to which outputs figures/objects should be saved.
treeSpecies	TODO
defineLeading	TODO
leadingPercentage	TODO
treeType	TODO
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

list of file names corresponding to the figures and/or objects written to disk
list of filepaths corresponding to the images and/or objects written to disk

plotSpatial *Create a ggplot of a raster or sf object.*

Description

Can be used with Spades.core::Plots.

Usage

```
plotSpatial(x, plotTitle, limits = NULL, field = NULL)
```

Arguments

x	SpatRaster, RasterLayer, SpatVector or sf object
plotTitle	character. A title for the plot passed to ggplot::labs(title = plotTitle).
limits	TODO
field	character. If x is sf or SpatVector, a field to plot.

plotVTM

Summary plots of leading vegetation types

Description

Create raster of leading vegetation types and Plot a bar chart summary and a vegetation type map.
NOTE: plot order will follow colors order.

Usage

```
plotVTM(
  speciesStack = NULL,
  vtm = NULL,
  vegLeadingProportion = 0.8,
  sppEquiv,
  sppEquivCol,
  colors,
  title = "Leading vegetation types"
)
```

Arguments

speciesStack	A SpatRaster, RasterStack or RasterBrick of percent-cover-by-species layers.
vtm	An optional vegetation type map (RasterLayer or SpatRaster). If not supplied, will be produced internally by makeVegTypeMap.
vegLeadingProportion	Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading".
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See data("sppEquivalencies_CA", "LandR"). For functions that have mixedType, this only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use data("sppEquivalencies_CA", "LandR").
sppEquivCol	the column name to use from sppEquiv. For functions that have mixedType, only necessary if mixedType == 2. If not provided and mixedType == 2, will attempt to use "Boreal".
colors	Named vector of colour codes, named using species names. NOTE: plot order will follow this order.
title	The title to use for the generated plots.

Author(s)

Eliot McIntire

```
prepEcoregions      Prepare ecoregions objects
```

Description

Prepare ecoregions objects

Usage

```
prepEcoregions(  
  ecoregionRst = NULL,  
  ecoregionLayer,  
  ecoregionLayerField = NULL,  
  rasterToMatchLarge,  
  rstLCCAdj,  
  pixelsToRm,  
  cacheTags  
)
```

Arguments

`ecoregionRst` an optional raster object that could be passed to `sim`, representing ecoregions

`ecoregionLayer` an `sf` polygons object representing ecoregions

`ecoregionLayerField` optional. The field in `ecoregionLayer` that represents ecoregions.

`rasterToMatchLarge` the `rasterToMatchLarge` object from `sim`

`rstLCCAdj` `RasterLayer` representing land cover adjusted for non-forest classes

`pixelsToRm` a vector of pixels to remove

`cacheTags` `UserTags` to pass to cache

```
prepEcozonesRst    Prepare ecozones raster
```

Description

Download, rasterize, crop, mask, and reproject Canadian national ecozones shapefile.

Usage

```
prepEcozonesRst(url, destinationPath, studyArea = NULL, rasterToMatch = NULL)
```

Arguments

url	character. ecozones shapefile url. Default: http://sis.agr.gc.ca/cansis/nsdb/ecostrat/zone/ecozone_shp.zip .
destinationPath	path to data directory where objects will be downloaded or saved to
studyArea	A SpatialPolygons* object used as the principle study region, passed to <code>reproducible::prepInputs()</code>
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).

Value

RasterLayer

prepInputsCanDEM	<i>Get digital elevation map of Canada</i>
------------------	--

Description

Defaults to using 3 arcmin DEM of Canada in lonlat.

Usage

```
prepInputsCanDEM(studyArea, rasterToMatch, destinationPath)
```

Arguments

studyArea	A SpatialPolygons* object used as the principle study region, passed to <code>reproducible::prepInputs()</code>
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
destinationPath	path to data directory where objects will be downloaded or saved to

Value

RasterLayer

```
prepInputsEcoregion
```

Prepare ecoregion table

Description

Get the dummy ecoregion table from LANDIS-II examples.

Usage

```
prepInputsEcoregion(url = NULL, dPath, cacheTags = NULL)
```

Arguments

url	If NULL (the default), uses one from the LANDIS-II project: https://github.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/ecoregion.txt).
dPath	The destination path.
cacheTags	User tags to pass to Cache.

Value

A data.table

```
prepInputsFireYear
```

Create a raster of fire perimeters

Description

Create a raster of fire perimeters

Usage

```
prepInputsFireYear(..., rasterToMatch, fireField = "YEAR", earliestYear = 1950)
```

Arguments

...	Additional arguments passed to reproducible::prepInputs()
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
fireField	field used to rasterize fire polys
earliestYear	the earliest fire date to allow

Value

a SpatRaster layer of fire perimeters with fire year values.

Examples

```

library(terra)
library(reproducible)

opts <- options(
  reproducible.useTerra = TRUE,
  reproducible.rasterRead = "terra::rast"
)

targetCRS <- crs(SpaDES.tools::randomStudyArea())
randomPoly <- SpaDES.tools::randomStudyArea(
  center = vect(cbind(-115, 50), crs = targetCRS),
  size = 1e+7,
)
buffExt <- buffer(randomPoly, 1e+3) |> ext()
ras2match <- rast(res = 10, ext = ext(randomPoly), crs = crs(randomPoly))
ras2match <- rasterize(randomPoly, ras2match)

firePerimeters <- prepInputsFireYear(
  url = paste0(
    "https://cwfis.cfs.nrcan.gc.ca/downloads/",
    "nfdb/fire_poly/current_version/NFDB_poly.zip"
  ),
  destinationPath = file.path(tempdir(), "ex_prepInputsFireYear"),
  rasterToMatch = ras2match,
  earliestYear = 1930
)

if (interactive()) {
  plot(firePerimeters)
  plot(randomPoly, add = TRUE)
}

options(opts)

```

```
prepInputsLCC
```

Simple prepInputs for Canadian LCC data

Description

A wrapper around prepInputs for the Canadian Land Cover Classification product(s).

Usage

```

prepInputsLCC(
  year = 2010,
  destinationPath = asPath("."),
  method = c("ngb", "near"),
  writeTo = NULL,

```

```
    ...
  )
```

Arguments

year Numeric, either 2010 or 2015. See note re: backwards compatibility for 2005.

destinationPath Character string of a directory in which to download and save the file that comes from url and is also where the function will look for archive or targetFile. NOTE (still experimental): To prevent repeated downloads in different locations, the user can also set options("reproducible.inputPaths") to one or more local file paths to search for the file before attempting to download. Default for that option is NULL meaning do not search locally.

method passed to terra::intersect or raster::intersect, and reproducible::prepInputs

writeTo passed to reproducible::prepInputs

... Arguments passed to terra::mask (for maskTo), terra::project (for projectTo) or terra::writeRaster (for writeTo) and not used for cropTo, as well postProcess's rasterToMatch and studyArea arguments (see below). Commonly used arguments might be method, touches, and datatype. If filename is passed, it will be ignored; use writeTo = . If reproducible.gdalwarp = TRUE, then these will be passed to the gdal* functions. See them for details.

Note

As of May 2021, NRCAN no longer provides/hosts the LCC2005 data. A privately hosted version of the data is available to maintain backwards compatibility, but new users/projects should use the 2010 (or newer) data.

prepInputsSpecies *Download and prepare a species traits table for use with Biomass_core module*

Description

TODO: add detailed description

Usage

```
prepInputsSpecies(url = NULL, dPath, cacheTags = NULL)

prepInputsMainInput(url = NULL, dPath = tempdir(), cacheTags = NULL)
```

Arguments

url	If NULL (the default), uses one from the LANDIS-II project: https://github.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt).
dPath	The destination path.
cacheTags	User tags to pass to Cache.

Value

A `data.table` with columns ... TODO

Note

This one is tailored to Canadian forests (?)

```
prepInputsSpeciesEcoregion
      Prepare species ecoregion table
```

Description

Get the dummy ecoregion table from LANDIS-II examples.

Usage

```
prepInputsSpeciesEcoregion(url = NULL, dPath, cacheTags = NULL)
```

Arguments

url	If NULL (the default), uses one from the LANDIS-II project: https://github.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession-dynamic-inputs_test.txt).
dPath	The destination path.
cacheTags	User tags to pass to Cache.

Value

A `data.table`

```
prepInputsStandAgeMap Create standAgeMap
```

Description

Create the standAgeMap raster containing age estimates for pixelCohortData. A separate `reproducible::prepInputs()` call will source Canadian National Fire Data Base data to update ages of recently burned pixels. To suppress this, pass NULL/NA fireURL

Usage

```
prepInputsStandAgeMap(
  ...,
  ageURL = NULL,
  ageFun = "terra::rast",
  maskWithRTM = TRUE,
  method = "bilinear",
  datatype = "INT2U",
  destinationPath = NULL,
  writeTo = NULL,
  firePerimeters = NULL,
  fireURL = paste0("https://cwfis.cfs.nrcan.gc.ca/downloads/nfdb/",
    "fire_poly/current_version/NFDB_poly.zip"),
  fireFun = "terra::vect",
  fireField = "YEAR",
  rasterToMatch = NULL,
  startTime
)
```

Arguments

...	additional arguments passed to <code>reproducible::prepInputs()</code>
ageURL	url where age map is downloaded
ageFun	passed to 'fun' arg of <code>reproducible::prepInputs()</code> of stand age map
maskWithRTM	passed to <code>reproducible::prepInputs()</code> of stand age map
method	passed to <code>reproducible::prepInputs()</code> of stand age map
datatype	passed to <code>reproducible::prepInputs()</code> of stand age map
destinationPath	path to data directory where objects will be downloaded or saved to
writeTo	passed to <code>reproducible::prepInputs()</code> of stand age map
firePerimeters	fire raster layer fire year values.
fireURL	url to download fire polygons used to update age map. If NULL or NA age imputation is bypassed. Requires passing rasterToMatch. Only used if firePerimeters is missing.

<code>fireFun</code>	passed to <code>reproducible::prepInputs()</code> of fire data. Only used if <code>firePerimeters</code> is missing.
<code>fireField</code>	field used to rasterize fire polys. Only used if <code>firePerimeters</code> is missing.
<code>rasterToMatch</code>	A <code>RasterLayer</code> objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
<code>startTime</code>	date of the last fire year to be considered (e.g., start of fire period counting backwards). If missing, last fire year available will be used.

Value

a raster layer stand age map corrected for fires, with an attribute vector of pixel IDs for which ages were corrected. If no corrections were applied the attribute vector is `integer(0)`.

Examples

```
## Not run:
library(SpaDES.tools)
library(terra)
library(reproducible)
randomPoly <- vect(randomStudyArea(size = 1e7))
randomPoly
ras2match <- rast(res = 250, ext = ext(randomPoly), crs = crs(randomPoly))
ras2match <- rasterize(randomPoly, ras2match)
tempDir <- file.path(tempdir(), "ex_prepInputsStandAgeMap")

## NOT USING FIRE PERIMETERS TO CORRECT STAND AGE
## rasterToMatch does not need to be provided, but can be for masking/cropping.
standAge <- prepInputsStandAgeMap(
  destinationPath = tempDir,
  rasterToMatch = ras2match,
  fireURL = NA
) ## or NULL
attr(standAge, "imputedPixID")

## USING FIRE PERIMETERS TO CORRECT STAND AGE
## ideally, get the firePerimeters layer first
firePerimeters <- Cache(prepareInputsFireYear,
  url = paste0(
    "https://cwfis.cfs.nrcan.gc.ca/downloads",
    "/nfdb/fire_poly/current_version/NFDB_poly.zip"
  ),
  destinationPath = tempDir,
  rasterToMatch = ras2match
)

standAge <- prepInputsStandAgeMap(
  destinationPath = tempDir,
  firePerimeters = firePerimeters,
  rasterToMatch = ras2match
)
attr(standAge, "imputedPixID")
```



```

## not providing firePerimeters is still possible, but will be deprecated
## in this case 'rasterToMatch' MUST be provided
standAge <- prepInputsStandAgeMap(
  destinationPath = tempDir,
  rasterToMatch = ras2match
)
attr(standAge, "imputedPixID")

## End(Not run)

```

```
prepInputs_NTEMS_LCC_FAO
```

Obtain an LCC layer for a given year from NTEMS, with forest matching the FAO definition

Description

Obtain an LCC layer for a given year from NTEMS, with forest matching the FAO definition

Usage

```

prepInputs_NTEMS_LCC_FAO(
  year = 2010,
  disturbedCode = 1,
  resampleMethod = "near",
  ...
)

```

Arguments

year	stack of species layers rasters
disturbedCode	value assigned to pixels that are forest per FAO definition but not in LCC year
resampleMethod	method used when resampling LCC layers to match rasterToMatch
...	passed to prepInputs

Value

a SpatRaster with corrected forest pixels

```
prepInputs_NTEMS_Nonforest
```

Reclassify non-flammable pixels that become flammable - herbaceous or shrubby - vegetation

Description

Reclassify non-flammable pixels that become flammable - herbaceous or shrubby - vegetation

Usage

```
prepInputs_NTEMS_Nonforest(  
  rstLCC,  
  endYear = 2019,  
  lccToAdjust = 33,  
  nonforestLCC = c(50, 100),  
  ...  
)
```

Arguments

rstLCC	input lcc layer with bare soil class that may become vegetated
endYear	NTEMS LCC year to use for correcting transition from bare to non-forest
lccToAdjust	lcc values of the bare class
nonforestLCC	allowable lcc values for bare to become
...	non-spatial arguments passed to prepInputs e.g. destinationPath

Value

a SpatRaster with non-flammable pixels corrected if they become flammable non-forest

```
prepRasterToMatch      Create rasterToMatch and rasterToMatchLarge
```

Description

rasterToMatch and rasterToMatchLarge raster layers are created from studyArea and studyAreaLarge polygons (respectively) using a template raster (often rawBiomassMap)

Usage

```

prepRasterToMatch(
  studyArea,
  studyAreaLarge,
  rasterToMatch,
  rasterToMatchLarge,
  destinationPath,
  templateRas,
  studyAreaName,
  cacheTags = NULL
)

```

Arguments

studyArea A SpatialPolygons* object used as the principle study region, passed to `reproducible::prepInputs()`
studyAreaLarge same as `studyArea`, but larger and completely covering it.
rasterToMatch A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
rasterToMatchLarge the rasterToMatchLarge object from `sim`
destinationPath path to data directory where objects will be downloaded or saved to
templateRas a template raster used to make `rasterToMatch` and/or `rasterToMatchLarge`. Must match `studyAreaLarge`.
studyAreaName character string giving the study area name.
cacheTags a character vector of strings to pass to `Cache(userTags)`

```

prepRawBiomassMap      Create rawBiomassMap

```

Description

Create the `rawBiomassMap` raster containing biomass estimates for `pixelCohortData`.

Usage

```

prepRawBiomassMap(studyAreaName, cacheTags, ...)

```

Arguments

studyAreaName character string giving the study area name.
cacheTags a character vector of strings to pass to `Cache(userTags)`
... arguments passed to `reproducible::prepInputs()` and `reproducible::Cache()`.
 If the following arguments are not provided, the following values will be used:

- url: by default, the 2001 kNN stand biomass map is downloaded from the NRCan National Forest Inventory
- useSAcrs and projectTo: FALSE and NA
- method: "bilinear"
- datatype: "INT2U"
- writeTo: suffix("rawBiomassMap.tif", paste0("_", studyAreaName))
- overwrite: TRUE
- userTags: c(cacheTags, "rawBiomassMap")
- omitArgs: c("destinationPath", "targetFile", "userTags", "stable")

Value

a rawBiomassMap raster

`prepSpeciesLayers_KNN` *Prepare species layers*

Description

TODO: description needed

Usage

```
prepSpeciesLayers_KNN(
  destinationPath,
  outputPath,
  url = NULL,
  studyArea,
  rasterToMatch,
  sppEquiv,
  sppEquivCol,
  thresh = 10,
  ...
)
```

```
prepSpeciesLayers_CASFRI(
  destinationPath,
  outputPath,
  url = NULL,
  studyArea,
  rasterToMatch,
  sppEquiv,
  sppEquivCol,
  ...
)
```

```
prepSpeciesLayers_Pickell(  
  destinationPath,  
  outputPath,  
  url = NULL,  
  studyArea,  
  rasterToMatch,  
  sppEquiv,  
  sppEquivCol,  
  ...  
)  
  
prepSpeciesLayers_ForestInventory(  
  destinationPath,  
  outputPath,  
  url = NULL,  
  studyArea,  
  rasterToMatch,  
  sppEquiv,  
  sppEquivCol,  
  ...  
)  
  
prepSpeciesLayers_MBFRI(  
  destinationPath,  
  outputPath,  
  url = NULL,  
  studyArea,  
  rasterToMatch,  
  sppEquiv,  
  sppEquivCol,  
  ...  
)  
  
prepSpeciesLayers_ONFRI(  
  destinationPath,  
  outputPath,  
  url = NULL,  
  studyArea,  
  rasterToMatch,  
  sppEquiv,  
  sppEquivCol,  
  ...  
)
```

Arguments

destinationPath path to data directory where objects will be downloaded or saved to
outputPath TODO: description needed

url	if NULL, the default, use the default source url
studyArea	A SpatialPolygons* object used as the principle study region, passed to <code>reproducible::prepInputs()</code>
rasterToMatch	A RasterLayer objects to use as the template for all subsequent raster operations (i.e., the one used throughout the simulation).
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
thresh	threshold \ If at least one pixel has cover \geq thresh , the species is considered "present". Otherwise the raster is excluded from the output. Defaults to 10.
...	other arguments, used for compatibility with other <code>prepSpeciesLayers</code> functions.

Value

TODO: description needed

randomStudyArea *Create default study areas for use with LandR modules*

Description

This simply re-exports `SpaDES.tools::randomStudyArea`

Usage

```
randomStudyArea(center = NULL, size = 10000, seed = NULL)
```

Arguments

center	SpatialPoints object specifying a set of coordinates and a projection. Default is an area in southern Alberta, Canada.
size	Numeric specifying the approximate size of the area in m ² . Default 1e4.
seed	Numeric indicating the random seed to set internally (useful for ensuring the same study area is produced each time).

replaceAgeInFires	<i>Replace stand age with time since last fire</i>
-------------------	--

Description

Replace stand age with time since last fire

Usage

```
replaceAgeInFires(standAgeMap, firePerimeters, startTime)
```

Arguments

standAgeMap	a raster layer stand age map
firePerimeters	the earliest fire date to allow
startTime	date of the last fire year to be considered (e.g., start of fire period counting backwards). If missing, last fire year available will be used.

Value

a raster layer stand age map corrected for fires, with an attribute vector of pixel IDs for which ages were corrected. If no corrections were applied the attribute vector is integer(0).

Examples

```
## Not run:
randomPoly <- terra::vect(SpaDES.tools::randomStudyArea(size = 1e7))
randomPoly
ras2match <- terra::rast(resolution = 250,
                        ext = terra::ext(randomPoly),
                        crs = terra::crs(randomPoly))
ras2match <- terra::rasterize(randomPoly, ras2match)
tempDir <- file.path(tempdir(), "ex_replaceAgeInFires")

standAge <- reproducible::prepInputsStandAgeMap(
  destinationPath = tempDir,
  rasterToMatch = ras2match,
  fireURL = NA
) ## or NULL
attr(standAge, "imputedPixID")

firePerimeters <- reproducible::Cache(prepareInputsFireYear,
  url = paste0(
    "https://cwfis.cfs.nrcan.gc.ca/downloads",
    "/nfdb/fire_poly/current_version/NFDB_poly.zip"
  ),
  destinationPath = tempDir,
  rasterToMatch = ras2match
)
```

```
standAge <- replaceAgeInFires(standAge, firePerimeters)
attr(standAge, "imputedPixID")

## End(Not run)
```

rescale	<i>Rescale function (as in scales::rescale)</i>
---------	---

Description

This is a simple function copied from the scales package (almost the same).

Usage

```
rescale(x, to)
```

Arguments

x	a numeric vector
to	a numeric vector of length 2. The new range of values.

rmMissingCohorts	<i>Remove missing cohorts from cohortData based on pixelGroupMap</i>
------------------	--

Description

Remove missing cohorts from cohortData based on pixelGroupMap

Usage

```
rmMissingCohorts(
  cohortData,
  pixelGroupMap,
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

Arguments

cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
pixelGroupMap	A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to pixelGroup column in cohortData.
cohortDefinitionCols	the columns in cohortData that define unique cohorts (default: c("pixelGroup", "speciesCode", "age")).
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Value

A list with 2 data.table objects, cohortData and pixelGroupMap, each updated based on missing pixelGroups in the other.

scheduleDisturbance *Test whether disturbance should be scheduled*

Description

Test whether disturbance should be scheduled

Usage

```
scheduleDisturbance(disturbanceLayer, currentYear)
```

Arguments

disturbanceLayer
 a RasterLayer object
 currentYear time of simulation

Value

Logical indicating whether to schedule a disturbance event

Examples

```
## Not run:
doEvent <- scheduleDisturbance(sim$rstCurrentBurn, time(sim), disturbanceType = "Burn")

## End(Not run)
```

speciesEcoregionLatestYear
 Pull out the values from speciesEcoregion table for current time

Description

Pull out the values from speciesEcoregion table for current time

Usage

```
speciesEcoregionLatestYear(speciesEcoregion, currentTime)
```

Arguments

- `speciesEcoregion` A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: `speciesCode` and `ecoregionGroup`, character representation of species and ecoregion groups respectively, `maxB` the maximum biomass for the species in a given 'ecoregion', `maxANPP` the maximum aboveground net primary productivity and `SEP` the species establishment probability. May contain columns `inflationFactor` (used to adjust `maxB`) and `mANPPproportion` (used to calculate `maxANPP`).
- `currentTime` The current simulation time e.g., `time(sim)`.

Value

The `speciesEcoregion` input object, but with data from only one year, the year that is less than or equal to the `currentTime`

`speciesEcoregionStack` *Create Stacks of the speciesEcoregion content*

Description

Each `RasterStack` show raster maps of one of the columns listed in `columns` and each `RasterLayer` will be one species.

Usage

```
speciesEcoregionStack(
  ecoregionMap,
  speciesEcoregion,
  columns = c("establishprob", "maxB", "maxANPP")
)
```

Arguments

- `ecoregionMap` The `ecoregionMap`, a raster of all the unique ecoregion groupings, which group together pixels of similar biophysical characteristics.
- `speciesEcoregion` A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: `speciesCode` and `ecoregionGroup`, character representation of species and ecoregion groups respectively, `maxB` the maximum biomass for the species in a given 'ecoregion', `maxANPP` the maximum aboveground net primary productivity and `SEP` the species establishment probability. May contain columns `inflationFactor` (used to adjust `maxB`) and `mANPPproportion` (used to calculate `maxANPP`).

columns The columns to use in the speciesEcoregion table. Default is c("establishprob", "maxB", "maxANPP")

Value

list of RasterStack or SpatRaster objects

speciesInStudyArea *Get species list in a given study area for a forest in Canada*

Description

speciesInStudyArea defaults to use a url of a dataset uploaded to Google Drive that is from Canadian Forests, but a different factor raster can be passed e.g., from speciesPresentFromKNN.

Usage

```
speciesInStudyArea(
  studyArea,
  url = NULL,
  speciesPresentRas = NULL,
  sppEquivCol = NULL,
  dPath = getOption("reproducible.destinationPath")
)
```

Arguments

studyArea A SpatialPolygons* object used as the principle study region, passed to [reproducible::prepInputs\(\)](#)

url A url to get a speciesPresence raster e.g., from speciesPresentFromKNN

speciesPresentRas A factor raster where the character string is a string of species names, separated by 2 underscores, sorted alphabetically. Can be produced with speciesPresentFromKNN

sppEquivCol An optional column from LandR::sppEquivalencies_CA. If passed the KNN species will be returned according to this naming convention.

dPath Passed to destinationPath in preProcess.

Value

A named list of length 2: speciesRas is a factor RasterLayer and speciesList is a character string containing the unique, sorted species on the speciesRas, for convenience.

speciesPresentFromKNN *Make a species factor raster*

Description

This will download all KNN layers in Forests of Canada, and make a factor raster at resolution provided by res (larger is faster).

Usage

```
speciesPresentFromKNN(
  year = 2011,
  dPath = asPath("."),
  res = 2000,
  minPctCover = 10
)
```

Arguments

year	Default (and only implemented) is 2011. This will download the 2011 KNN data layers
dPath	A character string indicating where to download all the KNN layers
res	The resolution (one dimension, in m) for the resulting raster
minPctCover	An integer indicating what percent cover a species must have in a pixel to be considered present in that pixel.

Value

A SpatRaster object with 2 layers: "speciesPresent" is a factor, with a legend (i.e., it is numbers on a map, that correspond to a legend) and "numberSpecies" which represents the number of species in each pixel.

Examples

```
## Not run:
if (requireNamespace("googledrive", quietly = TRUE)) {
  # Make the dataset
  speciesPresent <- speciesPresentFromKNN(dPath = "~/data/KNN")

  # To upload this:
  speciesPresentRas <- terra::rast(speciesPresent)[[1]]
  fn <- "SpeciesPresentInCanadianForests.tif"
  writeRaster(speciesPresentRas, file = fn)
  zipFn <- gsub(".tif", ".zip", fn)
  zip(files = dir(pattern = fn), zipFn)
  out <- googledrive::drive_put(zipFn)
}
```

```

## Note: previous file (cropped to boreal forest)
## available ot "10j78jJBeha5L6XDBBdWDAfimgNjYc9UD"
driveID <- "1J8fN7c1Zeqjd7yhiDWi13uoCBL80ensF"

## Get species list
sa <- LandR::randomStudyArea(size = 1e11)
species <- LandR::speciesInStudyArea(sa)
}

## End(Not run)

```

speciesTableUpdate *Change species table of parameters/traits*

Description

Changes longevity and shade tolerance values in the species table. Longevity values are changed to follow Burton & Cumming (1995) for the following species:

- *Abies balsamea*;
- *Abies lasiocarpa*;
- *Betula papyrifera*;
- *Larix laricina*;
- *Larix occidentalis*;
- *Picea engelmannii*;
- *Picea glauca*;
- *Picea mariana*;
- *Pinus banksiana*;
- *Pinus contorta*;
- *Pinus resinosa*;
- *Pinus strobus*;
- *Populus balsamifera* v. *balsamifera*;
- *Populus tremuloides*;
- *Pseudotsuga menziesii* var. *glauca*;
- *Pseudotsuga menziesii*;
- *Thuja plicata*;
- *Tsuga heterophylla*;
- *Tsuga mertensiana* x *heterophylla*;

Usage

```
speciesTableUpdate(species, speciesTable, sppEquiv = NULL, sppEquivCol = NULL)
```

Arguments

species

a data.table with species traits such as longevity, shade tolerance, etc. Must have column speciesCode, with species names/IDs. The following is a list of default trait columns:

- "species" same as "speciesCode" – species ID name
- "speciesCode"
- "Area" – inherited from LANDIS-II default table, the Canadian ecoregion from which traits were derived. Not used during the simulation
- "firetolerance" – modular *relative* (to other species) fire tolerance
- "growthcurve" and "mortalityshape" – growth curve shape parameters.
- "longevity" – maximum species age
- "postfireregen" – post-fire regeneration strategy ("serotiny", "resprout" or "none")
- "resproutprob" – probability of resprouting
- "resproutage_min" – minimum age at which species is capable of resprouting
- "resproutage_max" – maximum age at which species is capable of resprouting
- "seeddistance_eff" – effective dispersal distance
- "seeddistance_max" – maximum dispersal distance
- "shadetolerance" – *relative* (to other species) shade tolerance
- "sexualmature" – age at sexual maturity. Known optional parameters added/needed by some modules (the user may add others for their own modules):
- "inflationFactor" – Biomass_speciesParameters module: inflation factor for maxB
- "growthCurveSource" – Biomass_speciesParameters module: how "growthcurve" was estimated
- "mANPPproportion" – Biomass_speciesParameters module: multiplication factor to calculate maxANPP from maxB
- "thermocarsttol" – Biomass_disturbances module: proportion of biomass surviving after thermokarst (i.e. permafrost thaw). Applied equally across cohorts. Parameters inherited from LANDIS-II default table, but not used in LandR at the moment:
- "leaflongevity"
- "wooddecayrate"
- "leafLignin"
- "hardsoft" Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

speciesTable

A species traits table, with **at least** the following columns:

- speciesCode an character representation of species;
- Area a character of geographical area within a species range for which trait values are relevant;
- longevity species longevity in years, sexualmature age in years at sexual maturity;

- shadetolerance numeric value between 1-5 of relative shade tolerance (with respect to other species);
- seeddistance_eff the "effective" seed dispersal distance;
- seeddistance_max a numeric with the maximum seed dispersal distance;
- mortalityshape and growthcurve: growth curve shape parameters. Other columns (e.g. fire-related traits) can also be included depending on LandR modules in use. Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.

sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".

Details

and only for the following `speciesTable$Areas`:

- Boreal Shield West (BSW);
- Boreal Plains (BP);
- Montane Cordillera (MC).

Note that BSW and BP areas correspond more closely to the region considered in Table 2 of Burton & Cumming (1995), while MC will correspond to both tables.

Of the above species, shade tolerance values are changed for *Abies spp*, *Picea spp*, and *Tsuga spp* to better reflect the western boreal.

When different longevity/shade tolerance trait values exist for a given species, the minimum value across Area's (BSW, BP, MC) is kept.

ATTENTION: if none of species in `species` are from BSW, BP or MC area this function will not change any values.

All other species/Area trait values follow Dominic Cyr and Yan Boulanger's trait values (https://raw.githubusercontent.com/dcyr/LANDIS-II_IA_generalUseFiles/master/speciesTraits.csv).

Value

An updated `species data.table`

sppColors *Create species colour vector from a sppEquiv table*

Description

Create species colour vector from a sppEquiv table

Usage

```
sppColors(sppEquiv, sppEquivCol, newVals = NULL, palette)
```

Arguments

sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from sppEquiv. For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
newVals	An optional character vector of extra names to use, e.g., "Mixed".
palette	An RColorBrewer palette, e.g., "Accent". Can get RColorBrewer palette names from <code>rownames(RColorBrewer::brewer.pal.info)</code> .

Value

A named vector of colour codes, where the names are the species names plus any extra names passed with `newVals`.

sppEquivalencies_CA *Table of species name equivalencies for Canadian trees*

Description

A table containing the different species names used across different sources (e.g., LANDIS-II test parameter files and trait tables, the LandR standard, kNN species biomass layers, etc.). Each column refers to a different source or species naming approach. Presently only containing Canadian native tree species, with name equivalencies coming from:

- LANDIS-II test parameter tables (column `LANDIStest_names`; source: <https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/species.txt>);
- LANDIS-II Canada-wide trait table (column `LANDIStraits_names`; source: https://raw.githubusercontent.com/dcyr/LANDIS-II_IA_generalUseFiles/master/speciesTraits.csv);

- LandR family of SpaDES modules (column LandR_names;
- CFS kNN species biomass layers (column KNN_names; source: <http://tree.pfc.forestry.ca/kNN-Species.tar>);
- Canadian Common Attribute Schema for Forest Resource Inventories (column CASFRI_names; source http://www.borealbirds.ca/files/CAS_Document_Final_Mar_2010_ALL_APPENDICES.pdf).

Usage

sppEquivalencies_CA

Format

A data.frame with 271 rows and 10 variables:

LANDIS_test species names from LANDIS-II test parameter table

LANDIS_traits species names from LANDIS-II traits parameter table

LandR species names from LandR modules

KNN species names from kNN datasets

CASFRI species names from CASFRI database

Latin_full accepted species latin names as in <http://theplantlist.org>

EN_generic_short Short version of species' common names in English

EN_generic_full Full species common names in English

Leading Simple common English names used for leading species

Notes additional notes and information

Boreal Species present in the Boreal forests of Canada

Type Whether the species is a deciduous or conifer species

PSP species name from the module ianmseddy/PSP_Clean

BC_Forestry the species code adopted by the Government of British Columbia

FuelClass The fuel class used by the module PredictiveEcology/fireSense#'

Details

Remaining columns have been filled with some other useful ways to name species (e.g., for plotting).

This table is currently used as the default equivalencies table in LandR SpaDES modules, but can also serve as a template to customize species names equivalencies by the user.

sppEquivCheck	<i>Check and expand sppEquiv</i>
---------------	----------------------------------

Description

This will expand a sppEquiv object that is only a vector or only a one-column data.table into a many column data.table, if the columns that are present do not contain ensureColumns.

Usage

```
sppEquivCheck(sppEquiv, ensureColumns = NULL, sppEquivCol = NULL)
```

Arguments

sppEquiv	A character vector or data.table with named column(s). If this data.table does not have columns named ensureColumns, then it will attempt to merge this data.table with sppEquivalencies_CA to get ensureColumns.
ensureColumns	A character vector of column names that must be in sppEquiv. If these are not present, then the function will attempt to merge with sppEquivalencies_CA, so the column name(s) of sppEquiv must match column names in sppEquivalencies_CA.
sppEquivCol	Optional. Column in sppEquivalencies_CA to use for equivalent names when sppEquiv not provided (i.e., when sppEquivalencies_CA is used instead).

Value

A data.table with potentially all columns in sppEquivalencies_CA.

sppHarmonize	<i>Harmonize the three components that bring species into Biomass_** modules</i>
--------------	--

Description

This function will attempt to harmonize many potential issues/conflicts that may arise under different combinations of supplied objects to the three arguments. See manual for details.

Usage

```
sppHarmonize(
  sppEquiv,
  sppNameVector,
  sppEquivCol,
  sppColorVect,
  vegLeadingProportion = 0,
  studyArea,
  dPath = getOption("reproducible.destinationPath")
)
```

Arguments

sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppNameVector	A character vector of species to use. These species must all be from one naming convention, i.e., from one column in the <code>sppEquiv</code> .
sppEquivCol	A character string normally provided from the <code>P(sim)\$sppEquivCol</code> (see manual). If NA, the default, then this will try to determine which column the <code>sppNameVector</code> used and use that. If <code>sppNameVector</code> is NULL, then it will default to "Boreal".
sppColorVect	A named vector of colours to use for plotting. The names must conform with species name convention used (see sppEquivalencies_CA for possible naming conventions) and should also contain a colour for 'Mixed', when a mixed forest type is supported (see <code>vegLeadingProportion</code> parameter in vegTypeGenerator() for details on mixed forest types).
vegLeadingProportion	Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading".
studyArea	A <code>SpatialPolygons*</code> object used as the principle study region, passed to <code>reproducible::prepInputs()</code>
dPath	Passed to <code>speciesInStudyArea</code> (which then passes to <code>preProcess</code>)

Value

Returns a named list with the same names as the arguments. These should likely be assigned to the `sim` object in the module following this function call.

Examples

```
## not run. usage example within module
# sppOuts <- sppHarmonize(sim$sppEquiv, sim$sppNameVector, P(sim)$sppEquivCol)
# sim$sppEquiv <- sppOuts$sppEquiv
# sim$sppNameVector <- sppOuts$sppNameVector
# P(sim)$sppEquivCol <- sppOuts$sppEquivCol
```

standAgeMapGenerator *Produce stand age map based on cohortData*

Description

Produce stand age map based on cohortData

Usage

```
standAgeMapGenerator(
  cohortData,
  pixelGroupMap,
  weight = "biomass",
  doAssertion = getOption("LandR.assertions", FALSE)
)
```

Arguments

cohortData	A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
pixelGroupMap	A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to pixelGroup column in cohortData.
weight	variable by which to weight cohortData ages. one of "biomass" or NA. NA means use max unweighted age.
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Value

raster of the same type as pixelGroupMap.

statsModel	<i>The generic statistical model to run (lmer or glmer)</i>
------------	---

Description

This does a few things including R squared, gets the fitted values. It appears that running the models "as is" without this wrapper does not work with Cache. The return of the model in a list solves this problem. For Caching, the .specialData should be "omitted" via omitArgs, and uniqueEcoregionGroups should not be omitted.

Usage

```
statsModel(modelFn, uniqueEcoregionGroups, sumResponse, .specialData)
```

Arguments

modelFn	A quoted expression of type <code>package::model(Y ~ X, ...)</code> , omitting the data argument. E.g. <code>lme4::glmer(Y ~ X + (X G), family = poisson)</code>
uniqueEcoregionGroups	Unique values of ecoregionGroups. This is the basis for the statistics, and can be used to optimize caching, e.g. ignore .specialData in .omitArgs.
sumResponse	a sum of all the response variable values Also to be used to optimize caching, e.g. ignore .specialData in .omitArgs.
.specialData	The custom dataset required for the model.

subsetDT *Subset a data.table with random subsampling within by groups*

Description

Subset a data.table with random subsampling within by groups

Usage

```
subsetDT(DT, by, doSubset = TRUE, indices = FALSE)
```

Arguments

DT	A data.table
by	Character vector of column names to use for groups
doSubset	Logical or numeric indicating the number of subsamples to use
indices	Logical. If TRUE, this will return vector of row indices only. Defaults to FALSE, i.e., return the subsampled data.table

Examples

```
library(data.table)
dt <- data.table(Lett = sample(LETTERS, replace = TRUE, size = 1000), Nums = 1:100)
dt1 <- subsetDT(dt, by = "Lett", doSubset = 3)
```

sumRastersBySpecies *Function to sum rasters of species layers*

Description

Function to sum rasters of species layers

Usage

```
sumRastersBySpecies(speciesLayers, layersToSum, filenameToSave, newLayerName)
```

Arguments

speciesLayers	A RasterStack or RasterLayer that should contain species cover data in the study area
layersToSum	names/indices of layers to be summed - optional
filenameToSave	file path to save output raster
newLayerName	name of the output raster layer

uniqueCohortDefinition

The columns in a cohortData that define "unique"

Description

If two pixels have identical values in all of these columns, they are the same pixelGroup.

Usage

uniqueCohortDefinition

uniqueSpeciesEcoregionDefinition

Format

An object of class character of length 4.

An object of class character of length 2.

updateCohortData

Add cohorts to cohortData and pixelGroupMap

Description

This is a wrapper for generatePixelGroups, initiateNewCohort and updates to pixelGroupMap via assignment to new pixelIndex values in newPixelCohortData. By running these all together, there is less chance that they will diverge. There are some checks internally for consistency.

Calculate new values for B, add age, then rbindlist this with cohortData.

Usage

```
updateCohortData(
  newPixelCohortData,
  cohortData,
  pixelGroupMap,
  currentTime,
  speciesEcoregion,
  treedFirePixelTableSinceLastDisp = NULL,
  successionTimestep,
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  initialB = 10,
  verbose = getOption("LandR.verbose", TRUE),
  doAssertion = getOption("LandR.assertions", TRUE)
)
```

```
.initiateNewCohorts(
  newPixelCohortData,
  cohortData,
  pixelGroupMap,
  currentTime,
  cohortDefinitionCols = LandR::cohortDefinitionCols(),
  speciesEcoregion,
  successionTimestep,
  initialB = 10
)
```

Arguments

- newPixelCohortData** must be a complete cohortData object with newly created cohorts. They do not have to have pixelGroup values yet; they can be overlapping with cohortData, (i.e., they can be regenerated on empty pixels or on already occupied pixels). Must contain the columns: pixelIndex, speciesCode, ecoregionGroup. The remaining 4 (see cohortData) will be created with 0s.
- cohortData** A data.table with columns: pixelGroup, ecoregionGroup, speciesCode, and optionally age, B, mortality, aNPPAct, and sumB.
- pixelGroupMap** A RasterLayer with pixel values equal to a pixel group number that corresponds exactly to pixelGroup column in cohortData.
- currentTime** The current simulation time e.g., time(sim).
- speciesEcoregion** A data.table with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: speciesCode and ecoregionGroup, character representation of species and ecoregion groups respectively, maxB the maximum biomass for the species in a given 'ecoregion', maxANPP the maximum aboveground net primary productivity and SEP the species establishment probability. May contain columns inflationFactor (used to adjust maxB) and mANPPproportion (used to calculate maxANPP).
- treedFirePixelTableSinceLastDisp** A data.table with at least 2 columns, pixelIndex and pixelGroup. This will be used in conjunction with cohortData and pixelGroupMap to ensure that everything matches correctly.
- successionTimestep** The time between successive seed dispersal events. In LANDIS-II, this is called "Succession Timestep".
- cohortDefinitionCols** the columns in cohortData that define unique cohorts (default: c("pixelGroup", "speciesCode", "age")).
- initialB** the initial biomass of new cohorts. Defaults to 10. If NULL or NA, initial cohort biomass is calculated as in LANDIS-II Biomass Succession Extension v3.2.1 (Scheller & Miranda, 2015):

```

initialB = asInteger(pmin(maxANPP, asInteger(pmax(1, maxANPP \* exp(-1.6 \* sumB / maxB
where maxANPP and maxB_eco are the maximum ANPP and B parameters of the
species in question within the pixel's ecolocation, and sumB is the total stand
biomass excluding cohorts with ages less than successionTimestep.
verbose      Controls message output. Defaults to getOption("LandR.verbose")
doAssertion  A logical indicating whether some internal tests should be run to ensure the func-
tion is running correctly. Default: getOption("LandR.assertions", TRUE).

```

Details

Does the following:

1. add *new cohort* (not survivor) data into cohortData;
2. assign initial B and age for new cohort;
3. assign the new pixelGroup to the pixels that have new cohort;
4. update the pixelGroup map.

Note that if newPixelCohortData is generated after a disturbance it must contain a type column indicating the origin of the cohorts (e.g. "survivor", "serotiny", "resprouting"). "Survivor" cohorts will not be added to the output objects, as they are assumed to be accounted for in the input cohortData and, correspondingly, pixelGroupMap.

Value

A list of length 2, cohortData and pixelGroupMap, with newPixelCohortData inserted.

A data.table with a new rbindlisted cohortData

updateCohortDataPostHarvest

Add cohorts to cohortData and pixelGroupMap

Description

This is a wrapper for generatePixelGroups, initiateNewCohort and updates to pixelGroupMap via assignment to new pixelIndex values in newPixelCohortData. By running these all together, there is less chance that they will diverge. There are some checks internally for consistency.

Usage

```

updateCohortDataPostHarvest(
  newPixelCohortData,
  cohortData,
  pixelGroupMap,
  currentTime,
  speciesEcoregion,
  treedHarvestPixelTable = NULL,

```



```

    successionTimestep,
    provenanceTable,
    trackPlanting = FALSE,
    initialB = 10,
    cohortDefinitionCols = LandR::cohortDefinitionCols(),
    verbose = getOption("LandR.verbose", TRUE),
    doAssertion = getOption("LandR.assertions", TRUE)
)

```

Arguments

<code>newPixelCohortData</code>	must be a complete <code>cohortData</code> object with newly created cohorts. They do not have to have <code>pixelGroup</code> values yet; they can be overlapping with <code>cohortData</code> , (i.e., they can be regenerated on empty pixels or on already occupied pixels). Must contain the columns: <code>pixelIndex</code> , <code>speciesCode</code> , <code>ecoregionGroup</code> . The remaining 4 (see <code>cohortData</code>) will be created with 0s.
<code>cohortData</code>	A <code>data.table</code> with columns: <code>pixelGroup</code> , <code>ecoregionGroup</code> , <code>speciesCode</code> , and optionally <code>age</code> , <code>B</code> , <code>mortality</code> , <code>aNPPAct</code> , and <code>sumB</code> .
<code>pixelGroupMap</code>	A <code>RasterLayer</code> with pixel values equal to a pixel group number that corresponds exactly to <code>pixelGroup</code> column in <code>cohortData</code> .
<code>currentTime</code>	The current simulation time e.g., <code>time(sim)</code> .
<code>speciesEcoregion</code>	A <code>data.table</code> with species-ecoregion-specific species trait values. Ecoregion refers to "ecolocation", a categorical variable grouping sites with similar biophysical characteristics. The table should have at least the following columns: <code>speciesCode</code> and <code>ecoregionGroup</code> , character representation of species and ecoregion groups respectively, <code>maxB</code> the maximum biomass for the species in a given 'ecoregion', <code>maxANPP</code> the maximum aboveground net primary productivity and <code>SEP</code> the species establishment probability. May contain columns <code>inflationFactor</code> (used to adjust <code>maxB</code>) and <code>maxANPPproportion</code> (used to calculate <code>maxANPP</code>).
<code>treedHarvestPixelTable</code>	A <code>data.table</code> with at least 2 columns, <code>pixelIndex</code> and <code>pixelGroup</code> . This will be used in conjunction with <code>cohortData</code> and <code>pixelGroupMap</code> to ensure that everything matches correctly.
<code>successionTimestep</code>	The time between successive seed dispersal events. In LANDIS-II, this is called "Succession Timestep".
<code>provenanceTable</code>	A <code>data.table</code> with three columns: New cohorts are initiated at the <code>ecoregionGroup</code> <code>speciesEcoregion</code> from the corresponding <code>speciesEcoregion</code> listed in the <code>Provenance</code> column
<code>trackPlanting</code>	if true, planted cohorts in <code>cohortData</code> are tracked with TRUE in column 'planted'
<code>initialB</code>	the initial biomass of new cohorts. Defaults to ten, even if NA/NULL is passed.

cohortDefinitionCols	the columns in cohortData that define unique cohorts (default: c("pixelGroup", "speciesCode", "age")).
verbose	Controls message output. Defaults to getOption("LandR.verbose")
doAssertion	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: getOption("LandR.assertions", TRUE).

Details

Does the following:

1. add new cohort data into cohortData;
2. assign initial B and age for new cohort;
3. assign the new pixelGroup to the pixels that have new cohort;
4. update the pixelGroup map.

Value

A list of length 2, cohortData and pixelGroupMap, with newPixelCohortData inserted.

updateSpeciesTable *Customize species trait table values*

Description

Customize species trait table values

Usage

```
updateSpeciesTable(speciesTable, params)
```

Arguments

speciesTable	<p>A species traits table, with at least the following columns:</p> <ul style="list-style-type: none"> • speciesCode an character representation of species; • Area a character of geographical area within a species range for which trait values are relevant; • longevity species longevity in years, sexualmature age in years at sexual maturity; • shadetolerance numeric value between 1-5 of relative shade tolerance (with respect to other species); • seeddistance_eff the "effective" seed dispersal distance; • seeddistance_max a numeric with the maximum seed dispersal distance;
--------------	--

- `mortalityshape` and `growthcurve`: growth curve shape parameters. Other columns (e.g. fire-related traits) can also be included depending on LandR modules in use. Please see the LANDIS-II Biomass Succession Extension v3.2.1 manual (Scheller and Miranda 2015) for further detail.
- `params` A named list (of parameters) of named lists (by species), with species traits overrides (e.g., `list(seeddistance_eff = list(Abie_sp = 25))`).

Author(s)

Alex Chubaty and Ceres Barros

vegTypeGenerator *Generate and add vegetation type column to cohortData*

Description

This function is a simplification of `vegTypeMapGenerator` and instead of generating a map, it adds the vegetation type column to the `cohortData` table.

Usage

```
vegTypeGenerator(
  x,
  vegLeadingProportion = 0.8,
  mixedType = 2,
  sppEquiv = NULL,
  sppEquivCol,
  pixelGroupColName = "pixelGroup",
  doAssertion = getOption("LandR.assertions", TRUE),
  ...
)
```

Arguments

- | | |
|-----------------------------------|---|
| <code>x</code> | A <code>cohortData</code> object |
| <code>vegLeadingProportion</code> | Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading". |
| <code>mixedType</code> | An integer defining whether mixed stands are: not differentiated (0), any kind of species admixture (1), or deciduous mixed with conifer (2; default). |
| <code>sppEquiv</code> | table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> . |
| <code>sppEquivCol</code> | the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal". |

pixelGroupColName Name of the column in pixelGroup to use.
doAssertion A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: `getOption("LandR.assertions", TRUE)`.
... Additional arguments.

Value

x with a new column, 'leading', coding the vegetation type of each group defined by pixelGroupColName

Author(s)

Eliot McIntire, Ceres Barros, Alex Chubaty

Examples

```

library(data.table)
x <- data.table(
  pixelGroup = rep(1:2, each = 2), B = c(100, 200, 20, 400),
  speciesCode = rep(c("Pice_Gla", "Popu_Tre"), 2)
)
vegTypeGenerator(x)

```

vegTypeMapGenerator *Generate vegetation type map*

Description

Generate vegetation type map

Usage

```

vegTypeMapGenerator(x, ...)

## Default S3 method:
vegTypeMapGenerator(x, ..., doAssertion = getOption("LandR.assertions", FALSE))

## S3 method for class 'data.table'
vegTypeMapGenerator(
  x,
  pixelGroupMap,
  vegLeadingProportion = 0.8,
  mixedType = 2,
  sppEquiv = NULL,
  sppEquivCol,
  colors,
  pixelGroupColName = "pixelGroup",
  doAssertion = getOption("LandR.assertions", TRUE),

```

```
    ...
  )
```

Arguments

<code>x</code>	Either a <code>cohortData</code> object or a <code>speciesCover RasterStack/SpatRaster</code>
<code>...</code>	Additional arguments.
<code>doAssertion</code>	A logical indicating whether some internal tests should be run to ensure the function is running correctly. Default: <code>getOption("LandR.assertions", TRUE)</code> .
<code>pixelGroupMap</code>	A <code>RasterLayer</code> with pixel values equal to a pixel group number that corresponds exactly to <code>pixelGroup</code> column in <code>cohortData</code> .
<code>vegLeadingProportion</code>	Numeric between 0-1, determining the relative biomass threshold a species needs to pass to be considered "leading".
<code>mixedType</code>	An integer defining whether mixed stands are: not differentiated (0), any kind of species admixture (1), or deciduous mixed with conifer (2; default).
<code>sppEquiv</code>	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
<code>sppEquivCol</code>	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
<code>colors</code>	A named vector of colour codes. The names MUST match the names of species in <code>cohortData\$speciesCode</code> , plus an optional "Mixed" colour.
<code>pixelGroupColName</code>	Name of the column in <code>pixelGroup</code> to use.

Author(s)

Eliot McIntire, Ceres Barros, Alex Chubaty

Examples

```
library(data.table)
library(terra)
x <- data.table(
  pixelGroup = rep(1:2, each = 2), B = c(100, 200, 20, 400),
  speciesCode = rep(c("Pice_Gla", "Popu_Tre"), 2)
)
pixelGroupMap <- rast(ext(0, 3, 0, 3), res = 1)
pixelGroupMap[] <- sample(1:2, size = 9, replace = TRUE)
vtm <- vegTypeMapGenerator(x, pixelGroupMap = pixelGroupMap)
```

vtm2conifdecid *Create a summaries of vegetation type transitions*

Description

Create a summaries of vegetation type transitions

Plot vegetation type transitions

Usage

```
vtm2conifdecid(vtm, sppEquiv = NULL, sppEquivCol = "LandR", studyArea)
```

```
vegTransitions(vtm, ecoregion, field, studyArea, times, na.rm = FALSE)
```

```
plotVegTransitions(transitions_df)
```

Arguments

vtm	character vector of file paths to vegetation type maps (see vegTypeMapGenerator()).
sppEquiv	table with species name equivalencies between the kNN and final naming formats. See <code>data("sppEquivalencies_CA", "LandR")</code> . For functions that have <code>mixedType</code> , this only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use <code>data("sppEquivalencies_CA", "LandR")</code> .
sppEquivCol	the column name to use from <code>sppEquiv</code> . For functions that have <code>mixedType</code> , only necessary if <code>mixedType == 2</code> . If not provided and <code>mixedType == 2</code> , will attempt to use "Boreal".
studyArea	sf polygons object delineating the area to use for cropping and masking of ecoregion (e.g., <code>studyAreaReporting</code>).
ecoregion	SpatRaster of ecoregion (or other) codes by which to group.
field	character string of the column name in <code>ecoregion</code> to use for grouping.
times	numeric vector of years corresponding to <code>vtm</code> .
na.rm	logical. If TRUE, remove rows with NA values in <code>vegType</code> (they won't appear as a stratum in the alluvial diagram). If FALSE, these NA values will be replaced with "_NA_" so transitions between vegetated and non-vegetated pixels can be visualized.
transitions_df	A data frame with columns <code>pixelID</code> , <code>ecoregion</code> , <code>vegType</code> , and <code>time</code> . (i.e., output of <code>vegTransitions()</code>).

Value

- `vtm2conifdecid()` returns a character vector of file paths to the conifer-deciduous maps.
- `vegTransitions()` returns a data.frame with columns `pixelID`, `ecoregion`, `vegType`, and `time`.
- `plotVegTransitions()` returns a list of ggplot objects, one for each ecoregion.

Note

creating these plots for large landscapes can be computationally intensive (time and memory use).

WardKernel

Ward Dispersal Kernel – vectorized, optimized for speed

Description

A probability distribution used in LANDIS-II.

Usage

```
Ward(dist, cellSize, effDist, maxDist, k, b, algo = 2)
```

Arguments

dist	A vector of distances to evaluate kernel against
cellSize	A numeric, length 1, of the cell resolution (e.g., res(raster))
effDist	A vector of effective distance (parameter in kernel), with same length as dist
maxDist	A vector of maximum distance (parameter in kernel), with same length as dist
k	A parameter in the kernel
b	A parameter in the kernel
algo	Either 1 or 2. 1 is "simpler code" as it uses ifelse; 2 is faster and is default.

Author(s)

Eliot McIntire

Index

- * **datasets**
 - sppEquivalencies_CA, 112
 - uniqueCohortDefinition, 118
 - .MLLMaxBPartialPlotData, 9
 - .MLLMaxBplotData, 10
 - .NAvalueFlag, 12
 - .compareCRS (.compareRas), 6
 - .compareRas, 6
 - .fitNLMwCovariates, 7
 - .getMaxBCoefs, 8
 - .initiateNewCohorts (updateCohortData), 118
 - .interpolateLightProb, 9
 - .libPaths, 58
 - .projectExtent (.NAvalueFlag), 12
 - .stack (.NAvalueFlag), 12
- addNoPixel2CohortData, 13
- addPixels2CohortData, 13
- aggregateRasByDT, 14
- asInt (.NAvalueFlag), 12
- asInteger, 15
- assert1, 15
- assert2 (assert1), 15
- assertCohortData (assert1), 15
- assertCohortDataAttr (assert1), 15
- assertCohortDataERG (assert1), 15
- assertColumns (assert1), 15
- assertERGs (assert1), 15
- assertFireToleranceDif (assert1), 15
- assertPixelCohortData (assert1), 15
- assertPixelCohortDataValid (assert1), 15
- assertPostPartialDist, 20
- assertRepsAllCohortData (assert1), 15
- assertRstLCChange (assert1), 15
- assertSpeciesEcoregionCohortDataMatch (assert1), 15
- assertSpeciesLayers (assert1), 15
- assertSpeciesPlotLabels (assert1), 15
- assertSpeciesTable (assert1), 15
- assertSpeciesTable(), 56
- assertSpeciesTableRaw (assert1), 15
- assertSpeciesTableRaw(), 56
- assertSppMaxBMaxANPP (assert1), 15
- assertSppVectors (assert1), 15
- assertStandAgeMapAttr (assert1), 15
- assertUniqueCohortData (assert1), 15
- assignLightProb, 21
- bbmle::mle2, 8
- bbmle::mle2(), 8, 10, 11, 52, 58, 84
- biomod2::BIOMOD_EnsembleForecasting, 22
- biomod2::BIOMOD_EnsembleForecasting(), 21
- biomod2::BIOMOD_EnsembleModeling, 22, 23
- biomod2::BIOMOD_FormatingData, 24
- biomod2::BIOMOD_Modeling, 23, 24
- biomod2::BIOMOD_Projection, 22, 24
- biomodEnsembleFrcstWrapper, 21
- biomodEnsembleProjMaps, 22
- biomodEnsembleWrapper, 23
- biomodModelingWrapper, 23
- biomodProjWrapper, 24
- BioSIM_extractPoints, 25
- BioSIM_getMPBSLR, 25
- BioSIM_getWindAnnual, 26
- BioSIM_getWindMonthly, 26
- calcSeverityB, 27
- calcSiteShade, 28
- calculateClimateEffect, 28
- CASFRItoSpRasts, 29
- checkSpeciesTraits, 30
- cohortDefinitionCols, 31
- Colors, 32
- columnsForPixelGroups, 33
- convertUnwantedLCC, 33
- convertUnwantedLCC(), 81

- convertUnwantedLCC2, 35
- defineFlammable, 36
- describeCohortData, 36
- doResprouting, 37
- doSerotiny, 39
- dropTerm, 41
- ecoregionProducer, 41
- equivalentName, 42
- equivalentName(), 65, 77
- equivalentNameColumn (equivalentName), 42
- extractMaxB, 42
- fasterizeFromSp, 43
- FireDisturbance, 43
- FireDisturbancePM (FireDisturbance), 43
- fitNLMMModels, 50
- generatePixelGroups, 52
- genericExtract, 53
- genPGsPostDisturbance, 53
- getSpeciesTable, 55
- getSpeciesTable(), 20
- getWildfire_NFI, 56
- ggplot2::labs(), 58, 83
- ggplotMLL_maxB, 57
- installBioSIM, 58
- isInt (.NAvalueFlag), 12
- LANDISDisp, 59
- LandR (LandR-package), 5
- LandR-package, 5
- lccMapGenerator, 62
- loadCASFRI, 64
- loadkNNSpeciesLayers, 65
- loadkNNSpeciesLayersValidation (makeVegTypeMap), 76
- makeAndCleanInitialCohortData, 66
- makeBiomassMap, 67
- makeCohortDataFiles, 68
- makeDummyEcoregionFiles (makeDummyEcoregionMap), 69
- makeDummyEcoregionMap, 69
- makeDummyRawBiomassMap (makeDummyEcoregionMap), 69
- makeDummyRstLCC (makeDummyEcoregionMap), 69
- makeDummyStandAgeMap (makeDummyEcoregionMap), 69
- makeEcoregionDT, 70
- makeEcoregionMap, 71
- makeMinRelativeB, 72
- makeMinRelativeB(), 28
- makePickellStack, 72
- makePixelGroupMap, 73
- makePixelTable, 73
- makeSpeciesEcoregion, 74
- makeVegTypeMap, 76
- minRelativeBDefaults, 78
- modifySpeciesAndSpeciesEcoregionTable, 78
- NAcover2zero, 79
- nonForestedPixels, 80
- options(), 5
- overlayLCCs, 80
- overlayStacks, 82
- partialggplotMLL_maxB, 82
- partitionBiomass, 84
- PeatlandThermokarst (FireDisturbance), 43
- pixelFate, 85
- plantNewCohorts, 85
- plotLeadingSpecies, 86
- plotSpatial, 87
- plotVegTransitions (vtm2conifdecid), 126
- plotVTM, 88
- prepEcoregions, 89
- prepEcozonesRst, 89
- prepInputs_NTEMS_LCC_FAO, 97
- prepInputs_NTEMS_Nonforest, 98
- prepInputsCanDEM, 90
- prepInputsCanDEM(), 25
- prepInputsEcoregion, 91
- prepInputsFireYear, 91
- prepInputsLCC, 92
- prepInputsMainInput (prepInputsSpecies), 93
- prepInputsSpecies, 93
- prepInputsSpeciesEcoregion, 94
- prepInputsStandAgeMap, 95
- prepRasterToMatch, 98

- prepRawBiomassMap, 99
- prepSpeciesLayers_CASFRI
 - (prepSpeciesLayers_KNN), 100
- prepSpeciesLayers_ForestInventory
 - (prepSpeciesLayers_KNN), 100
- prepSpeciesLayers_KNN, 100
- prepSpeciesLayers_KNN2011
 - (makeVegTypeMap), 76
- prepSpeciesLayers_MBFRI
 - (prepSpeciesLayers_KNN), 100
- prepSpeciesLayers_ONFRI
 - (prepSpeciesLayers_KNN), 100
- prepSpeciesLayers_Pickell
 - (prepSpeciesLayers_KNN), 100
- prepSpeciesTable (getSpeciesTable), 55
- prepSpeciesTable(), 20

- randomStudyArea, 102
- raster::intersect, 93
- raster::mask(), 36
- reclass (.NAvalueFlag), 12
- replaceAgeInFires, 103
- reproducible::Cache, 22–24
- reproducible::Cache(), 65, 77, 99
- reproducible::postProcess(), 36
- reproducible::prepInputs, 93
- reproducible::prepInputs(), 65, 77, 90, 91, 95, 96, 99, 102, 107, 115
- rescale, 104
- rmMissingCohorts, 104

- scheduleDisturbance, 105
- speciesEcoregionLatestYear, 105
- speciesEcoregionStack, 106
- speciesInStudyArea, 107
- speciesPresentFromKNN, 108
- speciesTableUpdate, 109
- sppColors, 112
- sppEquivalencies_CA, 20, 31, 63, 112, 115
- sppEquivCheck, 114
- sppHarmonize, 114
- standAgeMapGenerator, 115
- statsModel, 116
- subsetDT, 117
- sumRastersBySpecies, 117

- terra::compareGeom(), 6
- terra::extract(), 53
- terra::intersect, 93

- uniqueCohortDefinition, 118
- uniqueSpeciesEcoregionDefinition
 - (uniqueCohortDefinition), 118
- updateCohortData, 118
- updateCohortDataPostHarvest, 120
- updateSpeciesTable, 122

- vegTransitions (vtm2conifdecid), 126
- vegTypeGenerator, 123
- vegTypeGenerator(), 20, 31, 115
- vegTypeMapGenerator, 77, 124
- vegTypeMapGenerator(), 126
- vtm2conifdecid, 126

- Ward (WardKernel), 127
- Ward(), 60
- WardKernel, 127