LandR Manual

v. 0.0.9000

Edited by: Ceres Barros, Alex M. Chubaty, Ian M. S. Eddy, Eliot J. B. McIntire

Last updated: 2022-03-18

2

Contents

Chapter 1

Preface

LandR is a collection of `SpaDES` modules [see 5] which are aimed at simulating forest dynamics across large spatial scales, while taking into account various disturbances that affected them (e.g., wildfire and climate change), as well as their interactions with other components of forest systems, such as bird communities and ungulates (via changes in the habitat of these species) or carbon cycling.

The present manual is a "live" document, in that will grow and change according to the existing number of LandR modules, as well as their development. In addition, sub-manuals may be produced describing smaller collections of LandR modules – e.g., the LandR Biomass manual describes only the LandR modules that are essential for the simulation of the vegetation components of forest succession, hence excluding fire disturbance modules.

Chapter 2

LandR *Biomass_core* Module

This documentation is work in progress. Potential discrepancies and omissions may exist for the time being. If you find any, do contact us using the link aboveˆˆ

2.0.0.1 Authors:

Yong Luo yluo1@lakeheadu.ca [aut], Eliot J B McIntire eliot.mcintire@canada.ca [aut, cre], Jean Marchal jean.d.marchal@gmail.com [ctb], Alex M. Chubaty achubaty@for-cast.ca [ctb], Ceres Barros cbarros@mail.ubc.ca [ctb]

2.1 Module Overview

2.1.1 Module summary

LandR *Biomass_core* (hereafter *Biomass_core*) is the core forest succession simulation module of the LandR ecosystem of `SpaDES` modules [see 5]. It simulates tree cohort ageing, growth, mortality and competition for light resources, as well as seed dispersal (Fig. 2.1), in a spatially explicit manner and using a yearly time steps. The model is based on the LANDIS-II Biomass Succession Extension v.3.2.1 [LBSE; Scheller and Miranda [16]], with a few changes (see Differences between *Biomass_core* and LBSE). Nonetheless, the essential functioning of the succession model still largely follows its LANDIS-II counterpart,

and we refer the reader to the corresponding LANDIS-II BSE manual [16] for a detailed reading of the mechanisms implemented in the model.
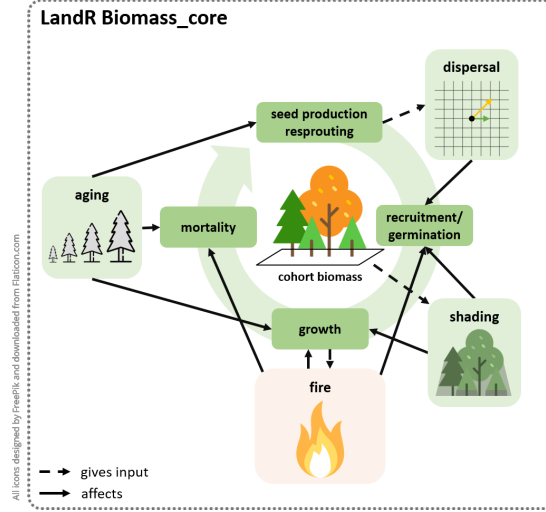
Figure 2.1: *Biomass_core* simulates tree cohort growth, mortality, recruitment and dispersal dynamics, as a function of cohort ageing and competition for light (shading) and space, as well as disturbances like fire (simulated using other modules).

2.1.2   Module inputs and parameters

Biomass_core is capable of running on dummy datasets from which it estimates parameters linked to vegetation growth and seed germination (such as the maximum biomass per species, per pixel, and the probability of seed germination – *i.e.*, species establishment probability not due to resprouting), but also builds and initializes forest communities (based on biomass, age, species composition, land cover and ecological zones like ecodistricts.

Ideally, however, the user should supply realistic versions of these data and the essential initialization objects that *Biomass_core* requires to run.

Table 2.1.2 shows a full list of input objects that *Biomass_core* expects. Of these, the only input that **must** be provided (*i.e.*, *Biomass_core* does not have a default for) is `studyArea`. All other input objects and parameters have internal defaults (see Tables 2.2.3.1 and 2.2.3.2.

List of *Biomass_core* input objects and their description.

objectName

desc

biomassMap

total biomass raster layer in study area (in g/mˆ2), filtered for pixels covered by cohortData. Only used if `P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.

cceArgs

a list of quoted objects used by the `growthAndMortalityDriver calculateClimateEffect` function

cohortData

`data.table` with cohort-level information on age and biomass, by pixelGroup and ecolocation (i.e., `ecoregionGroup`). If supplied, it must have the following columns: `pixelGroup` (integer), `ecoregionGroup` (factor), `speciesCode` (factor), `B` (integer in g/mˆ2), `age` (integer in years)

ecoregion

ecoregion look up table

ecoregionMap

ecoregion map that has mapcodes match ecoregion table and `speciesEcoregion` table. Defaults to a dummy map matching `rasterToMatch` with two regions

lastReg

an internal counter keeping track of when the last regeneration event occurred

minRelativeB

table defining the relative biomass cut points to classify stand shadeness

pixelGroupMap

initial community map that has mapcodes match initial community table

rasterToMatch

a raster of the `studyArea` in the same resolution and projection as `biomassMap`

species

a table that has species traits such as longevity, shade tolerance, etc. Default is partially based on Dominic Cyr and Yan Boulanger's project

speciesEcoregion

table defining the maxANPP, maxB and SEP, which can change with both ecoregion and simulation time. Defaults to a dummy table based on dummy data os biomass, age, ecoregion and land cover class

speciesLayers

percent cover raster layers of tree species in Canada. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps

from 2001 using a cover threshold of 10 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

sppColorVect

A named vector of colors to use for plotting. The names must be in `sim$sppEquiv[[sim$sppEquivCol]]`, and should also contain a color for 'Mixed'

sppEquiv

table of species equivalencies. See `LandR::sppEquivalencies_CA`.

studyArea

Polygon to use as the study area. Must be provided by the user

studyAreaReporting

multipolygon (typically smaller/unbuffered than studyArea) to use for plotting/reporting. Defaults to `studyArea`.

sufficientLight

table defining how the species with different shade tolerance respond to stand shade. Default is based on LANDIS-II Biomass Succession v6.2 parameters

treedFirePixelTableSinceLastDisp

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

Of the above, we draw particular attention to the the following inputs, which are crucial to run *Biomass_core* on a realistic setting (see Input objects section of the manual for further detail):

- Spatial layers: `ecoregionMap`, `studyArea`

- Trait and parameter tables: `ecoregion`, `minRelativeB`, `species`, `speciesEcoregion`, `sufficientLight`, `sppEquiv`, `sppColorVect`

- Cohort-simulation related: `cohortData`, `pixelGroupMap`

For the beginner user, we suggest running *Biomass_core* without supplying any inputs and inspecting the above mentioned objects to understand their structure and format. The user can later either feed these objects via `simInit`, or make a module that makes them and provides necessary inputs to *Biomass_core* (see e.g. *Biomass_borealDataPrep*)

Besides the above mentioned inputs, *Biomass_core* uses several other parameters, which can be changed by the user if need be (Table 2.1.2). Please see the Parameters section of the manual for a list of the most useful parameters.

List of *Biomass_core* parameters and their description.

paramName

paramDesc

calcSummaryBGM

A character vector describing when to calculate the summary of biomass, growth and mortality Currently any combination of 5 options is possible: 'start'- as before vegetation succession events, i.e. before dispersal, 'postDisp' - after dispersal, 'postRegen' - after post-disturbance regeneration (currently the same as 'start'), 'postGM' - after growth and mortality, 'postAging' - after aging, 'end' - at the end of vegetation succesion events, before plotting and saving. The 'end' option is always active, being also the default option.

calibrate

Do calibration? Defaults to `FALSE`

cohortDefinitionCols

`cohortData` columns that determine what constitutes a cohort This parameter should only be modified if additional modules are adding columns to cohortData

cutpoint

A numeric scalar indicating how large each chunk of an internal data.table is, when processing by chunks

gmcsGrowthLimits

if using `LandR.CS` for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model

gmcsMortLimits

if using `LandR.CS` for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model

gmcsMinAge

if using `LandR.CS` for climate-sensitive growth and mortality, the minimum age for which to predict climate-sensitive growth and mortality. Young stands ($<$ 30) are poorly represented by the PSP data used to parameterize the model.

growthAndMortalityDrivers

package name where the following functions can be found: `calculateClimateEffect`, `assignClimateEffect` (see `LandR.CS` for climate sensitivity equivalent functions, or leave default if this is not desired)

growthInitialTime

Initial time for the growth event to occur

initialBiomassSource

Currently, there are three options: 'spinUp', 'cohortData', 'biomassMap'. If 'spinUp', it will derive biomass by running spinup derived from Landis-II. If 'cohortData', it will be taken from the `cohortData` object, i.e., it is already correct, by cohort. If 'biomassMap', it will be taken from `sim$biomassMap`, divided across species using `sim$speciesLayers` percent cover values 'spinUp' uses `sim$standAgeMap` as the driver, so biomass is an output . That means it will be unlikely to match any input information about biomass, unless this is set to 'biomassMap', and a `sim$biomassMap` is supplied. Only the 'cohortData' option is currently active.

keepClimateCols

include growth and mortality predictions in `cohortData`?

minCohortBiomass

cohorts with biomass below this threshold (g/mˆ2) are removed. Not a LANDIS-II BSE parameter.

mixedType

How to define mixed stands: 1 for any species admixture; 2 for deciduous > conifer. See `?LandR::vegTypeMapGenerator`.

plotOverstory

swap max age plot with overstory biomass

seedingAlgorithm

choose which seeding algorithm will be used among 'noDispersal', 'universalDispersal', and 'wardDispersal' (default). See Scheller & Miranda (2015) - Biomass Succession extension, v3.2.1 User Guide

spinupMortalityfraction

defines the mortality loss fraction in spin up-stage simulation. Only used if `P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.

sppEquivCol

The column in `sim$sppEquiv` data.table to use as a naming convention

successionTimestep

defines the simulation time step, default is 10 years. Note that growth and mortality always happen on a yearly basis.

vegLeadingProportion

a number that defines whether a species is leading for a given pixel

.maxMemory

maximum amount of memory (in GB) to use for dispersal calculations.

.plotInitialTime

Vector of length $= 1$, describing the simulation time at which the first plot event should occur. To plotting off completely use `P(sim)$.plots`.

.plotInterval

defines the plotting time step. If `NA`, the default, .plotInterval is set to successionTimestep.

.plots

Passed to `types` in `Plots` (see `?Plots`). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at `end(sim)`. If `NA` plotting is off completely (this includes plot saving).

.plotMaps

Controls whether maps should be plotted or not. Set to `FALSE` if `P(sim)$.plots == NA`

.saveInitialTime

Vector of length $= 1$, describing the simulation time at which the first save event should occur. Set to `NA` if no saving is desired. If not `NA`, then saving will occur at `P(sim)$.saveInitialTime` with a frequency equal to `P(sim)$.saveInterval`

.saveInterval

defines the saving time step. If `NA`, the default, .saveInterval is set to `P(sim)$successionTimestep`.

.studyAreaName

Human-readable name for the study area used. If `NA`, a hash of `studyArea` will be used.

.useCache

Internal. Can be names of events or the whole module name; these will be cached by `SpaDES`

.useParallel

Used only in seed dispersal. If numeric, it will be passed to `data.table::setDTthreads` and should be $<= 2$; If `TRUE`, it will be passed to `parallel::makeCluster`; and if a cluster object, it will be passed to `parallel::parClusterApplyB`.

2.1.3   Events

Events are scheduled as follows:

- Module initiation (`init` event)
- Seed dispersal (every `successionTimestep`; `Dispersal` event)
- Mortality and growth (`mortalityAndGrowth` event)
- Reclassification of age cohorts (every `successionTimestep`; `cohortAgeReclassification` event)
- Summary tables of regeneration (`summaryRegen` event), biomass, age, growth and mortality (`summaryBGM\*` event)
- Plots of maps (`plotMaps` event) and averages (`plotAvgs` and `plotSummaryBySpecies` events)
- Save (`save`)

2.1.4   Module outputs

The module produces the following outputs types. – Plotting – live and/or saved plot objects/images (depending on `.plots`)

– Saved biomass, mortality, leading vegetation raster layers – Whatever objects supplied to `outputs` argument in `simInit`, that are within the `simList` object.

All `simList` objects that are changed by *Biomass_core* (*i.e.*, the definition of a module output) are listed in Table 2.1.4.

List of *Biomass_core* output objects and their description.

objectName

desc

activePixelIndex

internal use. Keeps track of which pixels are active

activePixelIndexReporting

internal use. Keeps track of which pixels are active in the reporting study area

ANPPMap

ANPP map at each succession time step

cohortData

`data.table` with cohort-level information on age, biomass, aboveground primary productivity (year's biomass gain) and mortality (year's biomass loss), by pixelGroup and ecolocation (i.e., `ecoregionGroup`). Contains at least the following columns: `pixelGroup` (integer), `ecoregionGroup` (factor), `speciesCode` (factor), B (integer in g/m^2), `age` (integer in years), `mortality`

(integer in g/mˆ2), `aNPPAct` (integer in g/mˆ2). May have other columns depending on additional simulated processes (i.e., cliamte sensitivity; see, e.g., `P(sim)$keepClimateCols`).

ecoregionMap

ecoregion map that has mapcodes match `ecoregion` table and `speciesEcoregion` table. Defaults to a dummy map matching rasterToMatch with two regions

inactivePixelIndex

internal use. Keeps track of which pixels are inactive

inactivePixelIndexReporting

internal use. Keeps track of which pixels are inactive in the reporting study area

lastFireYear

Year of the most recent fire year

lastReg

an internal counter keeping track of when the last regeneration event occurred

minRelativeB

define the cut points to classify stand shade

mortalityMap

Mortality map at each succession time step

pixelGroupMap

updated community map at each succession time step

regenerationOutput

If `P(sim)$calibrate == TRUE`, an summary of seed dispersal and germination success (i.e., number of pixels where seeds successfully germinated) per species and year.

reproductionMap

Regeneration map at each succession time step

simulatedBiomassMap

Biomass map at each succession time step (in g/mˆ2)

simulationOutput

contains simulation results by ecoregion (main output)

simulationTreeOutput

Summary of several characteristics about the stands, derived from `cohortData`

species

a table that has species traits such as longevity, shade tolerance, etc. Currently obtained from LANDIS-II Biomass Succession v.6.0-2.0 inputs

speciesEcoregion

define the maxANPP, maxB and SEP change with both ecoregion and simulation time

speciesLayers

biomass percentage raster layers by species in Canada species map

spinupOutput

Spin-up output

summaryBySpecies

The total species biomass (in g/mˆ2 as in `cohortData`), average age and aNPP (in g/mˆ2 as in `cohortData`), across the landscape (used for plotting and reporting).

summaryBySpecies1

No. pixels of each leading vegetation type (used for plotting and reporting).

summaryLandscape

The averages of total biomass (in tonnes/ha , not g/mˆ2 like in `cohortData`), age and aNPP (also in tonnes/ha) across the landscape (used for plotting and reporting).

treedFirePixelTableSinceLastDisp

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

vegTypeMap

Map of leading species in each pixel, colored according to `sim$sppColorVect`. Species mixtures calculated according to `P(sim)$vegLeadingProportion` and `P(sim)$mixedType`.

2.1.5   Links to other modules

Intended to be used with other landscape modules, such as *LandMine*, *fireSense*, *Biomass_borealDataPrep*, *Biomass_regeneration* and possibly many others. You can see all *potential* module linkages within the LandR ecosystem here. Select *Biomass_core* from the drop-down menu to see linkages.

2.1.6 Getting help

- https://github.com/PredictiveEcology/Biomass_core/issues

2.2 Module manual

2.2.1 Introduction

LandR *Biomass_core* (hereafter *Biomass_core*) a forest landscape model based on the LANDIS-II Biomass Succession Extension v.3.2.1 model [LBSE; Scheller and Miranda [16]]. It is the core forest succession model of the LandR ecosystem of `SpaDES` modules. Similarly to the LBSE, *Biomass_core* simulates changes in tree cohort aboveground biomass (g/m^2.) by calculating growth, mortality and recruitment as functions of pixel and species characteristics, competition and disturbances (Fig. 2.1). Specifically, growth is driven by both invariant (`growthcurve`) and spatially varying species growth traits (maximum biomass, `maxB`, and maximum annual net primary productivity, `maxANPP`), while mortality depends only on invariant species traits (`age`, `longevity` and `mortalityshape`). Disturbances (e.g., fire) can also cause cohort mortality, but are simulated in separate modules (e.g., *Biomass_regeneration* simulates the death of all cohorts immediately after a fire). The parameters `growthcurve` and `mortalityshape` directly influence the shape of species growth curves, by determining how fast they grow and how soon age mortality starts with respect to longevity. Cohort recruitment is determined by available "space" (*i.e.*, pixel shade), invariant species traits (regeneration mode, age at maturity, shade tolerance) and spatially varying traits (species establishment probability, `SEP`). The available "growing space" is calculated as species `maxB` minus the occupied biomass (summed across other cohorts and species). If there is "space", a cohort can establish from one of three recruitment modes: serotiny, resprouting and germinating. Serotiny and resprouting occur only in response to fire and are simulated in two separate, but interchangeable modules, *Biomass_regeneration* and *Biomass_regenerationPM*. Germination occurs if seeds are made available from local sources (the pixel), or via seed dispersal. Seed dispersal can be of three modes: 'no dispersal', 'universal dispersal' (only interesting for dummy case studies) or 'ward dispersal' [16]. The 'ward dispersal' algorithm describes a flexible kernel that calculates the probability of a species colonising a neighbour pixel as a function of distance from the source and dispersal-related (and invariant) species traits, and is used by default. We refer the reader to Scheller and Miranda [16], Scheller and Domingo [15] and Scheller and Domingo [14] for further details with respect to the mechanisms implemented in the module.

2.2.2   Differences between *Biomass_core* and LBSE

2.2.2.1   Algorithm changes

Upon porting LBSE into R, we made six minor modifications to the original model's algorithms to better reflect ecological processes. This did not result in dramatic changes in simulation outputs and we note that these changes might also have been implemented in more recent versions of LBSE.

First, for each year and community (*i.e.*, 'pixel group' in *Biomass_core*, see below), LBSE calculates the competition index for a cohort sequentially (*i.e.*, one cohort at a time) after updating the growth and mortality (*i.e.*, the biomass gain and loss, respectively) of other cohorts, and with the calculation sequence following cohort age in descending order, but no explicit order of species. This sorting of growth and mortality calculations from oldest to youngest cohorts in LBSE was aimed at capturing size-asymmetric competition between cohorts, under the assumption that older cohorts have priority for growing space given their greater height (Scheller pers. comm.). We felt that sequential, within-year growth, death and recruitment may be not ecologically accurate, and that the size-asymmetric competition was being accounted for twice, as the calculation of the competition index already considers the competitive advantage of older cohorts [as shown in the User's Guide; Scheller and Miranda [16]]. Hence, in *Biomass_core* growth, mortality and the competition index are calculated at the same time across all cohorts and species.

Second, the unknown species-level sorting mechanism contained within LBSE (which changed depending on the species order in the input species list file), led to different simulation results depending on the input species list file (e.g., Table 2.2 and Fig. 2.3). The calculation of competition, growth and mortality for all cohorts at the same time also circumvented this issue.

Third, in LBSE the calculation of total pixel biomass for the purpose of calculating the initial biomass of a new cohort included the (previously calculated) biomass of other new cohorts when succession time step = 1, but not when time step was > 1. This does not reflect the documentation in the User's Guide, which stated that "Bsum [total pixel biomass] is the current total biomass for the site (not including other new cohorts)" [16, pp. 4], when the succession time step was set to 1. Additionally, together with the lack of explicit ordering, it generated different results in terms of the biomass assigned to each new cohort (e.g. Table 2.3 and Fig. 2.4). In *Biomass_core* the initial biomass of new cohorts is no longer calculated sequentially (as with competition, growth and mortality), and thus the biomass of new cohorts is never included in the calculation of total pixel biomass.

Fourth, in LBSE, serotiny and resprouting could not occur in the same pixel following a fire, with serotiny taking precedence if activated. We understand that this provides an advantage to serotinous species, which could perhaps be disadvantaged with respect to fast-growing resprouters. However, we feel that

it is ecologically more realistic that serotinous and resprouter species be able to both regenerate in a given community following a fire and allow the competition between serotinous and resprouting species to arise from species traits. **Note that this change was implemented in the *Biomass_regeneration* and *Biomass_regenerationPM* modules.**

Fifth, in *Biomass_core*, species shade tolerance values can have decimal values to allow for finer adjustments of between-species competition.

Sixth, we added a new parameter called `minCohortBiomass`, that allows the user to control cohort removal bellow a certain threshold of biomass. In some simulation set-ups, we noticed that *Biomass_core* (and LBSE) were able to generate many very small cohorts in the understory that, due to cohort competition, were not able to gain biomass and grow. However, because competition does not increase mortality, only decreases growth, these cohorts survived at very low biomass levels until they reached sufficient age to suffer age-related mortality. We felt this is unlikely to be realistic in many cases. By default, this parameter is left at 0 to follow LBSE behaviour (*i.e.*, no cohorts removal based on minimum biomass).

2.2.2.2 Other enhancements

In addition to the five minor changes in growth, mortality and regeneration, we separated the components that govern vegetation responses to disturbances – only fire at the moment – into two independent modules, used interchangeably, and implemented hashing, caching and testing to improve the model's computational efficiency and insure its performance.

2.2.2.2.1 Modularity Unlike in LBSE, post-disturbance regeneration is not part of *Biomass_core per se*, but belongs to two separate modules, used interchangeably (*Biomass_regeneration* and *Biomass_regenerationPM*). These need to be loaded and added to the "modules folder" of the project in case the user wants to simulate forest responses to disturbances (only fire disturbances at the moment). Again, this enables higher flexibility when swapping between different approaches to regeneration. For instance, default (*i.e.*, not climate sensitive) growth and mortality functions are part of the `LandR` R package, which needs to be loaded prior to running *Biomass_core*. Should the user wish to change the growth/mortality algorithms, they would need to provide compatible functions (with the same names) to the simulation via `simInit` – user-provided functions will replace those loaded with a package . Note that the `LandR` package provides other supporting functions and objects to the simulation, and still needs to be loaded prior to running *Biomass_core*.

2.2.2.2.2 Hashing Our first strategy to improve simulation efficiency in *Biomass_core* was to use a hashing mechanism [23]. Instead of assigning a key

to each pixel in a raster and tracking the simulation for each pixel in a lookup table, we indexed pixels using a *pixelGroup* key that contained unique combinations of ecolocation and community, and tracked and stored simulation data for each *pixelGroup* (Fig. 2.5). Ecolocation (called 'ecoregion' in LBSE and in model objects) is a spatial unit with similar biophysical characteristics. In our applications, we define ecolocation as the combination of land-cover types from the Land Cover Map of Canada 2005 (v1) and ecodistricts from the National Ecological Framework for Canada (). Hence, these ecolocations contain relatively fine scale land cover information plus coarse scale regional information. In turn, community is the species composition and age structure of a particular pixel. This algorithm was able to ease the computational burden by significantly reducing the size of the lookup table and speeding-up the simulation process. After recruitment and disturbance events, pixels are rehashed into new pixel groups.

2.2.2.2.3 Caching The second strategy aimed at improving model efficacy was the implementation of caching, and data-driven parametrisation and initialisation. Caching automatically archives outputs of a given function to disk (or memory) and reads them back when subsequent calls of this function are given identical inputs. All caching operations were achieved using the `reproducible` R package [9]. In the current version of *Biomass_core*, the spin-up phase was replaced by data-driven landscape initialisation and many model parameters were derived from data, using "data modules" (e.g., *Biomass_borealDataPrep*). To avoid having to repeat data downloads and treatment, statistical estimation of parameters and landscape initialisation every time the simulation is re-run under the same conditions (*i.e.*, no data or algorithm changes), many of these pre-simulation steps are automatically cached. This means that the pre-simulation phase is significantly faster upon a second call when inputs have not changed (e.g., the input data and parametrisation methods), and when inputs do change only directly affected steps are re-run (see main text for examples). When not using data modules, *Biomass_core* still relies on caching for the preparation of its theoretical inputs.

2.2.2.2.4 Testing Finally, we implemented code testing, to facilitate bug detection by comparing the outputs of functions [etc.] to expected outputs [21]. We built and integrated code tests in *Biomass_core* and across all LandR modules and the `LandR` R package and the in the form of assertions and integration tests. Assertions are run automatically during simulations (but can be turned off), while integration are be run manually. Tests were also implemented in R package dependencies of *Biomass_core*, such as the `LandR` R package and `SpaDES`, which are routinely tested using GitHub Actions continuous integration (CI) or automated checks on CRAN. For the `LandR` R package, we use GitHub Actions CI to automatically test for installation and execution errors.

Finally, because *Biomass_core* (and all other LandR modules) code is hosted

in public GitHub repositories, there is a potentially high number of users that can identify issues and contribute to improve module code.

2.2.2.3 Performance and accuracy of *Biomass_core* with respect to LBSE

In the recoding of *Biomass_core*, we ensured similar outputs of each demographic process (namely, growth, mortality and recruitment) to the outputs from its counterpart in LBSE, using integration tests. Here, we report the comparisons of the overall simulation (i.e., including all demographic processes) between LBSE and *Biomass_core* using three randomly generated initial communities (Tables 2.4-2.6). The remaining input parameters were taken from a LANDIS-II training course (Tables 2.7-2.10), and contained species attributes information of 16 common tree species in boreal forests and 2 ecolocations. We ran simulations for 1000 years, with a succession time step of 10 and three repetitions, which were enough to account for the variability produced by stochastic processes. Seed dispersal was set as "ward dispersal".

The results suggested that *Biomass_core* had a good agreement with LBSE using the three randomly generated initial communities (Fig. 2.6), with very small deviations for LBSE-generated biomasses. Notably, the mean differences between LBSE and *Biomass_core* were 0.03% (range: -0.01% ~ 0.13%), 0.03% (range: -0.01% ~ 0.11%) and 0.05% (-0.02% ~ 0.15%) for each initial community, respectively (right panels in Fig. 2.6 of this appendix).

To examine how running time changed with map size, we ran simulations using maps with increasing number of pixels from 22,201 to 638,401. All maps were initialised with a single ecolocation and 7 different communities. Simulations were run for 120 years using a succession time step of 10 and replicated three times. To eliminate the effect of hardware on running time, we used machines that were all purchased at the same time, with equal specifications and running Windows 7. Each simulation ran on 2 CPU threads with a total RAM of 4000 Mb. For both LBSE and *Biomass_core*, the simulation time increased linearly with number of pixels, but the increase rate was smaller for *Biomass_core* (Fig. 2.7a). This meant that while both models had similar simulation efficiencies in small maps ($<$ 90,000 pixels), as map size increased *Biomass_core* was ~2 times faster than LBSE (maps $>$ 100,000 pixels; Fig. 2.7a). *Biomass_core* also scaled better with map size, as LBSE speeds fluctuated between 19 to 25 seconds per 1,000 pixels across all map sizes, while *Biomass_core* decreased from 21 to 11 seconds per 1,000 pixels from smaller to larger maps (Fig. 2.7b).

2.2.3 Initialization, inputs and parameters

Unlike the initialization in LBSE, which "iterates the number of time steps equal to the maximum cohort age for each site", beginning at t – oldest cohort age

and adding cohorts at the appropriate time [16], *Biomass_core* initializes the simulation by deriving initial biomasses from available data, using data modules. If data modules are not available, *Biomass_core* initializes itself with theoretical data.

To be initialized, *Biomass_core* requires the following input objects and parameters:

2.2.3.1   Input objects

All of *Biomass_core*'s input objects have (theoretical) defaults that are produced automatically by the module (when running the `.inputObjects` function during the `simInit` call, and in the `init` event during the `spades` call – see `?SpaDES.core::events` and `SpaDES.core::simInit`). We suggest that new users run *Biomass_core* by itself supplying only a `studyArea` object. This will enable them to become familiar with all the input objects before attempting to supply their own, or combine *Biomass_core* with data modules.

List of *Biomass_core* input objects and their description.

objectName

objectClass

desc

sourceURL

biomassMap

RasterLayer

total biomass raster layer in study area (in g/m^2), filtered for pixels covered by cohortData. Only used if `P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.

cceArgs

list

a list of quoted objects used by the `growthAndMortalityDriver calculateClimateEffect` function

NA

cohortData

data.table

`data.table` with cohort-level information on age and biomass, by pixelGroup and ecolocation (i.e., `ecoregionGroup`). If supplied, it must have the following columns: `pixelGroup` (integer), `ecoregionGroup` (factor), `speciesCode` (factor), `B` (integer in g/m^2), `age` (integer in years)

NA

ecoregion

data.table

ecoregion look up table

https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/ecoregions.txt

ecoregionMap

RasterLayer

ecoregion map that has mapcodes match ecoregion table and `speciesEcoregion` table. Defaults to a dummy map matching `rasterToMatch` with two regions

NA

lastReg

numeric

an internal counter keeping track of when the last regeneration event occurred

NA

minRelativeB

data.frame

table defining the relative biomass cut points to classify stand shadeness

NA

pixelGroupMap

RasterLayer

initial community map that has mapcodes match initial community table

NA

rasterToMatch

RasterLayer

a raster of the `studyArea` in the same resolution and projection as `biomassMap`

NA

species

data.table

a table that has species traits such as longevity, shade tolerance, etc. Default is partially based on Dominic Cyr and Yan Boulanger's project

https://raw.githubusercontent.com/dcyr/LANDIS-II_IA__generalUseFiles/
master/speciesTraits.csv

speciesEcoregion

data.table

table defining the maxANPP, maxB and SEP, which can change with both
ecoregion and simulation time. Defaults to a dummy table based on dummy
data os biomass, age, ecoregion and land cover class

NA

speciesLayers

RasterStack

percent cover raster layers of tree species in Canada. Defaults to the Canadian
Forestry Service, National Forest Inventory, kNN-derived species cover maps
from 2001 using a cover threshold of 10 - see https://open.canada.ca/data/en/
dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

http://ftp.maps.canada.ca/pub/nrcan__rncan/Forests_Foret/canada-forests-
attributes_attributs-forests-canada/2001-attributes__attributs-2001/

sppColorVect

character

A named vector of colors to use for plotting.   The names must be in
`sim$sppEquiv[[sim$sppEquivCol]]`, and should also contain a color for
'Mixed'

NA

sppEquiv

data.table

table of species equivalencies. See `LandR::sppEquivalencies_CA`.

NA

studyArea

SpatialPolygonsDataFrame

Polygon to use as the study area. Must be provided by the user

NA

studyAreaReporting

SpatialPolygonsDataFrame

multipolygon (typically smaller/unbuffered than studyArea) to use for plot-
ting/reporting. Defaults to `studyArea`.

NA

sufficientLight

data.frame

table defining how the species with different shade tolerance respond to stand shade. Default is based on LANDIS-II Biomass Succession v6.2 parameters

https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt

treedFirePixelTableSinceLastDisp

data.table

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

NA

Of the inputs in Table 2.2.3.1, the following are particularly important and deserve special attention:

- **Spatial layers**

    - `ecoregionMap` – a raster layer with ecolocation IDs (note that the term "ecoregion" was inherited from LBSE and kept as is for consistency with original LBSE code). Ecolocations group pixels or similar biophysical conditions using up to two levels of grouping. In many of our applications, we use the Natural Ecoregion classification of Canada as the first grouping level and a land-cover classification as the second level. The raster layer must be defined as a categorical variable, with an associated Raster Attribute Table (RAT; see, e.g., `raster::ratify`). The RAT must contain the columns: `ID` (the value in the raster layer), `ecoregion` (the first level of ecolocation grouping) and `ecoregionGroup` (the full ecolocation "name" written as <firstlevel_secondlevel>). Note that `ecoregionGroup` usually originated from combining two raster layers and, thus, the grouping level IDs are also integers. For instance, if Natural Ecoregion 2 has land-cover types 1, 2 and 3, the RAT will contain `ID = {1,2,3}`, `ecoregion = {2}` and `ecoregionGroup = {2_1, 2_2, 2_3}`. All ecolocations are listed in the `ecoregion data.table`.

    - `rasterToMatch` – a RasterLayer, with a given resolution and projection determining the pixels (*i.e.*, non NA values) where forest dynamics will be simulated. Needs to match `studyArea`. If not supplied, *Biomass_core* attempts to produce it, using `biomassMap` as the template for spatial resolution and projection.

– `studyArea` – shapefile. A `SpatialPolygonsDataFrame` with a single
polygon determing the where the simulation will take place. This is
the only input object that **must be supplied by the user**.

- **Species traits and other parameter tables**

    – `ecoregion` – `data.table` listing all ecolocation "names" (*ecoregion-
    Group* column; see `ecoregionMap` above for details) and their state
    (active – `yes` – or inactive – `no`)

    – `minRelativeB` – `data.table` of minimum relative biomass values.
    This is a spatially variant trait used to determine the shade level in
    each pixel [see 16, pp. 14], yet in our applications we often keep values
    constant across ecolocations. The table must contain the following
    columns:

        * *ecoregionGroup* – character.   Ecolocation names.   See
          `ecoregionMap` and `ecoregion` objects above.
        * *X0-X5* – six numeric columns, one per shade class (no-shade, 0,
          to maximum shade, 5), with 0 to 1 values determining the min-
          imum threshold of biomass (relative to the species/ecolocation
          `maxB`) necessary to reach a given shade-level. This means that
          shade-levels are determined on a species by species basis [see 16,
          pp. 14]

    – `species` – `data.table` of *invariant species traits.* There are species
    traits that do no vary spatially, nor temporally (e.g., longevity). The
    table must contain the following trait values (*i.e.*, columns) in order
    to run *Biomass_core* (note that columns should follow the data type
    indicated):

        * *speciesCode* – character. Species ID.
        * *longevity* – integer. Maximum age in years [see 15, pp. 18].
        * *sexualmature* – integer. Age at sexual maturity in years [see 15,
          pp. 18].
        * *shadetolerance* – integer OR numeric. *Relative* shade tolerance
          (see Algorithm changes).
        * *seeddistance_eff* – integer. Eeffective seed distance in meters.
          [see 15, pp. 18]
        * *seeddistance_max* – integer. Maximum seed distance in meters.
          Note that is the pixel size is larger than the maximum seed dis-
          tance, the species will not be able to disperse to neighbouring
          pixels [see 15, pp. 18].
        * *mortalityshape* – integer. Shape of growth curve determining
          how quickly mortality begins [see 16, pp. 15].
        * *growthcurve* – numeric. Shape of growth curve determining
          ANPP reaches its maximum [see 16, pp. 16].

- **speciesEcoregion** – `data.table` of *spatiotemporally-varying species traits*. There are species traits vary spatially and, potentially, temporally. The table must contain the following columns in order to run *Biomass_core*:

  * *ecoregionGroup* – character. Ecolocation names. See `ecoregionMap` and `ecoregion` objects above.
  * *speciesCode* – character. Species ID.
  * *establishprob* – numeric. Species establishment probability (`SEP`) for a given species in an ecolocation and, potentially year. SEP influences the success of incoming seed germination, given pixel biophysical characteristics (note that *actual* success is determined by both SEP and light conditions in the pixel) [see 16, pp. 18].
  * *maxB* – integer. Maximum biomass for a given species in an ecolocation in units of g biomass / m^2. Note that the actual maximum biomass reached by a species in a pixel may exceed `maxB` because `maxB` is applied at the cohort level an species may have several cohorts in a given pixel [see 16, pp. 18].
  * *maxANPP* – numeric. Maximum aboveground net primary productivity in units of g biomass / m^2 / year, by default it is calculated as 1/30 of `maxB` [see 16, pp. 18]
  * *year* – integer. Used when varying `SEP`, `maxB` and `maxANPP` values in time. Otherwise, use fill all lines with `0`.

- **sufficientLight** – `data.table` defining the probability of germination for a species, given its `shadetolerance` level (see `species` above) and the shade level in the pixel (see `minRelativeB` above). Must contain columns:

  * *speciesshadetolerance* – integer. Species shade tolerance levels, from 1-5 (all levels must be present in this table).
  * *X0-X5* – six integer columns, one per shade class (no-shade, 0, to maximum shade, 5), filled with 0s OR 1s values determining the probability of germination (or resprouting) for a species given a shade-level[see 16, pp. 14]. Unlike LBSE, species `shadetolerance` values can take decimal values between 1-5, in which case the resulting probability of germination in a given pixel is interpolated between the corresponding lower and upper shade tolerance values.

- **sppEquiv** – a `data.table` of species name equivalencies between various conventions. It must contain the columns *LandR* (species IDs following in LandR format) *EN_generic_short* (short generic species names in English – or any other language; used for plotting), *Type* (type of species, *Conifer* or *Deciduous*, as in "broadleaf") and *Leading* (same as *EN_generic_short* but with "leading" appended – e.g.,

"Poplar leading") . See `?LandR::sppEquivalencies_CA` for more information.

– `sppColorVect` – character. A named vector of colours used to plot species dynamics. Should contain one colour per species in the `species` table and, potentially a colour for species mixtures (named "Mixed"). Vector names must follow `species$speciesCode`.

- **Cohort-simulation-related objects**

    – `cohortData` – a `data.table` containing initial cohort information per *pixelGroup* (see `pixelGroupMap` below). This table is updated during the simulation as cohort dynamics are simulated. Must contain the following columns

        * *pixelGroup* – integer. *pixelGroup* ID. See Hashing.
        * *ecoregionGroup* – character. Ecolocation names. See `ecoregionMap` and `ecoregion` objects above.
        * *speciesCode* – character. Species ID.
        * *age* – integer. Cohort age.
        * *B* – integer. cohort biomass in g/m^2.
        * *mortality* – integer. cohort dead biomass in the current year in g/m^2. Should be filled with 0s in initial conditions.
        * *aNPPAct* – integer. Actual aboveground net primary productivity of the current year in g/m^2. Hence *B* is the result of the previous year's *B* minus *mortality* plus *aNPPAct*. See "1.1.3 Cohort growth and ageing" section of Scheller and Miranda [16].

    – `pixelGroupMap` – a raster layer with *pixelGroup* IDs per pixel. Pixels are always grouped based on identical *ecoregionGroup*, *speciesCode*, *age* and *B* composition, even if the user supplies other initial groupings (e.g., this is possible in the *Biomass_borealDataPrep* data module).

2.2.3.2 Parameters

Table 2.2.3.2 lists all parameters used in *Biomass_core*. Note that a few of these parameters are only relevant when simulating climate effects of cohort growth and mortality, which require also loading the `LandR.CS` R package. Like with input objects, default values are supplied for all parameters and we suggest the user becomes familiarized with them before attempting any changes. We also note that the `"spin-up"` and `"biomassMap"` options for the `initialBiomassSource` are currently deactivated, since *Biomass_core* no longer generates initial cohort biomass conditions using a spin-up based on initial stand age like LANDIS-II (`"spin-up"`), nor does it attempt to fill initial cohort biomasses using `biomassMap` (`"biomassMap"`). A list of useful parameters and their description is shown in Table 2.2.3.2.

List of *Biomass_core* parameters and their description.

paramName

paramClass

default

min

max

paramDesc

calcSummaryBGM

character

end

NA

NA

A character vector describing when to calculate the summary of biomass, growth and mortality Currently any combination of 5 options is possible: 'start'- as before vegetation succession events, i.e. before dispersal, 'postDisp' - after dispersal, 'postRegen' - after post-disturbance regeneration (currently the same as 'start'), 'postGM' - after growth and mortality, 'postAging' - after aging, 'end' - at the end of vegetation succesion events, before plotting and saving. The 'end' option is always active, being also the default option.

calibrate

logical

FALSE

NA

NA

Do calibration? Defaults to `FALSE`

cohortDefinitionCols

character

pixelGro. . . .

NA

NA

`cohortData` columns that determine what constitutes a cohort This parameter should only be modified if additional modules are adding columns to cohortData

cutpoint

numeric

1e+10

NA

NA

A numeric scalar indicating how large each chunk of an internal data.table is, when processing by chunks

gmcsGrowthLimits

numeric

66.66666. . . .

NA

NA

if using `LandR.CS` for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model

gmcsMortLimits

numeric

66.66666. . . .

NA

NA

if using `LandR.CS` for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model

gmcsMinAge

numeric

21

0

NA

if using `LandR.CS` for climate-sensitive growth and mortality, the minimum age for which to predict climate-sensitive growth and mortality. Young stands ($<$ 30) are poorly represented by the PSP data used to parameterize the model.

growthAndMortalityDrivers

character

LandR

NA

NA

package name where the following functions can be found: `calculateClimateEffect`, `assignClimateEffect` (see `LandR.CS` for climate sensitivity equivalent functions, or leave default if this is not desired)

growthInitialTime

numeric

start(sim)

NA

NA

Initial time for the growth event to occur

initialBiomassSource

character

cohortData

NA

NA

Currently, there are three options: 'spinUp', 'cohortData', 'biomassMap'. If 'spinUp', it will derive biomass by running spinup derived from Landis-II. If 'cohortData', it will be taken from the `cohortData` object, i.e., it is already correct, by cohort. If 'biomassMap', it will be taken from `sim$biomassMap`, divided across species using `sim$speciesLayers` percent cover values 'spinUp' uses `sim$standAgeMap` as the driver, so biomass is an output . That means it will be unlikely to match any input information about biomass, unless this is set to 'biomassMap', and a `sim$biomassMap` is supplied. Only the 'cohortData' option is currently active.

keepClimateCols

logical

FALSE

NA

NA

include growth and mortality predictions in `cohortData`?

minCohortBiomass

numeric

0

NA

NA

cohorts with biomass below this threshold (g/m^2) are removed. Not a LANDIS-II BSE parameter.

mixedType

numeric

2

NA

NA

How to define mixed stands: 1 for any species admixture; 2 for deciduous > conifer. See ?LandR::vegTypeMapGenerator.

plotOverstory

logical

FALSE

NA

NA

swap max age plot with overstory biomass

seedingAlgorithm

character

wardDisp. . . .

NA

NA

choose which seeding algorithm will be used among 'noDispersal', 'universalDispersal', and 'wardDispersal' (default). See Scheller & Miranda (2015) - Biomass Succession extension, v3.2.1 User Guide

spinupMortalityfraction

numeric

0.001

NA

NA

defines the mortality loss fraction in spin up-stage simulation. Only used if
`P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.

sppEquivCol

character

Boreal

NA

NA

The column in `sim$sppEquiv` data.table to use as a naming convention

successionTimestep

numeric

10

NA

NA

defines the simulation time step, default is 10 years. Note that growth and
mortality always happen on a yearly basis.

vegLeadingProportion

numeric

0.8

0

1

a number that defines whether a species is leading for a given pixel

.maxMemory

numeric

5

NA

NA

maximum amount of memory (in GB) to use for dispersal calculations.

.plotInitialTime

numeric

start(sim)

NA

NA

Vector of length = 1, describing the simulation time at which the first plot event should occur. To plotting off completely use `P(sim)$.plots`.

.plotInterval

numeric

NA

NA

NA

defines the plotting time step. If `NA`, the default, .plotInterval is set to successionTimestep.

.plots

character

object

NA

NA

Passed to `types` in `Plots` (see `?Plots`). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at `end(sim)`. If `NA` plotting is off completely (this includes plot saving).

.plotMaps

logical

TRUE

NA

NA

Controls whether maps should be plotted or not. Set to `FALSE` if `P(sim)$.plots == NA`

.saveInitialTime

numeric

NA

NA

NA

Vector of length = 1, describing the simulation time at which the first save event should occur. Set to `NA` if no saving is desired. If not `NA`, then saving will occur at `P(sim)$.saveInitialTime` with a frequency equal to `P(sim)$.saveInterval`

.saveInterval

numeric

NA

NA

NA

defines the saving time step. If `NA`, the default, .saveInterval is set to `P(sim)$successionTimestep`.

.studyAreaName

character

NA

NA

NA

Human-readable name for the study area used. If `NA`, a hash of `studyArea` will be used.

.useCache

character

.inputOb. . . .

NA

NA

Internal. Can be names of events or the whole module name; these will be cached by `SpaDES`

.useParallel

ANY

2

NA

NA

Used only in seed dispersal. If numeric, it will be passed to `data.table::setDTthreads` and should be $<= 2$; If `TRUE`, it will be passed to `parallel::makeCluster`; and if a cluster object, it will be passed to `parallel::parClusterApplyB`.

Required inputs	Description
Plotting & saving	
`.plots`	activates/deactivates plotting and defines type fo plotting (see **?Plots**)
`.plotInitialTime`	defines when plotting starts
`.plotInterval`	defines plotting frequency
`.plotMaps`	activates/deactivates map plotting
`.saveInitialTime`	defines when saving starts
`.saveInterval`	defines saving frequency
Simulation	
`seedingAlgorithm`	dispersal type (see above)
`successionTimestep`	defines frequency of dispersal/local recruitment event (growth and mortality are always yearly)
Other	
`mixedType`	how mixed forest stands are defined
`vegLeadingProportion`	relative biomass threshold to consider a species "leading" (*i.e.*, dominant)

: Useful *Biomass_core* parameters.

2.2.4   Simulation flow

2.2.4.1   No disturbances

Biomass_core itself does not simulate disturbances, or their effect on vegetation (*i.e.*, post-disturbance mortality and regeneration). The general flow of *Biomass_core* processes is:

1. Preparation of necessary objects for the simulation – either by accessory data prep. modules, or *Biomass_core* itself (using LANDIS-II test parameters and dummy data for stand age, biomass and land cover and ecological zoning)
2. Seed dispersal – see Scheller and Domingo [14] for details

- Seed dispersal can be a slow process and has been adapted to occur every 10 years. The user can set it to occur more often, but this should not make much of a difference to model outputs, because age classes are meant to be collapsed to tens.

3. Growth and mortality – based on Scheller and Mladenoff [18]

- unlike dispersal, growth and mortality should occur every year

4. Ageing – based on Scheller and Miranda [16]

- follows the same frequency as dispersal, collapsing ages to classes with resolution = to this frequency

5. Preparation of visual/saved outputs ... (repeat 2-4) ...

2.2.4.2   With disturbances

Note that should a post-disturbance regeneration module be used (e.g., *Biomass_regeneration*), regeneration will occur after the disturbance, but *before* dispersal and background vegetation growth and mortality. Hence, the disturbance should take place either at the very beginning or at the very end of each simulation time step. The general flow of *Biomass_core* processes when disturbances are included (by linking other modules) is:

1. Preparation of necessary objects for the simulation – either by accessory prep. data modules, or *Biomass_core* itself (using LANDIS-II test parameters and dummy data.)
2. Disturbances – simulated by a disturbance module
3. Post-disturbance regeneration – simulated by a regeneration module (*Biomass_regeneration* is an optional download)
4. Seed dispersal – see Scheller and Domingo [14] for details
5. Growth, ageing and mortality – based on Scheller and Miranda [16]
6. Preparation of visual/saved outputs ... (repeat 2-6) ...

2.3   Usage example

2.3.1   Load `SpaDES`

```r
if (!require(Require)) {
  install.packages("Require")
  library(Require)
}

Require(c("PredictiveEcology/SpaDES.install",
          "SpaDES", "PredictiveEcology/SpaDES.core@development",
          "PredictiveEcology/LandR"),
        install_githubArgs = list(dependencies = TRUE))# In general, a module code will be contro
```

```r
tempDir <- tempdir()
spadesModulesDirectory <- file.path(tempDir, "modules")

paths <- list(inputPath = normPath(file.path(tempDir, "inputs")),
              cachePath = normPath(file.path(tempDir, "cache")),
              modulePath = normPath(spadesModulesDirectory),
              outputPath = normPath(file.path(tempDir, "outputs")))

getModule("PredictiveEcology/Biomass_core",
                      modulePath = spadesModulesDirectory, overwrite = TRUE)

## make sure all necessary packages are installed:
makeSureAllPackagesInstalled(spadesModulesDirectory)
```

2.3.2   Get the module

See SpaDES-modules repository to see how to download this and other `SpaDES`
modules. Alternatively, it can be forked or cloned from its GitHub repository
directly.

2.3.3   Setup simulation

```r
times <- list(start = 0, end = 30)

studyArea <- Cache(randomStudyArea, size = 1e7) # cache this so it creates a random on

# Pick the species you want to work with – using the naming convention in "Boreal" col
speciesNameConvention <- "Boreal"
speciesToUse <- c("Pice_Gla", "Popu_Tre", "Pinu_Con")

sppEquiv <- LandR::sppEquivalencies_CA[get(speciesNameConvention) %in% speciesToUse]
# Assign a colour convention for graphics for each species
sppColorVect <- LandR::sppColors(sppEquiv, speciesNameConvention,
                                 newVals = "Mixed", palette = "Set1")

## Usage example
modules <- as.list(moduleName)
objects <- list(studyArea = studyArea, sppEquiv = sppEquiv, sppColorVect = sppColorVect

successionTimestep <- 10L

## keep default values for most parameters
```

```r
## (omitted from this list)
parameters <- list(
  Biomass_core = list(
    "sppEquivCol" = speciesNameConvention
    , "successionTimestep" = successionTimestep
    , ".plots" = c("screen", "object")
    , ".plotInitialTime" = times$start
    , ".plots" = c("screen", "png")
    , ".saveInitialTime" = times$start
    , ".useCache" = "init"
    , ".useParallel" = FALSE
  )
)

outputs <- data.frame(expand.grid(objectName = "cohortData",
                                  saveTime = unique(seq(times$start, times$end, by = 1)),
                                  eventPriority = 1,
                                  stringsAsFactors = FALSE))

graphics.off()
```

2.3.4 Run simulation

```r
mySim <- simInitAndSpades(times = times,
                          params = parameters,
                          modules = modules,
                          objects = objects,
                          paths = paths,
                          outputs = outputs,
                          debug = TRUE)
```

2.4 Appendix

2.4.1 Tables
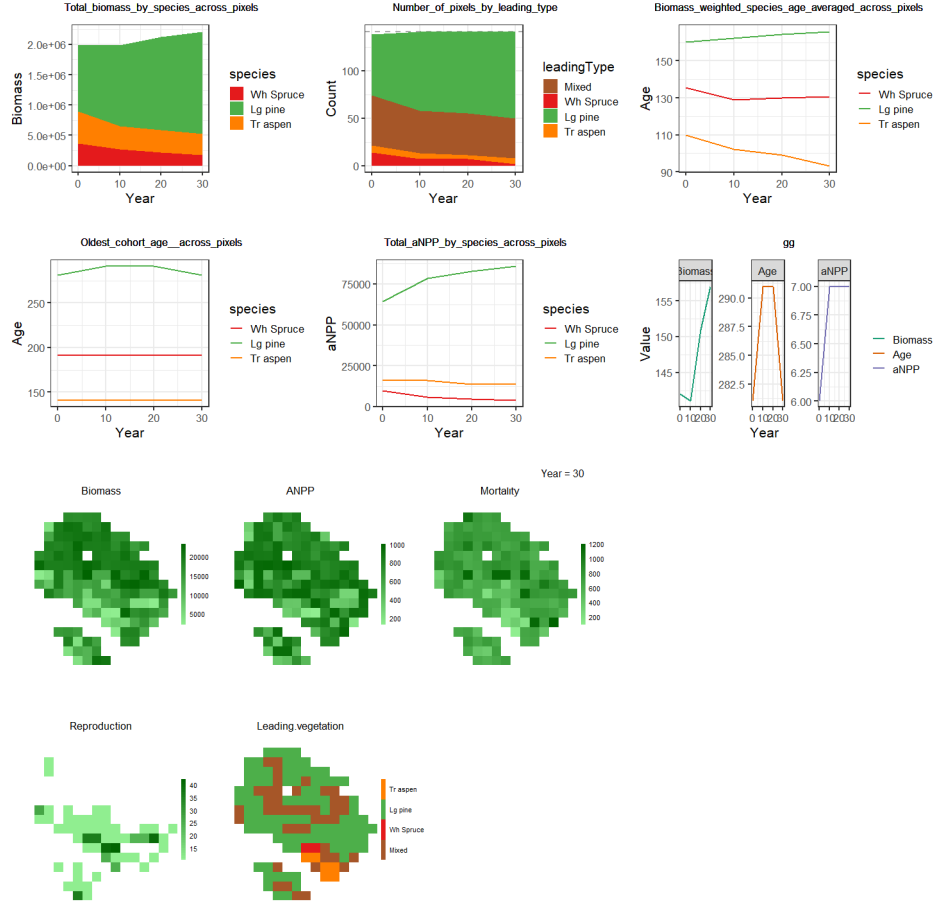
Figure 2.2: *Biomass_core* automatically generates simulation visuals of species dynamics across the landscape in terms of total biomass, number of presences and age and productivity (above), as well as yearly plots of total biomass, productivity, mortality, reproduction and leading species in each pixel (below).

Table 2.2: Input order and processing order (as determined by LBSE) for the same community used to assess the impact of sequential calculation of the competition index, combined with a lack of explicit species ordering. The input order was the order of species in the initial communities table input file. The processing order was the order used in the simulation, which was obtained from `Landis-log.txt` when `CalibrateMode` was set to 'yes'. Species starting ages are also shown.

Input order 1				Input order 2			
Community	Input order	Age	Processing order	Community	Input order	Age	Processing order
1	abiebals	20	poputrem	1	pinustro	20	thujocci
1	acerrubr	20	querelli	1	poputrem	20	tiliamer
1	acersacc	20	pinuresi	1	acerrubr	20	querelli
1	betualle	20	pinustro	1	pinubank	20	querrubr
1	betupapy	20	tiliamer	1	betualle	20	betupapy
1	fraxamer	20	tsugcana	1	piceglau	20	fraxamer
1	piceglau	20	querrubr	1	pinuresi	20	tsugcana
1	pinubank	20	thujocci	1	acersacc	20	abiebals
1	pinuresi	20	acersacc	1	querelli	20	acerrubr
1	pinustro	20	betualle	1	querrubr	20	pinubank
1	poputrem	20	abiebals	1	thujocci	20	pinustro
1	querelli	20	acerrubr	1	tiliamer	20	poputrem
1	querrubr	20	piceglau	1	tsugcana	20	pinuresi
1	thujocci	20	pinubank	1	abiebals	20	acersacc
1	tiliamer	20	betupapy	1	betupapy	20	betualle
1	tsugcana	20	fraxamer	1	fraxamer	20	piceglau

Table 2.3: Input order and processing order (as determined by LBSE) for the same community used to assess the impact of setting the succession time step to 1, combined with a lack of explicit species ordering. The input order was the order of species in the initial communities table input file. The processing order was the order used in the simulation, which was obtained from `Landis-log.txt` when `CalibrateMode` was set to 'yes'. Species starting ages are also shown.

Input order 1				Input order 2			
Community	Input order	Age	Processing order	Community	Input order	Age	Processing order
1	abiebals	1	poputrem	1	pinustro	1	thujocci

Input order 1			Input order 2		
1	acerrubr 1	querelli 1	1	poputrem 1	tiliamer
1	acersacc 1	pinuresi 1	1	acerrubr 1	querelli
1	betualle 1	pinustro 1	1	pinubank 1	querrubr
1	betupapy 1	tiliamer 1	1	betualle 1	betupapy
1	fraxamer 1	tsugcana 1	1	piceglau 1	fraxamer
1	piceglau 1	querrubr 1	1	pinuresi 1	tsugcana
1	pinubank 1	thujocci 1	1	acersacc 1	abiebals
1	pinuresi 1	acersacc 1	1	querelli 1	acerrubr
1	pinustro 1	betualle 1	1	querrubr 1	pinubank
1	poputrem 1	abiebals 1	1	thujocci 1	pinustro
1	querelli 1	acerrubr 1	1	tiliamer 1	poputrem
1	querrubr 1	piceglau 1	1	tsugcana 1	pinuresi
1	thujocci 1	pinubank 1	1	abiebals 1	acersacc
1	tiliamer 1	betupapy 1	1	betupapy 1	betualle
1	tsugcana 1	fraxamer 1	1	fraxamer 1	piceglau

Table 2.4: Randomly generated community combination no. 1 used in the recruitment comparison runs.

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
0	betupapy	1	37	45	46	85	NA	NA
0	piceglau	27	73	153	256	270	NA	NA
0	pinustro	157	159	181	220	223	303	307
0	querrubr	80	102	127	152	206	227	NA
1	acerrubr	3	91	126	145	NA	NA	NA
1	acersacc	138	144	276	NA	NA	NA	NA
1	betualle	24	106	136	149	279	NA	NA
1	piceglau	27	67	70	153	NA	NA	NA
1	pinubank	3	10	24	31	71	NA	NA
1	querelli	92	224	234	NA	NA	NA	NA
1	thujocci	73	146	262	NA	NA	NA	NA
2	fraxamer	108	118	137	147	204	NA	NA
2	piceglau	40	128	131	159	174	NA	NA
2	pinustro	78	156	237	245	270	NA	NA
2	querelli	67	97	186	292	NA	NA	NA
2	tiliamer	70	103	121	152	178	180	245
3	acerrubr	5	83	125	126	127	NA	NA
3	pinuresi	1	25	42	49	76	79	103
3	poputrem	4	9	62	NA	NA	NA	NA
3	querelli	101	104	167	226	NA	NA	NA
3	tsugcana	37	135	197	404	405	NA	NA

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
4	acerrubr	15	29	63	70	105	133	NA
4	piceglau	67	132	189	NA	NA	NA	NA
4	tsugcana	21	26	110	146	341	462	463
5	acerrubr	128	137	145	147	NA	NA	NA
5	acersacc	241	245	261	277	NA	NA	NA
5	querrubr	23	72	120	142	188	NA	NA
5	tiliamer	4	68	98	118	139	197	NA
6	betualle	5	23	31	249	NA	NA	NA
6	pinubank	67	70	89	NA	NA	NA	NA
6	querelli	194	217	257	NA	NA	NA	NA

Table 2.5: Randomly generated community combination no. 2 used in the recruitment comparison runs.

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
0	acerrubr	22	26	30	40	47	145	146
0	betualle	23	41	43	120	209	227	270
0	fraxamer	25	90	119	173	185	282	NA
0	pinuresi	48	53	70	121	157	NA	NA
0	pinustro	5	82	126	298	352	NA	NA
0	querrubr	2	30	34	74	77	162	245
1	acerrubr	2	39	43	84	116	127	143
1	pinubank	34	57	75	NA	NA	NA	NA
1	querelli	108	202	218	243	NA	NA	NA
1	querrubr	5	117	131	186	189	246	NA
1	tiliamer	10	19	46	80	133	148	231
1	tsugcana	31	48	190	246	330	NA	NA
2	pinubank	11	37	38	47	67	93	NA
2	querrubr	11	48	57	177	180	228	236
2	tiliamer	28	42	78	79	223	250	NA
2	tsugcana	140	202	372	381	451	NA	NA
3	acersacc	48	107	262	265	NA	NA	NA
3	betupapy	4	12	45	65	83	96	NA
3	poputrem	13	20	37	75	90	NA	NA
3	querelli	72	90	104	115	116	265	278
3	tiliamer	20	21	56	98	237	NA	NA
3	tsugcana	86	224	425	429	NA	NA	NA
4	fraxamer	77	133	181	NA	NA	NA	NA
4	pinustro	13	37	67	220	287	293	375
4	querrubr	27	48	89	97	NA	NA	NA
4	thujocci	91	244	305	390	NA	NA	NA
5	abiebals	86	95	119	121	127	158	NA

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
5	betualle	83	113	136	161	216	231	NA
5	betupapy	10	38	64	NA	NA	NA	NA
5	piceglau	16	63	70	102	NA	NA	NA
6	acerrubr	8	34	112	NA	NA	NA	NA
6	betupapy	1	31	57	61	74	80	91
6	fraxamer	63	100	108	140	196	294	NA
6	pinubank	15	19	44	47	51	80	NA
6	thujocci	78	146	163	213	214	228	NA
6	tsugcana	47	108	387	389	449	NA	NA

Table 2.6: Randomly generated community combination no. 3 used in the recruitment comparison runs.

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
0	pinubank	7	26	32	37	48	85	90
0	pinuresi	11	103	109	179	188	197	NA
0	querrubr	89	139	180	206	NA	NA	NA
1	betupapy	36	39	45	49	66	68	NA
1	piceglau	13	165	254	NA	NA	NA	NA
1	pinubank	3	19	54	64	76	NA	NA
1	poputrem	22	59	93	NA	NA	NA	NA
1	thujocci	68	98	274	275	363	378	NA
1	tiliamer	13	20	105	124	248	NA	NA
1	tsugcana	36	90	142	NA	NA	NA	NA
2	fraxamer	11	241	279	NA	NA	NA	NA
2	piceglau	16	42	129	177	200	244	NA
2	pinustro	200	342	384	NA	NA	NA	NA
3	abiebals	31	57	61	92	108	162	183
3	piceglau	126	255	261	267	NA	NA	NA
3	poputrem	28	41	57	NA	NA	NA	NA
3	querrubr	83	91	144	173	184	238	NA
3	thujocci	6	66	68	204	NA	NA	NA
4	fraxamer	12	110	266	270	NA	NA	NA
4	pinustro	174	270	359	379	NA	NA	NA
4	poputrem	4	7	18	24	63	76	NA
4	tiliamer	126	136	197	NA	NA	NA	NA
4	tsugcana	49	91	128	194	411	487	NA
5	abiebals	35	53	108	114	147	174	195
5	acerrubr	1	2	101	145	NA	NA	NA
5	pinubank	14	15	38	40	59	69	83
6	acerrubr	4	46	117	NA	NA	NA	NA
6	betualle	36	41	116	213	253	NA	NA

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
6	betupapy	4	6	76	NA	NA	NA	NA
6	pinuresi	43	68	85	171	NA	NA	NA
6	querrubr	84	86	113	185	193	223	228
6	tiliamer	13	106	181	199	246	NA	NA

Table 2.7: Invariant species traits table used in comparison runs.

Species	Longevity	Sexualmaturity	Shadetolerance	Seeddistance	Seeffdistance	Mortalityslope	Growthcurve
abiebals	200	25	5	30	160	10	0.25
acerrubr	150	10	4	100	200	10	0.25
acersacc	300	40	5	100	200	10	0.25
betualle	300	40	4	100	400	10	0.25
betupapy	100	30	2	200	5000	10	0.25
fraxamer	300	30	4	70	140	10	0.25
piceglau	300	25	3	30	200	10	0.25
pinubank	100	15	1	20	100	10	0.25
pinuresi	200	35	2	20	275	10	0.25
pinustro	400	40	3	60	210	10	0.25
poputrem	100	20	1	1000	5000	10	0.25
querelli	300	35	2	30	3000	10	0.25
querrubr	250	25	3	30	3000	10	0.25
thujocci	400	30	2	45	60	10	0.25
tiliamer	250	30	4	30	120	10	0.25
tsugcana	500	30	5	30	100	10	0.25

Table 2.8: Minimum relative biomass table used in comparison runs. X0-5 represent site shade classes from no-shade (0) to maximum shade (5). All ecolocations shared the same values.

Ecolocation	X0	X1	X2	X3	X4	X5
All	0	0.15	0.25	0.5	0.8	0.95

Table 2.9: Probability of germination for species shade tolerance and shade level combinations (called *sufficient light* table in LBSE and `sufficientLight` input `data.table` in LandR *Biomass_core*) used in comparison runs.

Shadetolerance	0	1	2	3	4	5
1	1	0	0	0	0	0
2	1	1	0	0	0	0
3	1	1	1	0	0	0
4	1	1	1	1	0	0
5	0	0	1	1	1	1

Table 2.10: Species ecolocation table used in comparison runs. `SEP` stands for species establishment probability, `maxB` for maximum biomass and `maxANPP` for maximum aboveground net primary productivity. Values were held constant throughout the simulation.

Ecolocation	Species	SEP	maxANPP	maxB
1	abiebals	0.9	886	26580
1	acerrubr	1	1175	35250
1	acersacc	0.82	1106	33180
1	betualle	0.64	1202	36060
1	betupapy	1	1202	36060
1	fraxamer	0.18	1202	36060
1	piceglau	0.58	969	29070
1	pinubank	1	1130	33900
1	pinuresi	0.56	1017	30510
1	pinustro	0.72	1090	38150
1	poputrem	1	1078	32340
1	querelli	0.96	1096	32880
1	querrubr	0.66	1017	30510
1	thujocci	0.76	1090	32700
1	tiliamer	0.54	1078	32340
1	tsugcana	0.22	1096	32880

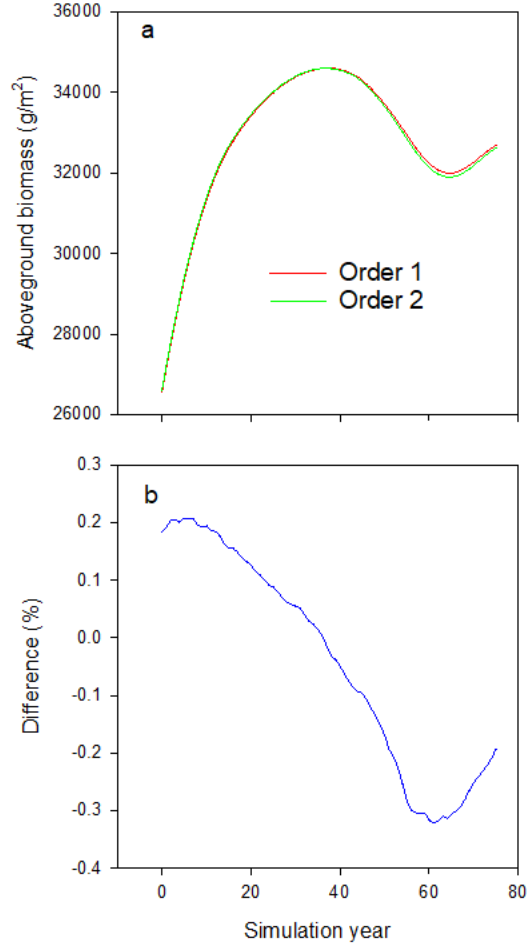## 2.4.2   Figures

2.5   References

Figure 2.3: Differences in total landscape aboveground biomass when using two different input species orders for the same community. These simulations demonstrate how the sequential calculation of the competition index, combined with a lack of explicit species ordering affect the overall landscape aboveground biomass in time when using different input species orders (see Table reftab:tableLBSEtest1). In order to prevent differences introduced by cohort recruitment, species' ages at sexual maturity were changed to the species' longevity values, and the simulation ran for 75 years to prevent any cohorts from reaching sexual maturity. The bottom panel shows the difference between the two simulations in percentage, calculated as $\frac{Biomass_{order2} - Biomass_{order1}}{Biomass_{order2}} * 100$
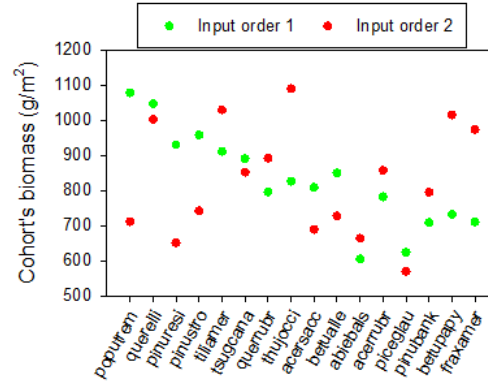
Figure 2.4: Differences in the biomasses assigned to new cohorts, summed for each species across pixels, when using two different input species orders for the same community and when the succession time step is 1. These simulations demonstrate how the different summation of total cohort biomass for a succession time step of 1 and the lack of explicit species ordering affect simulation results when changing the species order in the input file (see Table reftab:tableLBSEtest2). Here, initial cohort ages were also set to 1. We show the initial total biomass attributed to each species at the end of year 1.
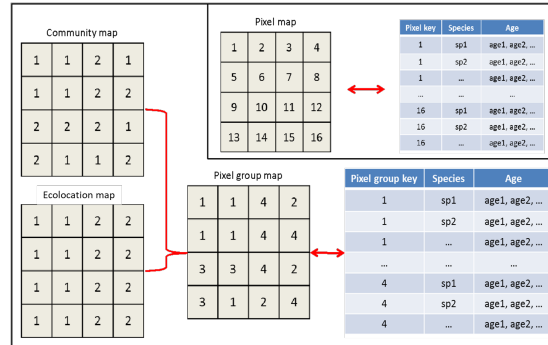

Figure 2.5: Hashing design for *Biomass_core*. In the re-coded *Biomass_core*, the pixel group map was hashed based on the unique combination of species composition (i.e., community map) and ecolocation map, and associated with a lookup table. The subfigure in the right upper corner was the original design that linked the map to the lookup table by pixel key.
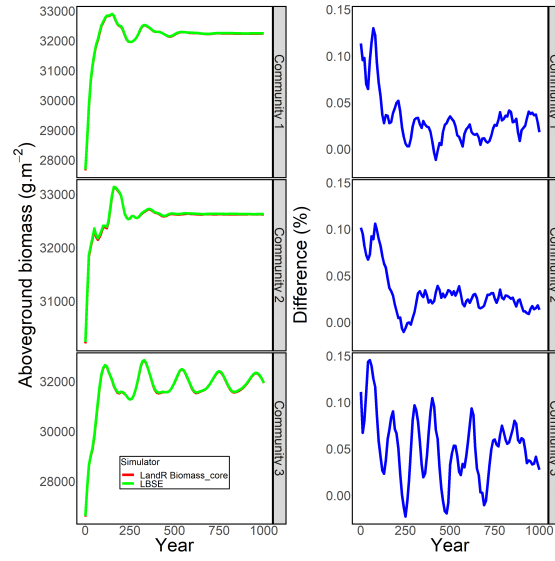
Figure 2.6: Visual comparison of simulation outputs for three randomly generated initial communities (left panels) and difference between those outputs (right panels). The % difference between LBSE and *Biomass_core* were calculated as $\frac{Biomass_{LBSE} - Biomass_{Biomass_{core}}}{Biomass_{LBSE}} * 100$
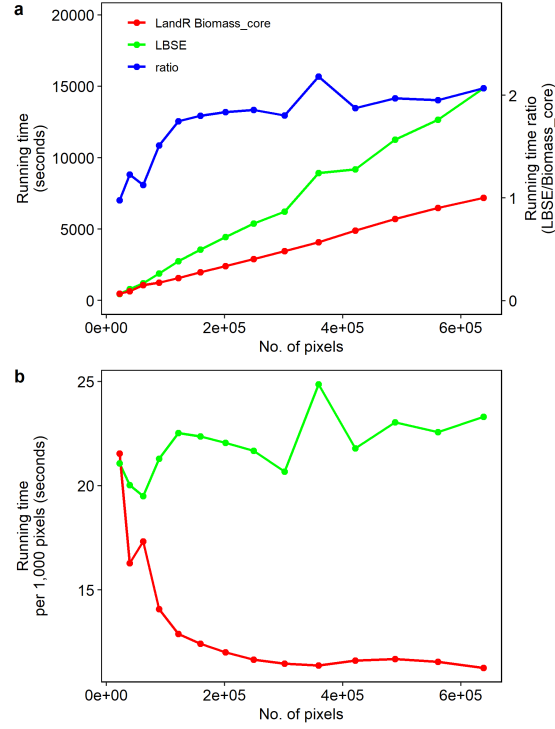
Figure 2.7: Simulation efficiencies of LBSE and *Biomass_core* with increasing map size, in terms of a) mean running time across repetitions (left y-axis) and the ratio LBSE to *Biomass_core* running times (right y-axis and blue line), and b) running time scalability as the mean running time per 1000 pixels.

Chapter 3

LandR Data Modules

The LandR ecosystem of SpaDES modules has a variety of 'data modules' that are used to obtain and pre-process input data, as well as estimate input parameters required by the core forest landscape simulation module *Biomass_core*. These modules are presented in the next chapters.

Chapter 4

LandR *Biomass_speciesData* Module

This documentation is work in progress. Potential discrepancies and omissions may exist for the time being. If you find any, do contact us using the link above^^

4.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca [aut, cre], Alex M. Chubaty achubaty@for-cast.ca [aut], Ceres Barros cbarros@mail.ubc.ca [aut]

4.1 Module Overview

4.1.1 Module summary

This module downloads and pre-process species % cover data layers to be passed to other LandR data modules (e.g., *Biomass_borealDataPrep*) or to the LandR forest simulation module *Biomass_core*.

57

4.1.2   Module inputs and parameters at a glance

Below are the full list of input objects (Table 4.1.2) and parameters (Table 4.1.2) that *Biomass_speciesData* expects. Of these, the only input that **must** be provided (i.e., *Biomass_speciesData* does not have a default for) is `studyAreaLarge`.

Raw data layers downloaded by the module are saved in `dataPath(sim)`, which can be controlled via `options(reproducible.destinationPath = ...)`.

List of *Biomass_speciesData* input objects and their description.

objectName

desc

rasterToMatchLarge

a raster of `studyAreaLarge` in the same resolution and projection the simulation's. Defaults to the using the Canadian Forestry Service, National Forest Inventory, kNN-derived stand biomass map.

sppColorVect

A named vector of colors to use for plotting. The names must be in sim*sppEquiv*[[*sim*sppEquivCol]], and should also contain a color for 'Mixed'

sppEquiv

table of species equivalencies. See `LandR::sppEquivalencies_CA`.

studyAreaLarge

Polygon to use as the parametrisation study area. Must be provided by the user. Note that `studyAreaLarge` is only used for parameter estimation, and can be larger than the actual study area used for LandR simulations (e.g, larger than `studyArea` in LandR Biomass_core).

studyAreaReporting

multipolygon (typically smaller/unbuffered than `studyAreaLarge` and `studyArea` in LandR Biomass_core) to use for plotting/reporting. If not provided, will default to `studyAreaLarge`.

List of *Biomass_speciesData* parameters and their description.

paramName

paramDesc

coverThresh

The minimum % cover a species needs to have (per pixel) in the study area to be considered present

dataYear

Passed to `paste0('prepSpeciesLayers_', types)` function to fetch data from that year (if applicable). Defaults to 2001 as the default kNN year.

sppEquivCol

The column in `sim$sppEquiv` data.table to group species by and use as a naming convention. If different species in, e.g., the kNN data have the same name in the chosen column, their data are merged into one species by summing their % cover in each raster cell.

types

The possible data sources. These must correspond to a function named paste0('prepSpeciesLayers_', types). Defaults to 'KNN' to get the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from year 'dataYear', using the `LandR::prepSpeciesLayers_KNN` function (see https://open.canada.ca/ data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for details on these data). Other currently available options are 'ONFRI', 'CASFRI', 'Pickell' and 'ForestInventory', which attempt to get proprietary data - the user must be granted access first. A custom function can be used to retrieve any data, just as long as it is accessible by the module (e.g., in the global environment) and is named as paste0('prepSpeciesLayers_', types).

vegLeadingProportion

a number that defines whether a species is leading for a given pixel. Only used for plotting.

.plotInitialTime

This describes the simulation time at which the first plot event should occur

.plotInterval

This describes the simulation time interval between plot events

.saveInitialTime

This describes the simulation time at which the first save event should occur

.saveInterval

This describes the simulation time interval between save events

.studyAreaName

Human-readable name for the study area used. If NA, a hash of `studyAreaLarge` will be used.

.useCache

Controls cache; caches the init event by default

.useParallel

Used in reading csv file with fread. Will be passed to data.table::setDTthreads.

4.1.3   Events

Biomass_speciesData only runs two events:

- Module "initiation" (`init` event), during which all species % cover layers are downloaded and processed.
- Plotting of the processed species cover layers (`initPlot` event).

4.1.4   Module outputs

The module produces the following outputs (Table 4.1.4):

List of *Biomass_speciesData* output objects and their description.

objectName

desc

speciesLayers

biomass percentage raster layers by species in Canada species map

treed

Table with one logical column for each species, indicating whether there were non-zero cover values in each pixel.

numTreed

a named vector with number of pixels with non-zero cover values for each species

nonZeroCover

A single value indicating how many pixels have non-zero cover

and automatically saves the processed species cover layers in the output path defined in `getPaths(sim)$outputPath`.

4.1.5   Links to other modules

Intended to be used with other LandR data modules (e.g., *Biomass_borealDataPrep*) that require species cover data and the LandR forest simulation *Biomass_core* module. You can see all *potential* module linkages within the LandR ecosystem here. Select *Biomass_speciesData* from the drop-down menu to see linkages.

4.1.6   Getting help

- https://github.com/PredictiveEcology/Biomass_speciesData/issues

4.2 Module manual

4.2.1 Detailed description

This module accesses and processes species percent cover (% cover) data for the parametrisation and initialization of LandR *Biomass_core*. This module ensures 1) all data use the same geospatial geometries and 2) that these are correctly re-projected to studyAreaLarge, and 3) attempts to sequentially fill-in and replace the lowest quality data with higher quality data when several data sources are used. It's primary output is a `RasterStack` of species % cover, with each layer corresponding to a species.

Currently, the module can access the Canadian Forest Inventory forest attributes kNN dataset [the default; Beaudoin et al. [3]], the Common Attribute Schema for Forest Resource Inventories [CASFRI; Cosco [6]] dataset, the Ontario Forest Resource Inventory (ONFRI), a dataset specific to Alberta compiled by Paul Pickell, and other Alberta forest inventory datasets. However, **only the NFI kNN data are freely available** – access to the other datasets must be granted by module developers and data owners, and a Google account is required. Nevertheless, the module is flexible enough that any user can use it to process additional datasets, provided that an adequate R function is passed to the module (see `types` parameter details in Parameters)

When multiple data sources are used, the module will use replace lower quality data with higher quality data following the order specified by the parameter `types` (see Parameters).

The module can also exclude species % cover layers if they don't have a minimum % cover value in at least one pixel. This means that the user should still inspect in how many pixels the species is deemed present, as it is possible that some data have only a few pixels with high % cover for a given species. In this case, the user may choose to exclude these species *a posteriori*. The summary plot automatically shown by *Biomass_speciesData* can help diagnose whether certain species are present in very few pixels (see Fig. 4.1).

4.2.2 Initialization, inputs and parameters

Biomass_speciesData initializes itself and prepares all inputs provided that it has internet access to download the raw data layers (or that these layers have been previously downloaded and stored in the folder specified by `options("reproducible.destinationPath")`).

The module defaults to processing cover data fo all species listed in the `Boreal` column of the default `sppEquiv` input `data.table` object, for which there are available % cover layers in the kNN dataset (Table 4.2.2; see `?LandR::sppEquivalencies_CA` for more information):

List of species cover data downloaded by default by *Biomass_speciesData.*

Species

Generic name

Abies balsamea

Balsam Fir

Abies lasiocarpa

Fir

Acer negundo

Boxelder maple

Acer pensylvanicum

Striped maple

Acer saccharum

Sugar maple

Acer spicatum

Mountain maple

Acer spp.

Maple

Alnus spp

Alder

Betula alleghaniensis

Swamp birch

Betula papyrifera

Paper birch

Betula populifolia

Gray birch

Betula spp.

Birch

Fagus grandifolia

American beech

Fraxinus americana

American ash

Fraxinus nigra

Black ash

Larix laricina

Tamarack

Larix lyallii

Alpine larch

Larix occidentalis

Western larch

Larix spp.

Larch

Picea engelmannii

Engelmann's spruce

Picea glauca

White.Spruce

Picea mariana

Black.Spruce

Picea spp.

Spruce

Pinus albicaulis

Whitebark pine

Pinus banksiana

Jack pine

Pinus contorta

Lodgepole pine

Pinus monticola

Western white pine

Pinus resinosa

Red pine

Pinus spp.

Pine

Populus balsamifera v. balsamifera

Balsam poplar

Populus trichocarpa

Black cottonwood

Populus grandidentata

White poplar

Populus spp.

Poplar

Populus tremuloides

Trembling poplar

Tsuga canadensis

Eastern hemlock

Tsuga spp.

Hemlock

4.2.2.1   Input objects

Biomass_speciesData requires the following input data layers

List of *Biomass_speciesData* input objects and their description.

objectName

objectClass

desc

sourceURL

rasterToMatchLarge

RasterLayer

a raster of `studyAreaLarge` in the same resolution and projection the simulation's. Defaults to the using the Canadian Forestry Service, National Forest Inventory, kNN-derived stand biomass map.

sppColorVect

character

A named vector of colors to use for plotting.   The names must be in sim*sppEquiv*[[*sim*sppEquivCol]], and should also contain a color for 'Mixed'

NA

sppEquiv

data.table

table of species equivalencies. See `LandR::sppEquivalencies_CA`.

studyAreaLarge

SpatialPolygonsDataFrame

Polygon to use as the parametrisation study area. Must be provided by the user. Note that `studyAreaLarge` is only used for parameter estimation, and can be larger than the actual study area used for LandR simulations (e.g, larger than `studyArea` in LandR Biomass_core).

NA

studyAreaReporting

SpatialPolygonsDataFrame

multipolygon (typically smaller/unbuffered than `studyAreaLarge` and `studyArea` in LandR Biomass_core) to use for plotting/reporting. If not provided, will default to `studyAreaLarge`.

NA

Of the inputs in Table 4.2.2.1, the following are particularly important and deserve special attention:

- `studyAreaLarge` – the polygon defining the area for which species cover data area desired. It can be larger (but never smaller) that the study area used in the simulation of forest dynamics (i.e., `studyArea` object in *Biomass_core*).

- `sppEquiv` – a table of correspondences between different species naming conventions. This table is used across several LandR modules, including *Biomass_core*. It is particularly important here because it will determine whether and how species (and their cover layers) are merged, if this is desired by the user. For instance, if the user wishes to simulate a generic *Picea spp.* that includes, *Picea glauca*, *Picea mariana* and *Picea engelmannii*, they will need to provide these three species names in the data column (e.g., `KNN` if obtaining forest attribute kNN data layers from the Canadian Forest Inventory), but the same name (e.g., "Pice_Spp") in the coumn chosen for the naming convention used throughout the simulation (the `sppEquivCol` parameter); see Table 4.2.2.1 for an example)

Example of species merging for simulation. Here the user wants to model *Abies balsamea*, *A. lasiocarpa* and *Pinus contorta* as separate species, but all *Picea* species as a generic *Picea spp.*. For this, all six species are identified in the 'KNN' column, so that their % cover layers can be obtained, but in the 'Boreal' column (which defines the naming convention used in the simulation in this example)

all *Picea* species have the same name. *Biomass_speciesData* will merge their % cover data into a single layer by summing their cover per pixel.

Species	KNN	Boreal	Modelled as
Abies balsamea	Abie_Bal	Abie_Bal	*Abies balsamea*
Abies lasiocarpa	Abie_Las	Abie_Las	*Abies lasiocarpa*
Picea engelmannii	Pice_Eng	Pice_Spp	*Picea spp.*
Picea glauca	Pice_Gla	Pice_Spp	*Picea spp.*
Picea mariana	Pice_Mar	Pice_Spp	*Picea spp.*
Pinus contorta	Pinu_Con	Pinu_Con	*Pinus contorta*

4.2.2.2 Parameters

Table 4.2.2.2 lists all parameters used in *Biomass_speciesData* and their detailed information.

List of *Biomass_speciesData* parameters and their description.

paramName

paramClass

default

min

max

paramDesc

coverThresh

integer

10

NA

NA

The minimum % cover a species needs to have (per pixel) in the study area to be considered present

dataYear

numeric

2001

NA

NA

Passed to `paste0('prepSpeciesLayers_', types)` function to fetch data from that year (if applicable). Defaults to 2001 as the default kNN year.

sppEquivCol

character

Boreal

NA

NA

The column in `sim$sppEquiv` data.table to group species by and use as a naming convention. If different species in, e.g., the kNN data have the same name in the chosen column, their data are merged into one species by summing their % cover in each raster cell.

types

character

KNN

NA

NA

The possible data sources. These must correspond to a function named paste0('prepSpeciesLayers_', types). Defaults to 'KNN' to get the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from year 'dataYear', using the `LandR::prepSpeciesLayers_KNN` function (see https://open.canada.ca/ data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for details on these data). Other currently available options are 'ONFRI', 'CASFRI', 'Pickell' and 'ForestInventory', which attempt to get proprietary data - the user must be granted access first. A custom function can be used to retrieve any data, just as long as it is accessible by the module (e.g., in the global environment) and is named as paste0('prepSpeciesLayers_', types).

vegLeadingProportion

numeric

0.8

0

1

a number that defines whether a species is leading for a given pixel. Only used for plotting.

.plotInitialTime

numeric

NA

NA

NA

This describes the simulation time at which the first plot event should occur

.plotInterval

numeric

NA

NA

NA

This describes the simulation time interval between plot events

.saveInitialTime

numeric

NA

NA

NA

This describes the simulation time at which the first save event should occur

.saveInterval

numeric

NA

NA

NA

This describes the simulation time interval between save events

.studyAreaName

character

NA

NA

NA

Human-readable name for the study area used. If NA, a hash of `studyAreaLarge` will be used.

.useCache

logical

init

NA

NA

Controls cache; caches the init event by default

.useParallel

numeric

16

NA

NA

Used in reading csv file with fread. Will be passed to data.table::setDTthreads.

Of the parameters listed in Table 4.2.2.2, the following are particularly important:

- `coverThresh` – integer. Defines a minimum % cover value (from 0-100) that the species must have in at least one pixel to be considered present in the study area, otherwise it is excluded from the final stack of species layers. Note that this will affect what species have data for an eventual simulation and the user will need to adjust simulation parameters (e.g., species in trait tables will need to match the species in the cover layers) accordingly.

- `types` – character. Which % cover data sources are to be used (see Detailed description). Several data sources can be passed, in which case the module will overlay the lower quality layers with higher quality ones **following the order of data sources specified by `types`** – i.e., if `types == c("KNN", "CASFRI", "ForestInventory")`, *KNN* is assumed to be the lowest quality data set and *ForestInventory* the highest: values in *KNN* layers are replaced with overlapping values from *CASFRI* layers and values from *KNN* and *CASFRI* layers are replaced with overlapping values of *ForestInventory* layers.

4.2.3   Simulation flow

The general flow of *Biomass_speciesData* processes is:

1. Download (if necessary) of and spatial processing of species cover layers from the first data source listed in the `types` parameter. Spatial processing consists in sub-setting the data to the area defined by `studyAreaLarge` and ensuring that the spatial projection and resolution match those of `rasterToMatchLarge`. After spatial processing, species layers that have no pixels with values >= to the `coverThresh` parameter are excluded.

2. If more than one data source is listed in `types`, the second set of species cover layers is downloaded and processed as above.

3. The second set of layers is assumed to be the highest quality dataset and used to replaced overlapping pixel values on the first (including for species whose layers may have been initially excluded after applying the `coverThresh` filter).

4. Steps 2 and 3 are repeated for remaining data sources listed in `types`.

5. Final layers are saved to disk and plotted. A summary of number of pixels with forest cover are calculated (`treed`and `numTreed` output objects; see Module outputs).

4.3 Usage example

4.3.1 Load `SpaDES` and other packages.

```r
if (!require(Require)) {
  install.packages("Require")
  library(Require)
}

Require(c("PredictiveEcology/SpaDES.install",
          "SpaDES", "PredictiveEcology/SpaDES.core@development",
          "PredictiveEcology/LandR"),
        install_githubArgs = list(dependencies = TRUE))
```

4.3.2 Get module, necessary packages and set up folder directories

```r
tempDir <- tempdir()
spadesModulesDirectory <- file.path(tempDir, "modules")

paths <- list(inputPath = normPath(file.path(tempDir, "inputs")),
              cachePath = normPath(file.path(tempDir, "cache")),
              modulePath = normPath(spadesModulesDirectory),
              outputPath = normPath(file.path(tempDir, "outputs")))

getModule("PredictiveEcology/Biomass_speciesData",
                    modulePath = spadesModulesDirectory, overwrite = TRUE)

## make sure all necessary packages are installed:
makeSureAllPackagesInstalled(spadesModulesDirectory)
```

4.3.3 Setup simulation

For this demonstration we are using all default parameter values, except `coverThresh` , which is lowered to 5%. The species layers (the major output of interest) are saved automatically, so there is no need to tell `spades` what to save using the `outputs` argument (see `?SpaDES.core::outputs`).

We pass the global parameter `.plotInitialTime = 1` in the `simInitAndSpades` function to activate plotting.

```r
# User may want to set some options -- see ?reproducibleOptions
#      -- e.g., often the path to the 'inputs' folder will be set outside of project by
# options(reproducible.inputPaths = "E:/Data/LandR_related/") # to re-use datasets acr
studyAreaLarge <- Cache(randomStudyArea, size = 1e7,
                         cacheRepo = paths$cachePath) # cache this so it creates a rand

# Pick the species you want to work with -- here we use the naming convention in "Bore
speciesNameConvention <- "Boreal"
speciesToUse <- c("Pice_Gla", "Popu_Tre", "Pinu_Con")

sppEquiv <- LandR::sppEquivalencies_CA[get(speciesNameConvention) %in% speciesToUse]
# Assign a colour convention for graphics for each species
sppColorVect <- LandR::sppColors(sppEquiv, speciesNameConvention,
                                  newVals = "Mixed", palette = "Set1")

## Usage example
modules <- list("Biomass_speciesData")
objects <- list("studyAreaLarge" = studyAreaLarge,
                "sppEquiv" = sppEquiv,
                "sppColorVect" = sppColorVect)
params <- list("Biomass_speciesData" = list("coverThresh" = 5L))
```

4.3.4   Run module

Note that because this is a data module (i.e., only attempts to prepare data for the simulation) we are not iterating it and so both the start and end times are set to 1 here.

```r
opts <- options(reproducible.useCache = TRUE,
                reproducible.inputPaths = paths$inputPath)

mySimOut <- simInitAndSpades(times = list(start = 1, end = 1),
                             modules = modules,
                             parameters = params,
                             objects = objects,
                             paths = paths,
                             .plotInitialTime = 1)
options(opts)
```

Here are some of the data retrieved by *Biomass_speciesData* for a randomly generated study area within Canada.
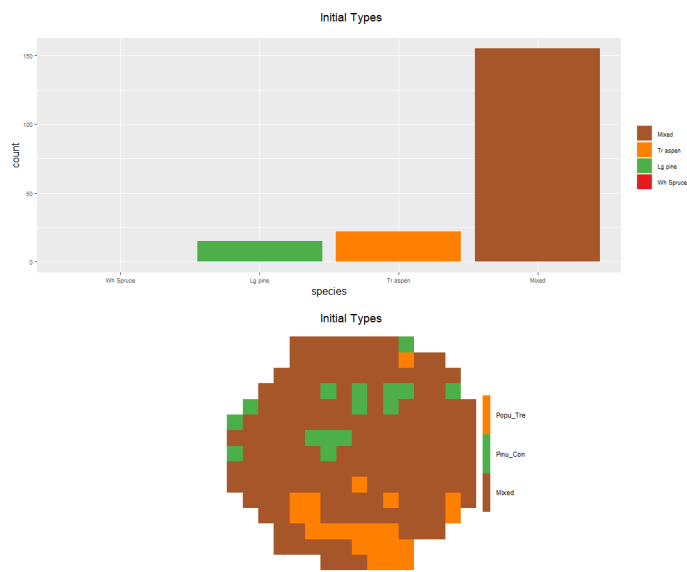
Figure 4.1: *Biomass_speciesData* automatically generates a plot of species dominance and number of presences in the study area when '.plotInitialTime=1' is passed as an argument.

4.4   References

Chapter 5

LandR *Biomass_borealDataPrep* Module

This documentation is work in progress. Potential discrepancies and omissions may exist for the time being. If you find any, do contact us using the link above^^

5.0.0.1 Authors:

Yong Luo yong.luo@canada.ca [aut], Eliot J B McIntire eliot.mcintire@canada.ca [aut, cre], Ceres Barros cbarros@mail.ubc.ca [ctb], Alex M. Chubaty achubaty@for-cast.ca [ctb]

5.1 Module Overview

5.1.1 Module summary

This module converts open datasets that are available for all of Canada's forests, into the input requirements for *Biomass_core*. It has been designed and tested for some parts of the Western Boreal Forest.

Specifically, it takes the stand biomass, stand age (defaulting to the Canadian Forest Inventory kNN-derived biomass/age maps), land-cover (Land Cover of

Canada map by default) and ecological zonation maps of Canada (ecodistricts by default), as well as species specific % cover maps of Canada (defaulting to Canadian Forest Inventory kNN-derived species % cover maps) and to i) statistically estimate species growth and establishment traits used in *Biomass_core*, and ii) define initial species biomass and age per pixel used by *Biomass_core* to start the simulation. It also defines ecolocations (groups of biophysically similar pixels, by default a combination of land-cover and ecozonation) used in the simulation.

Other species traits are taken from publicly available tables used by Dominic Cyr for LANDIS-II simulations, with some exceptions (see below).

Keeping data preparation outside of the LandR *Biomass_core* module maintains the modularity of the LandR modules.

5.1.2   Module inputs and parameters at a glance

Biomass_borealDataPrep requires internet access to retrieve default data. Raw data layers downloaded by the module are saved in `dataPath(sim)`, which can be controlled via `options(reproducible.destinationPath = ...)`.

We advise future users to run *Biomass_borealDataPrep* with defaults and inspect what the objects are like before supplying their own data, or alternative dataURLs. *Biomass_borealDataPrep* is meant to parametrise *Biomass_core* for Western Canadian boreal forests, but provides a good foundation to develop other other modules aimed at different geographical contexts.

Below are the full lists of input objects (Table 5.1.2) and parameters (Table 5.1.2) that *Biomass_borealDataPrep* expects. The only inputs that **must** be provided (i.e., *Biomass_borealDataPrep* does not have a default for) are `studyArea` (the study area used to simulate forest dynamics *Biomass_core*) and `studyAreaLarge` (a potentially larger study area used to derive parameter values – e.g., species traits). All other input objects and parameters have internal defaults (see Tables 5.2.4.1 and 5.2.4.2).

List of *Biomass_borealDataPrep* input objects and their description.

objectName

desc

cloudFolderID

The google drive location where cloudCache will store large statistical objects

columnsForPixelGroups

The names of the columns in `cohortData` that define unique pixelGroups. Default is c('ecoregionGroup', 'speciesCode', 'age', 'B')

ecoregionLayer

A `SpatialPolygonsDataFrame` that characterizes the unique ecological regions (`ecoregionGroup`) used to parameterize the biomass, cover, and species establishment probability models. It will be overlaid with landcover to generate classes for every ecoregion/LCC combination. It must have same extent and crs as `studyAreaLarge`. It is superseded by `sim$ecoregionRst` if that object is supplied by the user

ecoregionRst

A raster that characterizes the unique ecological regions used to parameterize the biomass, cover, and species establishment probability models. If this object is provided, it will supercede `sim$ecoregionLayer`. It will be overlaid with landcover to generate classes for every ecoregion/LCC combination. It must have same extent and crs as `rasterToMatchLarge` if supplied by user - use `reproducible::postProcess`. If it uses an attribute table, it must contain the field 'ecoregion' to represent raster values

rstLCC

A land classification map in study area. It must be 'corrected', in the sense that: 1) Every class must not conflict with any other map in this module (e.g., `speciesLayers` should not have data in LCC classes that are non-treed); 2) It can have treed and non-treed classes. The non-treed will be removed within this module if `P(sim)$omitNonTreedPixels` is `TRUE`; 3) It can have transient pixels, such as 'young fire'. These will be converted to a the nearest non-transient class, probabilistically if there is more than 1 nearest neighbour class, based on `P(sim)$LCCClassesToReplaceNN`. The default layer used, if not supplied, is Canada national land classification in 2010. The metadata (res, proj, ext, origin) need to match `rasterToMatchLarge`.

rasterToMatch

A raster of the `studyArea` in the same resolution and projection as `rawBiomassMap`. This is the scale used for all outputs for use in the simulation. If not supplied will be forced to match the default `rawBiomassMap`.

rasterToMatchLarge

A raster of the `studyAreaLarge` in the same resolution and projection as `rawBiomassMap`. This is the scale used for all inputs for use in the simulation. If not supplied will be forced to match the default `rawBiomassMap`.

rawBiomassMap

total biomass raster layer in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in tonnes/ha), unless 'dataYear' != 2001. If necessary, biomass values are rescaled to match changes in resolution. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.

speciesLayers

cover percentage raster layers by species in Canada species map. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from 2001 using a cover threshold of 10 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

speciesTable

a table of invariant species traits with the following trait colums: 'species', 'Area', 'longevity', 'sexualmature', 'shadetolerance', 'firetolerance', 'seeddistance_eff', 'seeddistance_max', 'resproutprob', 'resproutage_min', 'resproutage_max', 'postfireregen', 'leaflongevity', 'wooddecayrate', 'mortalityshape', 'growthcurve', 'leafLignin', 'hardsoft'. Names can differ, but not the column order. Default is from Dominic Cyr and Yan Boulanger's project

sppColorVect

named character vector of hex colour codes corresponding to each species

sppEquiv

table of species equivalencies. See `?LandR::sppEquivalencies_CA`.

sppNameVector

an optional vector of species names to be pulled from `sppEquiv`. If not provided, then species will be taken from the entire `P(sim)$sppEquivCol` in `sppEquiv`. See `LandR::sppEquivalencies_CA`.

standAgeMap

stand age map in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived biomass map from 2001, unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

studyArea

Polygon to use as the study area. Must be supplied by the user.

studyAreaLarge

multipolygon (potentially larger than `studyArea`) used for parameter estimation, Must be supplied by the user. If larger than `studyArea`, it must fully contain it.

List of *Biomass_borealDataPrep* parameters and their description.

paramName

paramDesc

biomassModel

Model and formula for estimating biomass (B) from `ecoregionGroup` (currently `ecoregionLayer` LandCoverClass), `speciesCode`, `logAge` (gives a

downward curving relationship), and `cover`. Defaults to a LMEM, which can be slow if dealing with very large datasets (e.g. 36 000 points take 20min). For faster fitting try `P(sim)$subsetDataBiomassModel == TRUE`, or `quote(RcppArmadillo::fastLm(formula = B ~ logAge speciesCode ecoregionGroup + cover speciesCode ecoregionGroup))`. A custom model call can also be provided, as long as the 'data' argument is NOT included.

coverModel

Model and formula used for estimating cover from `ecoregionGroup` and `speciesCode` and potentially others. Defaults to a GLMEM if there are > 1 grouping levels. A custom model call can also be provided, as long as the 'data' argument is NOT included

coverPctToBiomassPctModel

Model to estimate the relationship between % cover and % biomass, referred to as `P(sim)$fitDeciduousCoverDiscount` It is a number between 0 and 1 that translates % cover, as provided in several databases, to % biomass. It is assumed that all hardwoods are equivalent and all softwoods are equivalent and that % cover of hardwoods will be an overesimate of the % biomass of hardwoods. E.g., 30% cover of hardwoods might translate to 20% biomass of hardwoods. The reason this discount exists is because hardwoods in Canada have a much wider canopy than softwoods.

deciduousCoverDiscount

This was estimated with data from NWT on March 18, 2020 and may or may not be universal. Will not be used if `P(sim)$fitDeciduousCoverDiscount == TRUE`

fitDeciduousCoverDiscount

If TRUE, this will re-estimate `P(sim)$fitDeciduousCoverDiscount` This may be unstable and is not recommended currently. If `FALSE`, will use the current default

dataYear

Used to override the default 'sourceURL' of KNN datasets (species cover, stand biomass and stand age), which point to 2001 data, to fetch KNN data for another year. Currently, the only other possible year is 2011.

ecoregionLayerField

the name of the field used to distinguish ecoregions, if supplying a polygon. Defaults to `NULL` and tries to use 'ECODISTRIC' where available (for legacy reasons), or the row numbers of `sim$ecoregionLayer`. If this field is not numeric, it will be coerced to numeric.

exportModels

Controls whether models used to estimate maximum B/ANPP (`biomassModel`) and species establishment (`coverModel`) probabilities are exported for posterior analyses or not. This may be important when models fail to converge or hit singularity (but can still be used to make predictions) and the user wants to investigate them further. Can be set to 'none' (no models are exported), 'all' (both are exported), 'biomassModel' or 'coverModel'. BEWARE: because this is intended for posterior model inspection, the models will be exported with data, which may mean very large simList(s)!

fireURL

A URL to a fire database, such as the Canadian National Fire Database, that is a zipped shapefile with fire polygons, an attribute (i.e., a column) named 'Year'. If supplied (omitted with `NULL` or `NA`), this will be used to 'update' age pixels on `standAgeMap` with 'time since fire' as derived from this fire polygons map. Biomass is also updated in these pixels, when the last fire is more recent than 1986. If `NULL` or `NA`, no age and biomass imputation will be done in these pixels.

fixModelBiomass

should `modelBiomass` be fixed in the case of non-convergence? Only scaling of variables and attempting to fit with a new optimizer are implemented at this time

forestedLCCClasses

The classes in the `rstLCC` layer that are 'treed' and will therefore be run in Biomass_core. Defaults to forested classes in LCC2010 map.

imputeBadAgeModel

Model and formula used for imputing ages that are either missing or do not match well with biomass or cover. Specifically, if biomass or cover is 0, but age is not, or if age is missing (`NA`), then age will be imputed.

LCCClassesToReplaceNN

This will replace these classes on the landscape with the closest forest class `P(sim)$forestedLCCClasses`. If the user is using the LCC 2005 land-cover data product for `rstLCC`, then they may wish to include 36 (cities – if running a historic range of variation project), and 34:35 (burns) Since this is about estimating parameters for growth, it doesn't make any sense to have unique estimates for transient classes in most cases. If no classes are to be replaced, pass `'LCCClassesToReplaceNN' = numeric(0)` when supplying parameters.

minCoverThreshold

Pixels with total cover that is equal to or below this number will be omitted from the dataset

minRelativeBFunction

A quoted function that makes the table of min. relative B determining a stand shade level for each ecoregionGroup. Using the internal object `pixelCohortData` is advisable to access/use the list of `ecoregionGroups` per pixel. The function must output a `data.frame` with 6 columns, named `ecoregionGroup` and 'X1' to 'X5', with one line per `ecoregionGroup` code, and the min. relative biomass for each stand shade level X1-5. The default function uses values from LANDIS-II available at: https://github.com/dcyr/LANDIS-II_IA_generalUseFiles/blob/master/LandisInputs/BSW/biomass-succession-main-inputs_BSW_Baseline.txt%7E.

omitNonTreedPixels

Should this module use only treed pixels, as identified by `P(sim)$forestedLCCClasses`?

overrideBiomassInFires

should B values be re-estimated using Biomass_core for pixels within the fire perimeters obtained from `P(sim)$fireURL`, based on their time since fire age?

pixelGroupAgeClass

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

pixelGroupBiomassClass

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

rmImputedPix

Should `sim$imputedPixID` be removed from the simulation?

speciesUpdateFunction

Unnamed list of (one or more) quoted functions that updates species table to customize values. By default, `LandR::speciesTableUpdate` is used to change longevity and shade tolerance values, using values appropriate to Boreal Shield West (BSW), Boreal Plains (BP) and Montane Cordillera (MC) ecoprovinces (see `?LandR::speciesTableUpdate` for details). Set to `NULL` if default trait values from `speciesTable` are to be kept instead. The user can supply other or additional functions to change trait values (see `LandR::updateSpeciesTable`)

sppEquivCol

The column in `sim$speciesEquivalency` data.table to use as a naming convention.

speciesTableAreas

One or more of the Ecoprovince short forms that are in the `speciesTable` file, e.g., BSW, MC etc. Default is good for Alberta and other places in the western Canadian boreal forests.

subsetDataAgeModel

the number of samples to use when subsampling the age data model and when fitting `coverPctToBiomassPctModel`; Can be `TRUE`/`FALSE`/`NULL` or numeric; if `TRUE`, uses 50. If `FALSE`/`NULL` no subsetting is done.

subsetDataBiomassModel

the number of samples to use when subsampling the biomass data model (`biomassModel`); Can be `TRUE`/`FALSE`/`NULL` or numeric; if `TRUE`, uses 50. If `FALSE`/`NULL` no subsetting is done.

successionTimestep

defines the simulation time step, default is 10 years

useCloudCacheForStats

Some of the statistical models take long (at least 30 minutes, likely longer). If this is `TRUE`, then it will try to get previous cached runs from googledrive.

.plotInitialTime

This is here for backwards compatibility. Please use `.plots`

.plots

This describes the type of 'plotting' to do. See `?Plots` for possible types. To omit, set to NA

.plotInterval

This describes the simulation time interval between plot events

.saveInitialTime

This describes the simulation time at which the first save event should occur

.saveInterval

This describes the simulation time interval between save events

.seed

Named list of seeds to use for each event (names). E.g., `list('init' = 123)` will `set.seed(123)` at the start of the init event and unset it at the end. Defaults to `NULL`, meaning that no seeds will be set

.studyAreaName

Human-readable name for the study area used. If `NA`, a hash of studyArea will be used.

.useCache

Internal. Can be names of events or the whole module name; these will be cached by SpaDES

5.1.3 Events

The following events take place during a *Biomass_borealDataPrep* run. Note that this module only runs once (in one "time step").

- Module initiation (`init` event): after downloading all the necessary data (during the `.inputObjects` event), the module prepares the necessary objects and parameters for the simulation (see Detailed description). Depending on the objects, some are parametrised using empirical models, others based on literature [e.g., longevity values for western boreal species taken from 4], or expert knowledge (e.g., `sufficientLight` values adjusted to reflect western boreal forest succession dynamics) – – see `?LandR::speciesTableUpdate`.
- Plotting event: plots the estimated spatially-varying trait values.
- Saving event: saves any objects passed to `spades(..., outputs)`

5.1.4 Module outputs

The module produces the following outputs (Table 5.1.4):

List of *Biomass_borealDataPrep* output objects and their description.

objectName

desc

biomassMap

total biomass raster layer in study area, filtered for pixels covered by cohortData. Units in g/m2

cohortData

initial community table, containing corrected biomass (g/m2), age and species cover data, as well as ecolocation and `pixelGroup` information. This table defines the initial community composition and structure used by `Biomass_core`

ecoregion

`ecoregionGroup` look up table

ecoregionMap

`ecoregionGroup` map that has mapcodes match ecoregion table and `speciesEcoregion` table

imputedPixID

A vector of pixel IDs - matching rasterMatch IDs - that suffered data imputation. Data imputation may be in age (to match last fire event post 1950s, or 0 cover), biomass (to match fire-related imputed ages, correct for missing values or for 0

age/cover), land cover (to convert non-forested classes into to nearest forested class)

pixelGroupMap

initial community map that has mapcodes (`pixelGroup` IDs) match `cohortData`

pixelFateDT

A small table that keeps track of the pixel removals and cause. This may help diagnose issues related to understanding the creation of `cohortData`

minRelativeB

minimum relative biomass thresholds that determine a shade level in each pixel. X0-5 represent site shade classes from no-shade (0) to maximum shade (5).

modelCover

If `P(sim)$exportModels` is 'all', or 'cover', fitted cover model, as defined by `P(sim)$coverModel`.

modelBiomass

If `P(sim)$exportModels` is 'all', or 'biomass', fitted biomass model, as defined by `P(sim)$biomassModel`

rawBiomassMap

total biomass raster layer in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map (in tonnes/ha) from 2001, unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

species

a table that of invariant species traits. Will have the same traits as the input `speciesTable` with values adjusted where necessary

speciesEcoregion

table of spatially-varying species traits (`maxB`, `maxANPP`, `establishprob`), defined by species and `ecoregionGroup`)

studyArea

Polygon to use as the study area corrected for any spatial properties' mismatches with respect to `studyAreaLarge`.

sufficientLight

Probability of germination for species shade tolerance (in `species`) and shade level(`defined by`minRelativeB') combinations. Table values follow LANDIS-II test traits available at: https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt

5.1.5 Links to other modules

Intended to be used with *Biomass_core*, but can also be linked with other data modules that prepare inputs (e.g., *Biomass_speciesData* may be used upstream from *Biomass_borealDataPrep* to prepare species % cover layers using multiple data sources). You can see all *potential* module linkages within the LandR ecosystem here. Select *Biomass_borealDataPrep* from the drop-down menu to see linkages.

5.1.6 Getting help

- https://github.com/PredictiveEcology/Biomass_borealDataPrep/issues

5.2 Module manual

5.2.1 Detailed description

Biomass_borealDataPrep prepares all inputs necessary to run a realistic simulation of forest dynamics in western Canada boreal forests using *Biomass_core*. Part of this process involves cleaning up the input data and imputing missing data in some cases, which are presented thoroughly in Data acquisition and treatment.

After the cleaning and formatting the raw input data, the module prepares:

1. **invariant species traits** – spatio-temporally constant traits that mostly influence population dynamics (e.g., growth, mortality, dispersal) and responses to fire, and include the probabilities of germination for a given species tolerance and site shade combination (the `suffiencientLight` table) which link species shade tolerance values (`shadetolerance`) with site shade (determined by `minRelativeB`) to simulate germination success in any given pixel;

2. **spatially-varying species traits** – traits that vary by ecolocation, a spatial grouping of biophysically similar pixels. These are maximum biomass (`maxB`), maximum above-ground net primary productivity (`maxANPP`) and species establishment probability (`SEP`);

3. one **ecolocation-specific parameter** – shade thresholds that result in successful germination (minimum relative biomass, `minRelativeB`);

4. the species cohort table (`cohortData`) and corresponding map (`pixelGroupMap`) used to initialise and track cohorts across the landscape.

By default, ecolocations are defined as the spatial combination of ecodistricts of the National Ecological Framework for Canada, a broad-scale polygon system that captures sub-regional variation, and the Land Cover of Canada 2010 map, a raster-based database that distinguishes several forest and non-forest land-cover types. As *Biomass_core* only simulates trees, *Biomass_borealDataPrep* prepares all inputs and estimates parameters in pixels within forested land-cover classes (see Defining simulation pixels and ecolocations).

Note that ecolocations are called `ecoregionGroup`'s across LandR modules.

If a `studyAreaLarge` is supplied, the module uses it for all parameter estimation to account for larger spatial variability. It begins by calculating species biomass per pixel, multiplying the observed species % cover by the observed stand biomass and an adjustment factor, which can be statistically calibrated for the study area (a default value can also be used instead if `P(sim)$fitDeciduousCoverDiscount == FALSE`). Given that this adjusts the species biomass, this calibration step contributes to the calibration of `maxB` and `maxANPP` trait values, whose estimation is also based on species biomass. *Biomass_borealDataPrep* then estimates `maxB`, `maxANPP` and `SEP` from species biomasses per pixel using linear mixed effects models (LMEMs) by default (see Maximum biomass and maximum aboveground net primary productivity and Species establishment probability).

Invariant species traits, the probabilities of germination for a given shade tolerance and site shade and biomass thresholds defining site shade levels (`minRelativeB`) were obtained from a combination of published literature [e.g., longevity values followed 4] and values used in LANDIS-II applications in Canada's boreal forests. Default `minRelativeB` values are kept constant across all ecolocations due to the lack of data needed to derive ecolocation-specific values (see Minimum relative biomass). They are also adjusted by lowering the values of higher shade classes to reflect lower shade levels observed in Western Canadian forests with respect to their Eastern counterparts at similar density levels [11], which are likely driven by higher moisture limitation in the west [7, 13]. This adjustment can be by-passed by either supplying a `minRelativeB` table, or an alternative function call to `P(sim)$minRelativeBFunction` (which by default is `LandR::makeMinRelativeB`; see Minimum relative biomass for further detail).

After parameter estimation, *Biomass_borealDataPrep* performs data-based landscape initialisation, by creating tree species cohorts in forested pixels with age equal to the observed stand age and the previously calculated biomass.

In the next sections, we describe in greater detail the various data processing and parameter estimation steps carried out by *Biomass_borealDataPrep*.

5.2.2 Data acquisition and treatment

The only two objects that the user must supply are shapefiles that define the study area used to derive parameters (`studyAreaLarge`) and the study area where the simulation will happen (`studyArea`). The two objects can be identical if the user chooses to parametrise and run the simulations in the same area. If not identical, `studyArea` must be fully within `studyAreaLarge`. If `studyAreaLarge` and `studyArea` are in Canada, the module is able to automatically estimate and prepare all input parameters and objects for *Biomass_core*, as the default raw data are FAIR data [*sensu* 22] at the national-scale.

If no other inputs are supplied, *Biomass_borealDataPrep* will create raster versions of rasterize `studyAreaLarge` and `studyArea` (`rasterToMatchLarge` and `rasterToMatch`, respectively), using the stand biomass map layer (`rawBiomassMap`) as a template (i.e., the source of information for spatial resolution)

5.2.2.1 Defining simulation pixels and ecolocations

Biomass_borealDataPrep uses land-cover data to define and assign parameter values to the pixels where forest dynamics will be simulated (forested pixels).

By default it uses land-cover classes from the Land Cover Map of Canada 2010 v1 product. Pixels with classes 1 to 6 are included as forested pixels. It is possible to supply other land-cover products and where these include transient cover types (e.g., recent burns) the user may pass a vector of transient class IDs (via `LCCClassesToReplaceNN`) that will be reclassified as a "stable" forested class. The reclassification is done by searching the focal neighbourhood for a replacement forested cover class (up to a radius of 1250m from the focal cell). If no forested class is found within this perimeter, the pixel is not used to simulate forest dynamics. Reclassified pixels are omitted from the fitting of statistical models used for parameter estimation, but are assigned predicted values from these models.

Sub-regional spatial variation in `maxBiomass`, `maxANPP`, and `SEP` species traits is accounted for by ecolocation. Ecolocations are used as proxies for biophysical variation across the landscape when estimating model parameters that vary spatially. By default, they are defined as the combination of "ecodistricts" from the National Ecological Framework for Canada and the above land cover, but the user can change this by supplying different ecozonation or land-cover layers.

5.2.2.2 Species cover

Species percent cover (% cover) can be automatically obtained and pre-processed by *Biomass_borealDataPrep*. The module ensures that: 1. all data use the same geospatial geometries; 2. all layers these are correctly re-projected to

`studyAreaLarge` and `rasterToMatchLarge`; 3. species with no cover values above 10% are excluded.

By default it uses species % cover rasters derived from the MODIS satellite imagery from 2001, obtained from the Canadian National Forest Inventory [3] – hereafter 'kNN species data'.

5.2.2.3   Initial species age and biomass per pixel

Stand age and stand aboveground biomass (hereafter 'stand biomass') are used to derive parameters and define initial species age and biomass across the landscape. They are also derived from MODIS satellite imagery from 2001 prepared by the NFI [3], by default. *Biomass_borealDataPrep* downloads these data and performs a number of data harmonization operations to deal with data inconsistencies.

It first searches for mismatches between stand age (`standAge`), stand biomass (`standB`) and total stand cover (`standCover`), assuming that cover is the most accurate of the three, and biomass the least, and in the following order:

1. Pixels with `standCover < 5%` are removed;

2. Pixels with `standAge == 0`, are assigned `standB == 0`;

3. Pixels with `standB == 0`, are assigned `standAge == 0`.

Then, species is assigned one cohort per pixel according to the corrected stand age, stand biomass and % cover values. Cohort age is assumed to be the same as stand age and biomass is the product of stand biomass and species % cover. Before doing so, stand cover is rescaled to vary between 0 and 100%.

A next set of data inconsistencies in cohort age (`age`), biomass (`B`) and cover (`cover`) is looked for and solved in the following order:

4. if `cover > 0` and `age == 0`, `B` is set to 0 (and stand biomass recalculated);

5. if `cover == 0` and `age > 0`, or if `age == NA`, `age` is empirically estimated using the remainder of the data to fit the model supplied by `P(sim)$imputeBadAgeModel`, which defaults to:

```
## [[1]]
## lme4::lmer(age ~ log(totalBiomass) * cover * speciesCode + (log(totalBiomass) |
##     initialEcoregionCode))
```

Cohort biomass is then adjusted to reflect the different cover to biomass relationship of conifer and broadleaf species (see Adjustment of species biomass).

Finally, *Biomass_borealDataPrep* can use fire perimeters to correct stand ages. For this, it downloads the latest fire perimeter data from the Canadian Wildfire Data Base and changes pixel age inside fire perimeters to match the time since last fire, using fire years up to the first year of the simulation.

This assumes that the 1) *last fire was a stand replacing fire* and 2) that the *first year of the simulation is later than the first fire year* in the fire perimeter data. If the user does not want to assume 1), this data imputation step can be bypassed by setting the parameter `P(sim)$overrideBiomassInFires` to `FALSE` or `P(sim)$fireURL` to `NULL` or `NA`.

In pixels were ages are changed to match time since the last fire, cohort biomass needs to be corrected – in our default datasets we have noticed how it is overly inflated in many cases. For this the module uses a spin-up simulation that grows cohorts to their fixed age inside each pixel using estimated `maxB` and `maxANPP` parameters (see Maximum biomass and maximum aboveground net primary productivity).

Note that pixels that had data imputation can be removed from the simulation by setting `P(sim)$rmImputedPix == TRUE`.

5.2.2.4 Invariant species traits

Most species traits that do not vary spatio-temporally are obtained from available species trait tables used in LANDIS-II applications in Canada's boreal forests (available in Dominic Cyr's GitHub repository). Some are then adapted with minor adjustments to match Western Canadian boreal forests using published literature. Others (key growth and mortality traits) are estimated using statistical models.

The LANDIS-II species trait table contains species trait values for each Canadian Ecozone [12], which are by default filtered to the Boreal Shield West (BSW), Boreal Plains (BP) and Montane Cordillera Canadian Ecozones (via `P(sim)$speciesTableAreas`). Most trait values do not vary across these ecozones, but when they do, took the minimum value is used.

The function `LandR::speciesTableUpdate` is used by default to do further adjustments to trait values in this table (if this is not intended, a custom function call or `NULL` can be passed to `P(sim)$speciesUpdateFunction`): - Longevity values are adjusted to match the values from Burton and Cumming [4], which match BSP, BP and MC ecozones. These adjustments result in higher longevity for most species; - Shade tolerance values are lowered for *Abies balsamifera*, *Abies lasiocarpa*, *Picea engelmanii*, *Picea glauca*, *Picea mariana*, *Tsuga heterophylla* and *Tsuga mertensiana* to better **relative** shade tolerance levels in Western Canada. Because these are relative shade tolerances, the user should **always** check these values with respect to their own study areas and species pool.

The user can also pass more than one function call to `P(sim)$speciesUpdateFunction` if they want to make other adjustments in addition to those listed above (see `?LandR::updateSpeciesTable`).

Finally, the **probabilities of germination** (`suffiencientLight` table) are taken by default from a LANDIS-II test table.

5.2.3   Parameter estimation/calibration

5.2.3.1   Adjustment of species biomass

Biomass_core requires initial values of species-specific aboveground biomass (B) for every pixel that is tracked. *Biomass_borealDataPrep* estimates these based on stand biomass (`standB`) and individual species % cover. Initial `B` is estimated for each species in each pixel by multiplying `standB` by species % cover. Because the default cover layers are satellite-derived, the relationship between relative cover and relative biomass of broadleaf and conifer species needs to be adjusted to reflect their different canopy architectures (using `P(sim)$deciduousCoverDiscount`).

By default, *Biomass_borealDataPrep* uses a previously estimated `P(sim)$deciduousCoverDiscount` based on the Northwest Territories data. However, the user can chose to re-estimate it by setting `P(sim)$fitDeciduousCoverDiscount == TRUE`. In this case, by default *Biomass_borealDataPrep* will fit the the following model:

```
## [[1]]
## glm(I(log(B/100)) ~ logAge * I(log(totalBiomass/100)) * speciesCode *
##     lcc)
```

which relates the estimated biomass (`B`) with an interaction term between log-age (`logAge`), `standB` ('totalBiomass' above), `speciesCode` (i.e. species ID) and land cover ('lcc' above). The model is fitted to the `standB` and species cover on `studyAreaLarge`, using an optimization routine that searches for the best conversion factor between broadleaf species cover and `B` by minimizing AIC.

5.2.3.2   Maximum biomass and maximum aboveground net primary productivity

Biomass_borealDataPrep statistically estimates maximum biomass (`maxB`), maximum aboveground net primary productivity (`maxANPP`) using the processed species ages and biomass.

`maxB` is estimated by modelling the response of species-specific biomass (`B`) to species age and cover, while accounting for variation among ecolocations (`ecoregionGroup` below):

```
## [[1]]
## lme4::lmer(B ~ logAge * speciesCode + cover * speciesCode + (logAge +
##     cover | ecoregionGroup))
```

The coefficients are estimated by maximum likelihood and model fit is calculated as the proportion of explained variance explained by fixed effects only (marginal r2) and by the entire model (conditional r2) – both of which are printed as messages.

Because the model can take a while to fit, it is possible to sample pixels within each species and ecolocation combination via the `P(sim)$subsetDataBiomassModel` parameter. The module also attempts to refit the statistical model by re-sampling the data, re-fitting `lmer` with the `bobyqa` optimizer, and re-scaling the continuous predictors (`cover` and `logAge`) when there are convergence issues and `P(sim)$fixModelBiomass == TRUE`. These steps are tried additively until the convergence issue is resolved. If the module is still unable to solve the converge issue an message is printed and the module uses the last model it refit. Note that convergence issues are not usually problematic for parameter estimation - only for estimation of parameter standard errors. However, the user should always inspect the final model (especially if not converged) and make sure that the problems are not significant – if they are an alternative model call can be supplied via the `P(sim)$biomassModel` parameter. Note that if supplying a model call that does not use `lme4::lmer` the refitting process is likely to fail and may have to be turned off (via the `P(sim)$fixModelBiomass` parameter).

`maxB` is then predicted by species and ecolocation combination, by setting species cover to 100% and species log-age to the log of species longevity. When using `Biomass_speciesParameters`, `maxB` is calibrated so that species can achieve the maximum observed biomass during the simulation.

`maxANPP` is calculated as `maxB * mANPPproportion/100`, where `mANPPproportion` defaults to 3.33 (the value used in LANDIS-II applications in Canada's boreal forests), unless calibrated by *Biomass_speciesParameters*.

5.2.3.3 Minimum relative biomass

Minimum relative biomass (`minRelativeB`) is a spatially-varying parameter used to determine the shade level in each pixel.

Since we found no data to base the parametrisation of this trait, default values are based on publicly available values used in LANDIS-II applications in Canada's boreal forests (available in Dominic Cyr's GitHub repository), where all ecolocations shared the same values.

Initial runs revealed excessive recruitment of moderately shade intolerant species even as stand biomass increased, so values for shade levels X4 and X5 are ad-

justed downwards (X4: 0.8 to 0.75; X5: 0.90 to 0.85) to reflect higher competition for resources (e.g. higher water limitation) in Western Canadian forests with regards to Eastern Canadian forests [11].

The minimum biomass threshold of a shade level of `X0` is `0 standB`.

5.2.3.4   Species establishment probability

Species establishment probability (`SEP`) is estimated by modelling the probability of observing a given species in each ecolocation. For this, *Biomass_borealDataPrep* models the relationship between probability of occurrence of a species (π) using the following model by default:

```
## [[1]]
## glm(cbind(coverPres, coverNum - coverPres) ~ speciesCode * ecoregionGroup,
##     family = binomial)
```

whereby the probability of occurrence of a species (π) – calculated as the number of pixels with % cover > 0 divided by the total number of pixels, by species within each ecolocation – is modelled per species and ecolocation (`ecoregionGroup` above) following a binomial distribution (with a logit link). There is no data sub-sampling done before fitting the `SEP` statistical model, as the model fits quite fast even for very large sample sizes (e.g., > 20 million points).

`SEP` is then predicted by species and ecolocation combination, by setting species cover to 100%.

5.2.3.5   Updating species growth/mortality traits *Biomass_speciesParameters*

If using *Biomass_borealDataPrep* and *Biomass_speciesParameters*, the later module calibrates several species traits that are first prepared by *Biomass_borealDataPrep*: - `growthcurve`, `mortalityshape` – which initially come from publicly available LANDIS-II tables - `maxBiomass`, `maxANPP` – which are estimated statistically (see Maximum biomass and maximum aboveground net primary productivity)

Briefly, *Biomass_speciesParameters*:

1. Uses ~41,000,000 hypothetical species' growth curves (generated with *Biomass_core*), that cover a fully factorial combination of longevity, ratio of `maxANPP` to `maxBiomass`, `growthcurve`, `mortalityshape`;

2. Takes permanent and temporary sample plot (PSP) data in or near the study area for the target species, and finds which hypothetical species'

growth curve most closely matches the growth curve observed in the PSP data – on a species-by-species base. This gives us each species' `growthcurve`, `mortalityshape`, and `mANPPproportion`, a ratio of maximum aboveground net primary productivity (`maxANPP`) to maximum biomass (`maxBiomass`, not to be confounded with `maxB`) in the study area.

3. Introduces a new parameter, `inflationFactor`, and re-calibrates `maxB`. We recognize that `maxB`, as obtained empirically by *Biomass_borealDataPrep*, cannot be easily reached in simulations because all reasonable values of `growthcurve`, `mortalityshape` and `longevity` prevent the equation from reaching `maxB` (it acts as an asymptote that is never approached). The `inflationFactor` is calculated as the ratio of `maxBiomass` (the parameter used to generate theoretical growth curves in step 1) to the maximum biomass *actually* achieved by the theoretical growth curves (step 1). `maxB` is then recalibrated by multiplying it by `inflationFactor`. By doing this, resulting non-linear growth curves generated doing *Biomass_core* simulation will be able to achieve the the empirically estimated `maxB`.

4. Estimates species-specific `maxANPP` by multiplying the final `maxB` above by `mANPPproportion` (estimated in step 2).

In cases were there is not sufficient PSP data to perform the above steps, `maxB` and `maxANPP` are left as estimated by *Biomass_borealDataPrep* (see Maximum biomass and maximum aboveground net primary productivity).

5.2.4   Initialization, inputs and parameters

Biomass_borealDataPrep initializes itself and prepares all inputs provided it has internet access to retrieve the raw datasets used for parametrisation and preparing input objects for *Biomass_core*. Future users should run *Biomass_borealDataPrep* with defaults and inspect what the objects are like before supplying their own data, or alternative data URLs. Alternatively, user may develop their own module using *Biomass_borealDataPrep* as a template.

5.2.4.1   Input objects

Table 5.2.4.1 shows the full list of input objects used by the module.

List of *Biomass_borealDataPrep* input objects and their description.

objectName

objectClass

desc

sourceURL

cloudFolderID

character

The google drive location where cloudCache will store large statistical objects

NA

columnsForPixelGroups

character

The names of the columns in `cohortData` that define unique pixelGroups. Default is c('ecoregionGroup', 'speciesCode', 'age', 'B')

NA

ecoregionLayer

SpatialPolygonsDataFrame

A `SpatialPolygonsDataFrame` that characterizes the unique ecological regions (`ecoregionGroup`) used to parameterize the biomass, cover, and species establishment probability models. It will be overlaid with landcover to generate classes for every ecoregion/LCC combination. It must have same extent and crs as `studyAreaLarge`. It is superseded by `sim$ecoregionRst` if that object is supplied by the user

https://sis.agr.gc.ca/cansis/nsdb/ecostrat/district/ecodistrict_shp.zip

ecoregionRst

RasterLayer

A raster that characterizes the unique ecological regions used to parameterize the biomass, cover, and species establishment probability models. If this object is provided, it will supercede `sim$ecoregionLayer`. It will be overlaid with landcover to generate classes for every ecoregion/LCC combination. It must have same extent and crs as `rasterToMatchLarge` if supplied by user - use `reproducible::postProcess`. If it uses an attribute table, it must contain the field 'ecoregion' to represent raster values

NA

rstLCC

RasterLayer

A land classification map in study area. It must be 'corrected', in the sense that: 1) Every class must not conflict with any other map in this module (e.g., `speciesLayers` should not have data in LCC classes that are non-treed); 2) It can have treed and non-treed classes. The non-treed will be removed within this module if `P(sim)$omitNonTreedPixels` is `TRUE`; 3) It can have transient pixels,

such as 'young fire'. These will be converted to a the nearest non-transient class, probabilistically if there is more than 1 nearest neighbour class, based on `P(sim)$LCCClassesToReplaceNN`. The default layer used, if not supplied, is Canada national land classification in 2010. The metadata (res, proj, ext, origin) need to match `rasterToMatchLarge`.

NA

rasterToMatch

RasterLayer

A raster of the `studyArea` in the same resolution and projection as `rawBiomassMap`. This is the scale used for all outputs for use in the simulation. If not supplied will be forced to match the default `rawBiomassMap`.

NA

rasterToMatchLarge

RasterLayer

A raster of the `studyAreaLarge` in the same resolution and projection as `rawBiomassMap`. This is the scale used for all inputs for use in the simulation. If not supplied will be forced to match the default `rawBiomassMap`.

NA

rawBiomassMap

RasterLayer

total biomass raster layer in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in tonnes/ha), unless 'dataYear' != 2001. If necessary, biomass values are rescaled to match changes in resolution. See https://open.canada. ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2001-attributes_attributs-2001/NFI_MODIS250m_2001_kNN_Structure_Biomass_TotalLiveAboveGround_v1.tif

speciesLayers

RasterStack

cover percentage raster layers by species in Canada species map. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from 2001 using a cover threshold of 10 - see https://open.canada. ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2001-attributes_attributs-2001/

speciesTable

data.table

a table of invariant species traits with the following trait colums: 'species', 'Area', 'longevity', 'sexualmature', 'shadetolerance', 'firetolerance', 'seeddistance_eff', 'seeddistance_max', 'resproutprob', 'resproutage_min', 'resproutage_max', 'postfireregen', 'leaflongevity', 'wooddecayrate', 'mortalityshape', 'growthcurve', 'leafLignin', 'hardsoft'. Names can differ, but not the column order. Default is from Dominic Cyr and Yan Boulanger's project

https://raw.githubusercontent.com/dcyr/LANDIS-II_IA__generalUseFiles/master/speciesTraits.csv

sppColorVect

character

named character vector of hex colour codes corresponding to each species

NA

sppEquiv

data.table

table of species equivalencies. See `?LandR::sppEquivalencies_CA`.

NA

sppNameVector

character

an optional vector of species names to be pulled from `sppEquiv`. If not provided, then species will be taken from the entire `P(sim)$sppEquivCol` in `sppEquiv`. See `LandR::sppEquivalencies_CA`.

NA

standAgeMap

RasterLayer

stand age map in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived biomass map from 2001, unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

http://ftp.maps.canada.ca/pub/nrcan__rncan/Forests__Foret/canada-forests-attributes__attributs-forests-canada/2001-attributes__attributs-2001/NFI__MODIS250m__2001__kNN_Structure_Stand_Age_v1.tif

studyArea

SpatialPolygonsDataFrame

Polygon to use as the study area. Must be supplied by the user.

NA

studyAreaLarge

SpatialPolygonsDataFrame

multipolygon (potentially larger than `studyArea`) used for parameter estimation, Must be supplied by the user. If larger than `studyArea`, it must fully contain it.

NA

Of these inputs, the following are particularly important and deserve special attention:

- **Spatial layers**

    - `ecoregionLayer` or `ecoregionRst` – a shapefile or map containing ecological zones.

    - `rawBiomassMap` – a map of observed stand biomass (in g/m^2).

    - `rstLCC` – a land-cover raster.

    - `speciesLayers` – layers of species % cover data. The species must match those available in default (or provided) species traits tables (the `species` and `speciesEcoregion` tables).

    - `standAgeMap` – a map of observed stand ages (in years).

    - `studyArea` – shapefile. A `SpatialPolygonsDataFrame` with a single polygon determining the where the simulation will take place. This input object **must be supplied by the user**.

    - `studyAreaLarge` – shapefile. A `SpatialPolygonsDataFrame` with a single polygon determining the where the statistical models for parameter estimation will be fitted. It **must** contain `studyArea` fully, if they are not identical. This object **must be supplied by the user**.

- **Tables**

    - `speciesTable` – a table of invariant species traits that must have the following columns (even if not all are necessary to the simulation): "species", "Area", "longevity", "sexualmature", "shadetolerance", "firetolerance", "seeddistance_eff", "seeddistance_max", "resprout-prob", "resproutage_min", "resproutage_max", "postfireregen", "leaflongevity", "wooddecayrate", "mortalityshape", "growthcurve", "leafLignin", "hardsoft". The columns names can be different but not their order. See Scheller and Miranda [17] for details about these columns.

5.2.4.2   Parameters

Table 5.2.4.2 lists all parameters used in *Biomass_borealDataPrep* and their detailed information.

List of *Biomass_borealDataPrep* parameters and their description.

paramName

paramClass

default

min

max

paramDesc

biomassModel

call

lme4::lm. . . .

NA

NA

Model and formula for estimating biomass (B) from `ecoregionGroup` (currently `ecoregionLayer` LandCoverClass), `speciesCode`, `logAge` (gives a downward curving relationship), and `cover`. Defaults to a LMEM, which can be slow if dealing with very large datasets (e.g. 36 000 points take 20min). For faster fitting try `P(sim)$subsetDataBiomassModel == TRUE`, or `quote(RcppArmadillo::fastLm(formula = B ~ logAge speciesCode ecoregionGroup + cover speciesCode ecoregionGroup))`. A custom model call can also be provided, as long as the 'data' argument is NOT included.

coverModel

call

glm, cbi. . . .

NA

NA

Model and formula used for estimating cover from `ecoregionGroup` and `speciesCode` and potentially others. Defaults to a GLMEM if there are > 1 grouping levels. A custom model call can also be provided, as long as the 'data' argument is NOT included

coverPctToBiomassPctModel

call

glm, I(l. . . .

NA

NA

Model to estimate the relationship between % cover and % biomass, referred to as `P(sim)$fitDeciduousCoverDiscount` It is a number between 0 and 1 that translates % cover, as provided in several databases, to % biomass. It is assumed that all hardwoods are equivalent and all softwoods are equivalent and that % cover of hardwoods will be an overesimate of the % biomass of hardwoods. E.g., 30% cover of hardwoods might translate to 20% biomass of hardwoods. The reason this discount exists is because hardwoods in Canada have a much wider canopy than softwoods.

deciduousCoverDiscount

numeric

0.8418911

NA

NA

This was estimated with data from NWT on March 18, 2020 and may or may not be universal. Will not be used if `P(sim)$fitDeciduousCoverDiscount == TRUE`

fitDeciduousCoverDiscount

logical

FALSE

NA

NA

If TRUE, this will re-estimate `P(sim)$fitDeciduousCoverDiscount` This may be unstable and is not recommended currently. If `FALSE`, will use the current default

dataYear

numeric

2001

NA

NA

Used to override the default 'sourceURL' of KNN datasets (species cover, stand biomass and stand age), which point to 2001 data, to fetch KNN data for another year. Currently, the only other possible year is 2011.

ecoregionLayerField

character

NA

NA

the name of the field used to distinguish ecoregions, if supplying a polygon. Defaults to `NULL` and tries to use 'ECODISTRIC' where available (for legacy reasons), or the row numbers of `sim$ecoregionLayer`. If this field is not numeric, it will be coerced to numeric.

exportModels

character

NA

NA

Controls whether models used to estimate maximum B/ANPP (`biomassModel`) and species establishment (`coverModel`) probabilities are exported for posterior analyses or not. This may be important when models fail to converge or hit singularity (but can still be used to make predictions) and the user wants to investigate them further. Can be set to 'none' (no models are exported), 'all' (both are exported), 'biomassModel' or 'coverModel'. BEWARE: because this is intended for posterior model inspection, the models will be exported with data, which may mean very large simList(s)!

fireURL

character

https://....

NA

NA

A URL to a fire database, such as the Canadian National Fire Database, that is a zipped shapefile with fire polygons, an attribute (i.e., a column) named 'Year'. If supplied (omitted with `NULL` or `NA`), this will be used to 'update' age pixels on `standAgeMap` with 'time since fire' as derived from this fire polygons map. Biomass is also updated in these pixels, when the last fire is more recent than 1986. If `NULL` or `NA`, no age and biomass imputation will be done in these pixels.

fixModelBiomass

logical

FALSE

NA

NA

should `modelBiomass` be fixed in the case of non-convergence? Only scaling of variables and attempting to fit with a new optimizer are implemented at this time

forestedLCCClasses

numeric

1, 2, 3,. . . .

0

NA

The classes in the `rstLCC` layer that are 'treed' and will therefore be run in Biomass_core. Defaults to forested classes in LCC2010 map.

imputeBadAgeModel

call

lme4::lm. . . .

NA

NA

Model and formula used for imputing ages that are either missing or do not match well with biomass or cover. Specifically, if biomass or cover is 0, but age is not, or if age is missing (`NA`), then age will be imputed.

LCCClassesToReplaceNN

numeric

NA

NA

This will replace these classes on the landscape with the closest forest class `P(sim)$forestedLCCClasses`. If the user is using the LCC 2005 land-cover data product for `rstLCC`, then they may wish to include 36 (cities – if running a historic range of variation project), and 34:35 (burns) Since this is about estimating parameters for growth, it doesn't make any sense to have unique estimates for transient classes in most cases. If no classes are to be replaced, pass `'LCCClassesToReplaceNN' = numeric(0)` when supplying parameters.

minCoverThreshold

numeric

5

0

100

Pixels with total cover that is equal to or below this number will be omitted from the dataset

minRelativeBFunction

call

LandR::m. . . .

NA

NA

A quoted function that makes the table of min. relative B determining a stand shade level for each ecoregionGroup. Using the internal object `pixelCohortData` is advisable to access/use the list of `ecoregionGroups` per pixel. The function must output a `data.frame` with 6 columns, named `ecoregionGroup` and 'X1' to 'X5', with one line per `ecoregionGroup` code, and the min. relative biomass for each stand shade level X1-5. The default function uses values from LANDIS-II available at: https://github.com/dcyr/LANDIS-II_IA_generalUseFiles/blob/master/LandisInputs/BSW/biomass-succession-main-inputs_BSW_Baseline.txt%7E.

omitNonTreedPixels

logical

TRUE

FALSE

TRUE

Should this module use only treed pixels, as identified by `P(sim)$forestedLCCClasses`?

overrideBiomassInFires

logical

TRUE

NA

NA

should B values be re-estimated using Biomass_core for pixels within the fire perimeters obtained from `P(sim)$fireURL`, based on their time since fire age?

pixelGroupAgeClass

numeric

params(s. . . .

NA

NA

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

pixelGroupBiomassClass

numeric

100

NA

NA

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

rmImputedPix

logical

FALSE

NA

NA

Should `sim$imputedPixID` be removed from the simulation?

speciesUpdateFunction

list

LandR::s. . . .

NA

NA

Unnamed list of (one or more) quoted functions that updates species table to customize values. By default, `LandR::speciesTableUpdate` is used to change longevity and shade tolerance values, using values appropriate to Boreal Shield West (BSW), Boreal Plains (BP) and Montane Cordillera (MC) ecoprovinces (see `?LandR::speciesTableUpdate` for details). Set to `NULL` if default trait values from `speciesTable` are to be kept instead. The user can supply other or additional functions to change trait values (see `LandR::updateSpeciesTable`)

sppEquivCol

character

Boreal

NA

NA

The column in `sim$speciesEquivalency` data.table to use as a naming convention.

speciesTableAreas

character

BSW, BP, MC

NA

NA

One or more of the Ecoprovince short forms that are in the `speciesTable` file, e.g., BSW, MC etc. Default is good for Alberta and other places in the western Canadian boreal forests.

subsetDataAgeModel

numeric

50

NA

NA

the number of samples to use when subsampling the age data model and when fitting `coverPctToBiomassPctModel`; Can be `TRUE/FALSE/NULL` or numeric; if `TRUE`, uses 50. If `FALSE/NULL` no subsetting is done.

subsetDataBiomassModel

numeric

NA

NA

NA

the number of samples to use when subsampling the biomass data model (`biomassModel`); Can be `TRUE/FALSE/NULL` or numeric; if `TRUE`, uses 50. If `FALSE/NULL` no subsetting is done.

successionTimestep

numeric

10

NA

NA

defines the simulation time step, default is 10 years

useCloudCacheForStats

logical

TRUE

NA

NA

Some of the statistical models take long (at least 30 minutes, likely longer). If this is `TRUE`, then it will try to get previous cached runs from googledrive.

.plotInitialTime

numeric

start(sim)

NA

NA

This is here for backwards compatibility. Please use `.plots`

.plots

character

NA

NA

NA

This describes the type of 'plotting' to do. See `?Plots` for possible types. To omit, set to NA

.plotInterval

numeric

NA

NA

NA

This describes the simulation time interval between plot events

.saveInitialTime

numeric

NA

NA

NA

This describes the simulation time at which the first save event should occur

.saveInterval

numeric

NA

NA

NA

This describes the simulation time interval between save events

.seed

list

NA

NA

Named list of seeds to use for each event (names). E.g., `list('init' = 123)` will `set.seed(123)` at the start of the init event and unset it at the end. Defaults to `NULL`, meaning that no seeds will be set

.studyAreaName

character

NA

NA

NA

Human-readable name for the study area used. If `NA`, a hash of studyArea will be used.

.useCache

character

.inputOb. . . .

NA

NA

Internal. Can be names of events or the whole module name; these will be cached by SpaDES

Of these parameters, the following are particularly important:

- **Estimation of simulation parameters**

  - `biomassModel` – the statistical model (as a function call) used to estimate `maxB` and `maxANPP`.
  - `coverModel` – the statistical model (as a function call) used to estimate `SEP`.
  - `fixModelBiomass` – determines whether `biomassModel` is re-fit when convergence issues arise.

- – `imputeBadAgeModel` – model used to impute ages when they are missing, or do not match the input cover and biomass data. Not to be confounded with correcting ages from fire data
- – `subsetDataAgeModel` and `subsetDataBiomassModel` – control data sub-sampling for fitting the `imputeBadAgeModel` and `biomassModel`, respectively
- – `exportModels` – controls whether `biomassModel` or `coverModel` (or both) are to be exported in the simulation `simList`, which can be useful to inspect the fitted models and report on statistical fit.
- – `sppEquivCol` – character. the column name in the `speciesEquivalency` data.table that defines the naming convention to use throughout the simulation.

- **Data processing**

  - – `forestedLCCClasses` and `LCCClassesToReplaceNN` – define which land-cover classes in `rstLCC` are forested and which should be reclassified to forested classes, respectively.
  - – `deciduousCoverDiscount`, `coverPctToBiomassPctModel` and `fitDeciduousCoverDiscount` – the first is the adjustment factor for broadleaf species cover to biomass relationships; the second and third are the model used to refit `deciduousCoverDiscount` in the supplied `studyAreaLarge` and whether refitting should be attempted (respectively).

5.2.4.3 Outputs

- **Tables**

  - – `cohortData` – initial community table, containing corrected biomass (g/m2), age and species cover data, as well as ecolocation and `pixelGroup` information. This table defines the initial community composition and structure used by `Biomass_core`.
  - – `species` – table of invariant species traits. Will contain the same traits as in `speciesTable` above, but adjusted where necessary.
  - – `speciesEcoregion` – table of spatially-varying species traits (`maxB`, `maxANPP`, `SEP`).
  - – `minRelativeB` – minimum relative biomass thresholds that determine a shade level in each pixel. X0-5 represent site shade classes from no-shade (0) to maximum shade (5).
  - – `sufficientLight` – probability of germination for species shade tolerance (in `species`) and shade level(`defined by`minRelativeB')

- **Spatial layers**

– `biomassMap` – map of initial stand biomass values after adjustments for data mismatches.

– `pixelGroupMap` – a map containing `pixelGroup` IDs per pixel. This defines the initial map used for hashing within `Biomass_core`, in conjunction with `cohortData`.

– `ecoregionMap` – map of ecolocations.

5.2.5 Simulation flow

The general flow of *Biomass_borealDataPrep* processes is:

1. Preparation of all necessary data and input objects that do not require parameter fitting (e.g., invariant species traits table, creating ecolocations);

2. Fixing mismatched between raw cover, biomass and age data;

3. Imputing age values in pixels where mismatches exist or age data is missing;

4. Construction of an initial `data.table` of cohort biomass and age per pixel (with ecolocation information);

5. Sub-setting pixels in forested land-cover classes and (optional) converting transient land-cover classes to forested classes;

6. Fitting `coverModel`;

7. Fitting `biomassModel` (and re-fitting if necessary – optional);

8. Estimating `maxB`, `maxANPP` and `SEP` per species and ecolocation.

9. (Optional) Correcting ages in pixels inside fire perimeters and reassigning biomass.

10. (Optional) Plots of `maxB`, `maxANPP` and `SEP` maps.

5.3 Usage example

This module can be run stand-alone, but it won't do much more than prepare inputs for `Biomass_core`. Hence, we provide a usage example of this module and a few others in this repository and in Barros et al. [1].

5.4 References

Chapter 6

LandR *Biomass_speciesParameters* Module

Made with Markdown | Get help Report issues

This documentation is work in progress. Potential discrepancies and omissions may exist for the time being. If you find any, do contact us using the link above^^

6.0.0.1 Authors:

Ian Eddy ian.eddy@example.com [aut, cre]

6.1 Module Overview

6.1.1 Module summary

This module attempts to calibrate species growth and mortality trait values used in *Biomass_core*, by matching theoretical species' growth curves obtained with different trait values (generated by LandR `Biomass_speciesFactorial`; see Simulated species data) against observed growth curves derived from Permanent Sample Plots (PSP data) across Canada (see Permanent sample plot data), to find the combination of traits that allows a better match to the observed curves. In particular, it calibrates the `growthcurve`, `mortalityshape`, `maxANPP` and `maxB` traits (see Parameter estimation/calibration).

This module **will not** prepare other traits or parameters used in *Biomass_core* and so it is meant to be used in conjunction with another data module which does so (e.g. *Biomass_borealDataPrep*). However, it may be used stand-alone in an initial phase for easier inspection of the statistical calibration procedure.

Note that a Google Account is necessary to access the data.

As of 2022-03-18, the PSP data needed for this module is not freely available, and data sharing agreements must be obtained from the governments of SK, AB, and BC.

6.1.2   Module inputs and parameters at a glance

Biomass_speciesParameters requires an internet connection and authorized access to the default data used for the calibration (e.g., theoretical species growth curves and PSD data).

We advise future users to run *Biomass_speciesParameters* with defaults and inspect what the objects are like before supplying their own data or trying to run *Biomass_speciesFactorial* to generate their own theoretical curves.

Below are the full lists of input objects (Table 6.1) and parameters (Table 6.2) that *Biomass_speciesParameters* expects. The only inputs that **must** be provided (i.e., *Biomass_speciesParameters* does not have a default for) is `studyAreaANPP` (the study area used extract the PSP data from). All other input objects and parameters have internal defaults, but the user may need to request access to their online files (see Tables 6.4 and 6.5).

6.1.3   Events

The following events take place during a *Biomass_speciesParameters* run. Note that this module only runs once (in one "time step") and only executes one event (`init`).

- Module initiation (`init` event): after downloading all the necessary data (during the `.inputObjects` event), the module prepares the necessary objects and parameters for the simulation and performs the calibration (see Detailed description).

See Simulation flow for further detail.

6.1.4   Module outputs

The module produces the following outputs (Table 6.3):

Table 6.1: List of *Biomass_speciesParameters* input objects and their description.

objectName	desc
factorialSpeciesTable	table with species traits for matching to 'reducedFactorialCohortData'
reducedFactorialCohortData	results of factorial species trait simulation. This can be found by running 'SpeciesFactorial.R' but requires a specific commit of Biomass_core
PSPmeasure	merged PSP and TSP individual tree measurements. Must include the following columns: MeasureID, OrigPlotID1, MeasureYear, TreeNumber, Species, DBH and newSpeciesName the latter corresponding to species names in 'LandR::sppEquivalencies_CA$PSP'. Defaults to randomized PSP data stripped of real plotIDs
PSPplot	merged PSP and TSP plot data. Defaults to randomized PSP data stripped of real plotIDs. Must contain fields 'MeasureID', 'MeasureYear', 'OrigPlotID1', and 'baseSA' the latter being stand age at year of first measurement
PSPgis	Plot location sf object. Defaults to PSP data stripped of real plotIDs/location. Must include field 'OrigPlotID1' for joining to PSPplot object
species	a table of invariant species traits with the following trait colums: 'species', 'Area', 'longevity', 'sexualmature', 'shadetolerance', 'firetolerance', 'seeddistance_eff', 'seeddistance_max', 'resproutprob', 'mortalityshape', 'growthcurve', 'resproutage_min', 'resproutage_max', 'postfireregen', 'wooddecayrate', 'leaflongevity' 'leafLignin', 'hardsoft'. Only 'growthcurve' and 'mortalityshape' are used in this module. Default is from Dominic Cyr and Yan Boulanger's project
speciesEcoregion	table of spatially-varying species traits ('maxB', 'maxANPP', 'establishprob'), defined by species and 'ecoregionGroup') Defaults to a dummy table based on dummy data os biomass, age, ecoregion and land cover class
sppEquiv	table of species equivalencies. See '?LandR::sppEquivalencies_CA'.

Table 6.2: List of *Biomass_speciesParameters* parameters and their description.

paramName	paramDesc
.plotInitialTime	This describes the simulation time at which the first plot event should occur
.plotInterval	This describes the simulation time interval between plot events
.saveInitialTime	This describes the simulation time at which the first save event should occur
.saveInterval	This describes the simulation time interval between save events
.useCache	Should this entire module be run with caching activated? This is generally intended for data-type modules, where stochasticity and time are not relevant
biomassModel	The model used to calculate biomass from DBH. Can be either 'Lambert2005' or 'Ung2008'
constrainGrowthCurve	upper and lower bounds on range of potential growth curves when fitting traits. This module accepts a list of vectors, with names equal to 'sppEquivCol', so that traits are customizable
constrainMortalityShape	upper and lower bounds on mortality shape when fitting traits. low mortality curve needs to excessive cohorts with very little biomass as longevity is approached, adding computation strain. alternatively accepts a list of vectors, with names equal to 'sppEquivCol'
constrainMaxANPP	upper and lower bounds on 'maxANPP' when fitting traits. cohorts are initiated with 'B = maxANPP' which may be unreasonably high if 'maxANPP' is also high. Both 'maxANPP' and growthcurve params control when 'maxB' is reached. High 'maxANPP' results in earlier peaks alternatively accepts a list of vectors, with names equal to 'sppEquivCol'
GAMMiterations	number of iterations for GAMMs. This module accepts a list of vectors, with names equal to 'sppEquivCol', so that GAMMS are customizable
GAMMknots	the number of knots to use in the GAMM. Either 3 or 4 is recommended. This module accepts a list of vectors, with names equal to 'sppEquivCol', so that GAMMS are customizable

Table 6.3: List of *Biomass_speciesParameters* output objects and their description.

objectName	desc
species	a table that has species traits such as longevity...
speciesEcoregion	table of spatially-varying species traits ('maxB', 'maxANPP', 'establishprob'), defined by species and 'ecoregionGroup') Defaults to a dummy table based on dummy data os biomass, age, ecoregion and land cover class
speciesGAMMs	a list of mixed-effect general additive models (gamm) for each tree species modeling biomass as a function of age

6.1.5   Links to other modules

Intended to be used with another data module, like *Biomass_borealDataPrep*, which will prepare all other traits and parameters for *Biomass_core*. You can see all *potential* module linkages within the LandR ecosystem here. Select *Biomass_speciesParameters* from the drop-down menu to see linkages.

6.1.6   Getting help

- https://github.com/PredictiveEcology/Biomass_speciesParameters/issues

6.2   Module manual

6.2.1   Detailed description

Tree cohort growth and mortality in *Biomass_core* are essentially determined by five parameters: `growthcurve`, `mortalityshape`, maximum biomass (`maxB`), maximum aboveground net primary productivity (`maxANPP`) and `longevity`.

The `growthcurve` and `mortalityshape` parameters (called 'growth curve' and 'mortality shape' in LANDIS-II Biomass Succession Extension v3.2, the base model for *Biomass_core*) strongly modulate the shape of species growth curves and so it is important they they are calibrated to the study area in question.

Also, the growth and mortality equations used in *Biomass_core* are non-linear and their resulting actual biomass accumulation curve is an emergent phenomenon due to competition effects. This means that the ideal trait/parameter values cannot be estimated on a single species basis, as their resulting dynamics will be very different under in a multi-species context.

The *Biomass_speciesParameters* module attempts to address these issues (at least partially) using a "curve-matching" approach. It compares a GAMM fitted to permanent sample plot (PSP) data to a large collection of theoretical species curves, each representing a different set of growth and mortality parameters (`growthcurve` and `mortalityshape`, respectively).

The module generally follows other data modules, like *Biomass_boreaDataPrep*, as it also attempts to calibrate previously estimated spatially varying species traits (`maxB` and `maxANPP`).

6.2.1.1   Permanent sample plot data

Biomass_speciesParameters can use all the PSP data available (note that it may span several thousands of kilometres), or select the data based on a shapefile (`studyAreaANPP`; see Input objects).

By default, the PSP data are obtained from the National Forest Inventory (NFI), the Alberta Ministry of Agriculture, the Saskatchewan Ministry of the Environment, and the British Columbian Ministry of Forests, Lands, Natural Resource Operations, and Rural Development. These data were previously treated for errors and standardized into a single dataset with the exact location and identifying attributes anonymized.

The data include individual species and diameter at breast height (DBH) measurements for each tree in a plot, as well as stand age. As part of the standardization process, dead trees were removed from the dataset. Tree biomass was then derived from DBH using the model by Lambert, Ung, and Raulier [8] and summed by plot. Lastly, the proportional biomass of each species was calculated for every individual measurement of each plot.

Note that the model used to calculate biomass can be changed to Ung, Bernier, and Guo [20] via the `P(sim)$biomassModel` module parameter.

6.2.1.2   Simulated species data

The default dataset of theoretical species curves (biomass accumulation curves, to be more precise) was generated by running a simplified *Biomass_core* simulation with no reproduction, competition, disturbance, dispersal effects, on a single cohort and a single pixel. Each simulation varied in the combination of species trait values that influence growth and mortality dynamics, namely: `growthcurve`, `mortalityshape`, `longevity` and `mANPPproportion`, the ratio

of maximum aboveground net primary productivity (`maxANPP`) to maximum biomass (`maxBiomass`). `growthcurve` values varied from 0 to 1, in increments of 0.1; `mortalityshape` varied from 5 to 25, in increments of 1; `longevity` varied from 150 to 700 in increments of 25; `mANPPproportion` varied from 0.25 to 10 in increments of 0.25; `maxBiomass` was kept constant at 5000.

This resulted in over 200,000 combinations of parameter values.

Results from these simulations were compiled into a table that is accessed by *Biomass_speciesParameters*, so that the module can be run without needing to re-simulate the theoretical curves.

6.2.1.3 Parameter estimation/calibration

Biomass_speciesParameters calibrates `growthcurve`, `mortalityshape`, `maxB` and `maxANPP` by matching the theoretical species curves against observed species growth curves from permanent sample plot (PSP) data. It does not attempt to calibrate `longevity`, as we feel this parameter should be obtained from published sources. `maxB` and `maxANPP` were calibrated using the parameter `mANPPproportion`, as it reflects the relationship between them.

Before calculating the *observed* species growth curves, the module subsets the PSP data to stand ages below the 95th percent quantile for all species (this can be changed via the `P(sim)$quantileAgeSubset` module parameter), as records for larger age classes were limited and constituted statistical outliers. In some species, changing the quantile value may improve results, however. Two examples are *Pinus banksiana* and *Populus sp*, for which using the 99th percent quantile improved the models, because these are short-lived species for which data at advanced ages is scarce.

The module attempts to use "pure" stands to estimate parameters, while balancing sample size (see biomass weighting below). Hence, for a given species, it only includes plots where the species' relative biomass is at least 50% (e.g. for *Populus tremuloides*, plots are only included if 50% of the stand biomass is composed of *P. tremuloides*).

In addition, 50 points are added at the origin (age $= 0$ and biomass $= 0$) to force the intercept to be essentially 0 age and 0 biomass.

Observed growth curves for each species are then modelled using a generalized additive mixed models (GAMMs) that relate species biomass (B) with stand age (*standAge*), accounting for the random effects of the measurement year (*measureYear*) and plot (*plotID*) on the intercept:

$$B \sim f_1(standAge) + (\sim 1|measureYear + plotID) \qquad (6.1)$$

where f_1 denotes the smoother function. To avoid overfitting, the module constrains the smoother on stand age to a maximum smoothing degree of 3 (i.e. 3

knots and a polynomial degree of 2) and a point constraint at 0 that attempts to force the intercept to 0, by default. The smoother degree constraint, however, can be changed via the `P(sim)$GAMMknots` module parameter.

In addition, B is weighted with respect to species dominance. This consisted in 1) calculating the average biomass of each dominant species (i.e. relative biomass in a plot > 0.5; $domSpeciesB_1$), in each plot and measurement year, and 2) dividing the species average biomass by the average biomass across all n dominant species ($allDomSpeciesB$):

$$\frac{\overline{domSpeciesB_1}}{\overline{allDomSpeciesB}} \tag{6.2}$$

For the added 0 age and 0 biomass data the module uses weights equal to 1.

It is possible that some selected species do not have enough data to allow for model convergence. In this case, *Biomass_speciesParameters* skips parameter calibration, and parameter values remain unchanged.

After each species GAMM is calculated, *Biomass_speciesParameters* compares it to the theoretical curves, and picks the best one based on maximum likelihood. This best theoretical curve will be associated with a given combination of `growthcurve`, `mortalityshape` and `maxANPPproportion` values, which are then used directly as the calibrated values, in case of `growthcurve` and `mortalityshape`, or to calibrate `maxANPP` in the case of `maxANPPproportion` (see below).

The user has the option to constrain the values of the `growthcurve` and `mortalityshape` parameters. By default, `growthcurve` is forced to 0.5, `mortalityshape` is allowed to vary between 15 and 25, and `mANPPproportion` between 2.0 and 5.0 (see module parameters `P(sim)$constrainGrowthCurve`, `P(sim)constrainMortalityShape` and `P(sim)constrainMaxANPP`). These boundary values were based on preliminary runs and analyses using the default data and may not apply to other datasets, or to different spatial subsets of the default data.

If boundary values are used, *Biomass_speciesParameters* subsets the theoretical species growth curves to those with trait values within the selected boundaries. Similarly, the module selects only the theoretical curves with longevity values matching those used in the supplied traits table (`species` input object; see Input objects)

Since simulated growth curves never achieve the maximum biomass parameter (the `maxBiomass` parameter set to 5000 for all simulations of theoretical species curves, or the `maxB` parameter in *Biomass_core* simulations), it acts as an asymptotic limit that reflects the potential maximum biomass for a species in an ecolocation (ecological zone and land cover combination), *Biomass_speciesParameters* uses uses the relationship between the achieved

maximum biomass to the potential maximum biomass (`maxBiomass`), to rescale the `maxB` values estimated from data using an upstream module (e.g. *Biomass_borealDataPrep*). This way, species dynamics simulated in *Biomass_core* are able to achieve the maximum observed biomasses (used to *initially* estimate `maxB`).

Finally, the module calibrates `maxANPP` using the `mANPPproportion` value from the best matching theoretical growth curve as:

$$maxB \times \frac{mANPPproportion}{100} \qquad (6.3)$$

where `maxB` is already the calibrated version. In cases where there are not sufficient PSP data to fit the GAMMs and perform the calibration, `mANPPproportion` defaults to 3.33, which corresponds to the value used in LANDIS-II applications in Canada's boreal forests. As already stated above, the final `maxANPP` value is constrained between 2.0 and 5.0 by default.

6.2.2 Initialization, inputs and parameters

Biomass_speciesParameters initializes itself and prepares all inputs provided there is an active internet connection and the user has access to the data (and a Google Account to do so)

6.2.2.1 Input objects

Table 6.4 shows the full list of input objects used by the module.

Of these inputs, the following are particularly important and deserve special attention:

- **Spatial layers**

    - `studyAreaANPP` – shapefile. A `SpatialPolygonsDataFrame` with a single polygon determining the where the PSP should be subset to simulation will take place. This input object **must be supplied by the user**.

- **Tables**

    - `factorialSpeciesTable` and `reducedFactorialCohortData` – a tables of species trait combinations and the theoretical species grwoth curve data (respectively)
    - `PSPmeasure`, `PSPplot` and `PSPgis` – tree measurement, biomass growth and geographical data of the PSP datasets used to buildi observed species growth curves.

Table 6.4: List of *Biomass_speciesParameters* input objects and their description.

objectName	objectClass	desc	sourceURL
factorialSpeciesTable	data.table	table with species traits for matching to 'reducedFactorialCohortData'	https://drive.google.com/open?id=1q0ou0...
reducedFactorialCohortData	data.table	results of factorial species trait simulation. This can be found by running 'SpeciesFactorial.R' but requires a specific commit of Biomass_core	https://drive.google.com/open?id=1h8StXI...
PSPmeasure	data.table	merged PSP and TSP individual tree measurements. Must include the following columns: MeasureID, OrigPlotID1, MeasureYear, TreeNumber, Species, DBH and newSpeciesName the latter corresponding to species names in 'LandR::sppEquivalencies_CA$PSP'. Defaults to randomized PSP data stripped of real plotIDs	https://drive.google.com/file/d/1LmOaEtC...
PSPplot	data.table	merged PSP and TSP plot data. Defaults to randomized PSP data stripped of real plotIDs. Must contain fields 'MeasureID', 'MeasureYear', 'OrigPlotID1', and 'baseSA' the latter being stand age at year of first measurement	https://drive.google.com/file/d/1LmOaEtC...

- **species** – a table of invariant species traits that may have been produced by another module. It **must** contain the columns 'species', 'growthcurve' and 'mortality shape', whose values will be calibrated.
- **speciesEcoregion** – table of spatially-varying species traits that may have been produced by another module. It **must** contain the columns 'speciesCode', 'maxB' and 'maxANPP' and 'ecoregion-Group' (the ecolocation ID). 'maxB' and 'maxANPP' values will be calibrated by species.

6.2.2.2 Parameters

Table 6.5 lists all parameters used in *Biomass_speciesParameters* and their detailed information.

Of these parameters, the following are particularly important:

- **Calibration parameters**

  - **GAMMiterations** and **GAMMknots** – control the number of iterattions and smoother degree used to fit the GAMMs, respectively.
  - **constrainGrowthCurve**, **constrainMortalityShape** and **constrainMaxANPP** – determine the upper and lower boundaries of the calibrated values of **growthcurve**, **mortalityshape** and **maxANPP**, respectively.

- **Data processing**

  - **minimumPlotsPerGamm** – define a minimum number of PSP plots needed to fit the GAMMs.
  - **PSPperiod** – PSP data period to use.
  - **quantileAgeSubset** – upper quantile age value used to subset PSP data.

6.2.2.3 Outputs

- **Tables**

  - **species** and **speciesEcoregion** – tables with calibrated trait values.
  - **speciesGAMMs** – the fitted GAMM model objects for each species.

6.2.3 Simulation flow

The general flow of *Biomass_speciesParameters* processes is:

Table 6.5: List of *Biomass_speciesParameters* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
.plotInitialTime	numeric	NA	NA	NA	This describes the simulation time at which the first plot event should occur
.plotInterval	numeric	NA	NA	NA	This describes the simulation time interval between plot events
.saveInitialTime	numeric	NA	NA	NA	This describes the simulation time at which the first save event should occur
.saveInterval	numeric	NA	NA	NA	This describes the simulation time interval between save events
.useCache	logical	FALSE	NA	NA	Should this entire module be run with caching activated? This is generally intended for data-type modules, where stochasticity and time are not

1. Preparation of all necessary data and input objects that do not require parameter fitting (e.g., the theoretical species growth curve data);

2. Sub-setting PSP data and calculating the observed species growth curves using GAMMs;

3. Finding the theoretical species growth curve that best matches the observed curve, for each species, within curves produced with similar longevity values to those in the species traits table (`species`) and with `growthcurve` and `mortalityshape` values within the chosen boundaries (`P(sim)$constrainGrowthCurve`, `P(sim)$constrainMortalityShape`);

4. Calibrating `maxB` and `maxANPP`

5. Adjusting `maxANPP` to match the chosen boundaries (`P(sim)$constrainMaxANPP`)

6.3 Usage example

This module can be run stand-alone, but it won't do much more than calibrate species trait values based on dummy input trait values. We provide an example of this below, since it may be of value to run the module by itself to become acquainted with the calibration process and explore the fitted GAMMs. However, we remind that to run this example you will need a Google Account, and to be granted access to the data.

A realistic usage example of this module and a few others can be found in this repository and in Barros et al. [2].

6.3.1 Load `SpaDES` and other packages.

```r
if (!require(Require)) {
  install.packages("Require")
  library(Require)
}

Require(c("PredictiveEcology/SpaDES.install",
          "SpaDES", "PredictiveEcology/SpaDES.core@development"),
        install_githubArgs = list(dependencies = TRUE))
```

6.3.2 Get module, necessary packages and set up folder directories

```
tempDir <- tempdir()
spadesModulesDirectory <- file.path(tempDir, "modules")

paths <- list(inputPath = normPath(file.path(tempDir, "inputs")),
              cachePath = normPath(file.path(tempDir, "cache")),
              modulePath = normPath(spadesModulesDirectory),
              outputPath = normPath(file.path(tempDir, "outputs")))

getModule("PredictiveEcology/Biomass_speciesParameters@79896a4e3b785e34e5f509798ab6c22

## make sure all necessary packages are installed:
makeSureAllPackagesInstalled(spadesModulesDirectory)
```

6.3.3   Setup simulation

```
library(SpaDES)

times <- list(start = 0, end = 1)

modules <- list("Biomass_speciesParameters")
inputs <- list()
outputs <- list()
objects <- list()

parameters <- list(Biomass_speciesParameters =
                     list(GAMMiterations = 2,
                          GAMMknots = list(
                            "Abie_bal" = 3,
                            "Abie_las" = 3,
                            "Betu_pap" = 3,
                            "Lari_lar" = 4,
                            "Pice_eng" = 4,
                            "Pice_gla" = 3,
                            "Pice_mar" = 4,
                            "Pinu_ban" = 3,
                            "Pinu_con" = 4,
                            "Popu_tre" = 4,
                            "Pseu_men" = 3),
                          minimumPlotsPerGamm = 40,
                          constrainMortalityShape = list(
                            "Abie_bal" = c(15,25),
                            "Abie_las" = c(15,25),
```

```
                                  "Betu_pap" = c(15,20),
                                  "Lari_lar" = c(20,25),
                                  "Pice_eng" = c(20,25),
                                  "Pice_gla" = c(20,25),
                                  "Pice_mar" = c(15,25),
                                  "Pinu_ban" = c(15,25),
                                  "Pinu_con" = c(15,25),
                                  "Popu_tre" = c(20,25),
                                  "Pseu_men" = c(20,25)
                                ),
                                constrainGrowthCurve = list(
                                  "Abie_bal" = c(0, 1),
                                  "Abie_las" = c(0, 1),
                                  "Betu_pap" = c(0, 1),
                                  "Lari_lar" = c(0, 1),
                                  "Pice_eng" = c(0, 1),
                                  "Pice_gla" = c(0, 1),
                                  "Pice_mar" = c(0, 1),
                                  "Pinu_ban" = c(0, 1),
                                  "Pinu_con" = c(0, 1),
                                  "Popu_tre" = c(0, 1),
                                  "Pseu_men" = c(0, 1)
                                ),
                                quantileAgeSubset = list(
                                  "Abie_bal" = 95,
                                  "Abie_las" = 95,
                                  "Betu_pap" = 95,
                                  "Lari_lar" = 95,
                                  "Pice_eng" = 95,
                                  "Pice_gla" = 95,
                                  "Pice_mar" = 95,
                                  "Pinu_ban" = 95,
                                  "Pinu_con" = 99,
                                  "Popu_tre" = 99,
                                  "Pseu_men" = 99
                                )
                        ))


mySim <- simInitAndSpades(times = times,
                          params = parameters,
                          modules = modules,
                          paths = paths,
                          objects = objects)
```

```
## to inspect the fitted GAMM models:
mySim$speciesGAMMs$Pice_mar
```

6.4 References

Chapter 7

LandR Validation Modules

LandR 'validation modules' differ from 'data modules' in that their objective is not to obtain input data and estimate parameters to run a simulation, but rather to confront simulation outputs against observed data – even if these modules can potentially obtain and pre-process the validation data. At the moment, only one validation module is available *Biomass_validationKNN*, but we expect an increment in the number of validation modules as LandR usage expands.

Chapter 8

LandR *Biomass_validationKNN* Module

This documentation is work in progress. Potential discrepancies and omissions may exist for the time being. If you find any, do contact us using the link above^^

8.0.0.1 Authors:

Ceres Barros cbarros@mail.ubc.ca [aut, cre]

8.1 Module Overview

8.1.1 Module summary

An approach to validating outputs from LandR Biomass - notably the *Biomass_core* vegetation simulation module - using publicly available data for Canadian forests. This module produces both visual and statistical validation of *Biomass_core* outputs that are related to species abundance and presence/absence in the landscape. To do so, it downloads and prepares all necessary data (observed and simulated), calculates validation statistics and produces/saves validation plots.

8.1.2   Module inputs and parameters at a glance

Biomass_validationKNN requires access to outputs of simulations from *Biomass_core*, and internet access to retrieve the observed kNN datasets used for validation. Raw data layers downloaded by the module are saved in `dataPath(sim)`, which can be controlled via `options(reproducible.destinationPath = ...)`.

We advise future users to run *Biomass_validationKNN* with defaults and inspect what the objects are like before supplying their own data, or alternative data URLs. We expect the number of validation modules to increase as other validation approaches are developed based on project needs.

The module is able to compile all simulation output data provided that the user supplies the object names and their file paths via the `simulationOutputs` input object. Alternatively, the user may pass the pre-compiled outputs (namely the `cohortData` and `pixelGroupMap` objects) via the `allCohortData` and `pixelGroupMapStk` input objects. See Input objects for more detail.

Key parameters are those defining simulation years and replicates, (`validationYears`, `validationReps`) and plot control (`.plots`). Here's the full list of parameters:

Below are the full lists of input objects (Table 8.1.2) and parameters (Table 8.1.2) that *Biomass_validationKNN* expects. The only input that **must** be provided (i.e., *Biomass_validationKNN* does not have a default for) is `studyArea`. All other input objects and parameters have internal defaults (see Tables 8.2.3.1 and 8.2.3.2. Objects suffixed with `*Start` correspond to the same objects in the simulation without this suffix (e.g. `rawBiomassMapStart` is `rawBiomassMap` in the simulation), whereas other objects like `studyArea` and `rasterToMatch` have the same names in the simulation and should be exactly the same object.

List of *Biomass_validationKNN* input objects and their description.

objectName

desc

allCohortData

All `cohortData` tables saved during the simulation, particularly for the validation years. If not supplied, the module will attempt to retrieve them using the 'simulationOutputs' table

biomassMap

total biomass raster layer in study area (in g/m^2), filtered for pixels covered by `cohortData`. Only used to calculate total no. of pixels being simulated If not supplied, will default to `rawBiomassMapStart`

firePerimeters

A map of fire perimeters in the study area that can be used to exclude pixels that have been burnt during the validation period. If burnt pixels are not to be excluded Provide an empty `sf` object with the same properties as the default. Defaults to the latest Canadian Wildland Fire Information System National Burned Area Composite, subset to fires occuring up to last validation year (inclusively). Source URL determined by `fireURL`

fireURL

A URL to a fire database, such as the Canadian National Fire Database, that is a zipped shapefile with fire polygons, an attribute (i.e., a column) named 'Year'. If supplied (omitted with NULL or NA), this will be used to 'update' age pixels on `standAgeMap` with 'time since fire' as derived from this fire polygons map

pixelGroupMapStk

A stack of `pixelGroupMap`s saved during the simulation, particularly for the validation years. If not supplied, the module will attempt to make it using the 'simulationOutputs' table

rawBiomassMapStart

observed total biomass raster layer in study area at the first year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in ton/ha). See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.

rawBiomassMapEnd

observed total biomass raster layer in study area at the last year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2011 (in ton/ha) See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990

rasterToMatch

A raster of the `studyArea` in the same resolution and projection as `rawBiomassMapStart`. This is the scale used for all outputs for use in the simulation.

rstLCChange

A mask-type map of land cover changes in the study area that can be used to exclude pixels that have been disturbed during the validation period. If disturbed pixels are not to be excluded Provide an empty sf object with the same properties as the default. Defaults to Canada's forest change map between 1985-2011 (CFS), filtered for years 2001-2011 (inclusively) and all disturbances collapsed (map only has values of 1 and NA). See `P(sim)$LCChangeYr` parameter to change the period of disturbances, and https://opendata.nfis.org/mapserver/nfis-change_eng.html for more information.

rstLCChangeYr

An OPTIONAL map of land cover change years in the study area used to exclude pixels that have been disturbed during the validation period. It defaults to Canada's forest change year national map between 1985-2011 (CFS). If `P(sim)$LCChangeYr` is not NULL, this layer is used to filted disturbed pixels that fall within the years specified by `P(sim)$LCChangeYr`. If `P(sim)$LCChangeYr` is NULL this layer is not used. See https://opendata.nfis.org/mapserver/nfis-change_eng.html for more information.

simulationOutputs

An OPTIONAL table listing simulation outputs (as passed to `spades()`, or experiment) that will be used to make`allCohortData`,pixelGroupMapStk,
if these are not provided. </td>   </tr>   <tr>   <td style="text-align:left;">
speciesLayersStart </td>   <td style="text-align:left;"> observed
cover percentage raster layers by species in Canada species
map, at the first year of the validation period. Defaults to the
Canadian Forestry Service, National Forest Inventory, kNN-derived
species cover maps from 2001, using a cover threshold of 10% -
see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990
for metadata </td>   </tr>   <tr>   <td style="text-align:left;">
speciesLayersEnd </td>   <td style="text-align:left;"> observed
percent cover raster layers by species in Canada used for validation
at the last year of the validation period. Defaults to the Canadian
Forestry Service, National Forest Inventory, kNN-derived species
cover maps from 2011 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-8
for metadata </td>   </tr>   <tr>   <td style="text-align:left;">
sppColorVect </td>   <td style="text-align:left;"> A named
vector of colors to use for plotting. The names must be in sim$sppEquiv[[sim$sppEquivC
and should also contain a color for 'Mixed' </td>   </tr>   <tr>
<td style="text-align:left;"> sppEquiv </td>   <td style="text-align:left;">
table of species equivalencies. See`LandR::sppEquivalencies_CA`.

standAgeMapStart

observed stand age map in study area, at the first year of the validation period Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived biomass map from 2001 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

standAgeMapEnd

observed stand age raster layer in study area, at the last year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived stand age map from 2011. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990

studyArea

Polygon to use as the study area. Must be provided by the user

List of *Biomass_validationKNN* parameters and their description.

paramName

paramDesc

coverThresh

The minimum % cover a species needs to have (per pixel) in the study area to be considered present. Should be the same as the one used to obtain the species cover layers for simulation set up.

deciduousCoverDiscount

This was estimated with data from NWT on March 18, 2020 and may or may not be universal. Should be the same as the one used when preparing `cohortData` in the simulation set up.

LCChangeYr

OPTIONAL. An integer or vector of integers of the validation period years, defining which years of land-cover changes (i.e. disturbances) should be excluded. `NULL` by default, which presumes no subsetting based on years is done internally (either the user supplies a pre-filtered `rstLCChange`, or no filtering is desired). If not `NULL` `rstLCChangeYr` is used to filter disturbed pixels within the specified years. See https://opendata.nfis.org/mapserver/nfis-change_eng.html for more information.

minCoverThreshold

Cover that is equal to or below this number will be omitted from the dataset Should be the same as the one used when preparing `cohortData` in the simulation set up.

obsDeltaAgeB

When TRUE, the observed changes in biomass and age (deltaB, deltaAge) between the two validation years will be plotted as maps and scatterplots

pixelGroupBiomassClass

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 Should be the same as the one used when preparing `cohortData` in the simulation set up.

sppEquivCol

The column in `sim$sppEquiv` data.table to use as a naming convention

validationReps

The simulation repetitions for the validation. Defaults to 1:10. Set to NA if not using repetitions (i.e. only one run)

validationYears

The simulation years for the validation. Defaults to 2001 and 2011. Must select two years

.plotInitialTime

If NA plotting is off completely (this includes saving).

.plots

Passed to `types` in Plots (see ?Plots). There are a few plots that are made within this module, if set. Note that plots (or their data) are saved in file.path(outputPath(sim), 'figures'). If `NA`, plotting is off completely (this includes plot saving).

.saveInitialTime

This describes the simulation time at which the first save event should occur

.saveInterval

This describes the simulation time interval between save events

.studyAreaName

Human-readable name for the study area used. If `NA`, a hash of `studyArea` will be used.

.useCache

Controls cache; caches the init event by default

8.1.3 Events

The following events take place during a *Biomass_validationKNN* run. Note that this module only runs once (in one "time step").

- Module initiation (`init` event) - prepares both the observed and simulated data to be compared

- Calculation of summary variables for validation (`calculateValidVars` event)

- Calculation of validation statistics (`validationStats` event)

- Optional diagnostic plots of biomass and age changes (ΔB, ΔAge) in the observed data (`obsDeltaMaps` event).

- Plotting events:

  - Plots of landscape-wide (`landscapeWidePlots` event), pixel-level (`pixelLevelPlots` events) comparisons of all summary variables

  - Plots of biomass and age changes (ΔB, ΔAge) in observed and simulated data, with respect to the first year. (`deltaBComparisons` event)

8.1.4 Module outputs

The module produces the following outputs (Table 8.1.4):

List of *Biomass_validationKNN* output objects and their description.

objectName

desc

logLikelihood

A table of negative sum log-likelihood values calculated for different variables and averaged across repetitions. At the moment, log-likelihood values are calculated for biomass (landscape- and pixel-level), species presences and dominance (lanscape-level) and deltaB (landscape- and pixel-level. For biomass and count data (presences/dominance, we assume an underlying multinomial distribution, and for deltaB a multivariate Gaussian distribution - note that the later is still under development.

landscapeMAD

Mean absolute deviance values calculated on landscape-level relative abundances, species presences and dominance, and deltaB, per repetition and year (except for deltaB, which is integrated across years)

landscapeVars

A table containing observed and simulated landscape-averaged variables used for validation (by year and repetition, 'rep', in the case of simulated data), namely: species relative abundances ('relAbund'), species presenses ('count'), species dominance (as in no. pixels where a given species, has higher 'relAbund'; 'countDom') and species changes in biomass, as 2011 minus 2001 ('deltaB'). Observed data rows are labelled as 'observed' in 'dataType' column. In species dominance, pixels with $>= 2$ species with max(B) and pixels with no B are classified as 'Mixed' and 'No veg.', respectively in the 'speciesCode' column - note that this is 'vegType' column in `pixelCohortData`.

pixelCohortData

A table containing observed and simulated pixel-level data (by year and repetition, 'rep', in the case of simulated data) on species biomass (summed across cohorts, 'B'), total pixel biomass ('pixelB'), average biomass-weighted pixel age ('pixelAge'), species relative abundance (calculated as B/pixelB, 'relativeAbund'), species dominance (the species with max(B), 'vegType'), and lanscape-wide biomass ('landscapeB'). Observed data columns are suffixed with 'Obsrvd'. In species dominance, pixels with $>= 2$ species with max(B) (i.e. 'noDoms' $>= 2$) are classified as 'Mixed'.

pixelMAD

Mean absolute deviance values calculated on pixel-level relative abundances and deltaB, per repetition and year (except for deltaB, which is integrated across years)

pixelVars

The same as `landscapeVars`, but variables are calculated at the pixel-level

rstDisturbedPix

Raster of pixel IDs (as a mask) that have been disturbed by fire or suffered land-cover changes during the validation period. These pixels are excluded form the validation.

rawBiomassMapStart

observed total biomass raster layer in study area at the first year of the validation period. Filtered to exclude pixels that were disturbed during the validation period

rawBiomassMapEnd

observed total biomass raster layer in study area at the last year of the validation period. Filtered to exclude pixels that were disturbed during the validation period

speciesLayersStart

observed percent cover raster layers by species in Canada at the first year of the validation period. Filtered to exclude pixels that were disturbed during the validation period

speciesLayersEnd

observed percent cover raster layers by species in Canada at the last year of the validation period. Filtered to exclude pixels that were disturbed during the validation period

standAgeMapStart

observed stand age map in study area, at the first year of the validation period Filtered to exclude pixels that were disturbed during the validation period

standAgeMapEnd

observed stand age map in study area, at the last year of the validation period Filtered to exclude pixels that were disturbed during the validation period

and saves the validation figures in the output path defined in `getPaths(sim)$outputPath` (which was defined as `paths$outputPath`). Tables are not saved unless specified via `spades(outputs = data.frame(...), ...)`. If not saving objects to disk (such as tables), these can be looked at using, e.g., `mySimValidation$logLikelihood`.

8.1.5 Links to other modules

Intended to be used with *Biomass_core* and any other modules that link to it and affect cohort biomass (e.g., via `cohortData` table). You can see all *potential* module linkages within the LandR ecosystem here. Select *Biomass_validationKNN* from the drop-down menu to see linkages.

8.1.6 Getting help

- https://github.com/PredictiveEcology/Biomass_validationKNN/issues

8.2 Module manual

8.2.1 Detailed description

This module compares simulated outputs of two years (across replicates), with corresponding years of observed data. It was designed to compare the observed data for years 2001 (start point for the simulation) and 2011 (i.e., after 10 years of simulation) of the kNN forest layers of the Canadian National Forest Inventory – these are currently the only available FAIR datasets [*sensu* 19]) on stand biomass and species % cover changes across Canada. However, the user can supply other sources of observed data, as long as they have an identical format.

The validation is done both visually (using barplots and boxplots) and using two statistics: mean absolute deviation of simulated biomass (per species) and the sum of negative log-likelihoods (SNLL) of predictions with respect to observed data for species biomass, species presences/absences and changes in biomass (ΔB) - the later is still under development.

This module assumes that the simulation data preparation was carried out by *Biomass_borealDataPrep*, and so, to ensure that the comparison and the simulated datasets are built with the same assumptions, the data treatment steps in *Biomass_borealDataPrep* are repeated here. The module may also excludes disturbed pixels coded in `rstLCCChange` and fire perimeter data (`firePerimeters`). If this is not intended, pass a `rstLCCChange` with `NA`'s only and/or an empty `firePerimeters sf` object. *Biomass_validationKNN* then compares simulated species biomass, presences, dominance, and changes in biomass against observed data available for the starting conditions (2011 by default) and a second time point (e.g. 2011, or after 10 years of simulation). To do so, for each year and replicate, and for both the simulated and observed data, the module calculates:

- species relative abundances at the pixel- and landscape-level (across all pixels)

- species presences and dominance at the landscape level

- changes in species biomass (ΔB) at the pixel- and landscape-level for both the simulated and observed data. Biomass units respect those used in *Biomass_core* (g/m^2).

Pixel-level relative abundances are calculated as the species biomass (summed across cohorts) divided by the total pixel biomass (summed across cohorts and species), while landscape-wide relative abundances are calculated as the sum of a species biomass across all pixels divided by the sum of total biomass across all pixels. Species presences are calculated as the number of pixels where a given species is present, and species dominance is calculated as the number of pixels where a species has the highest relative biomass in a given pixel. Pixels where two or more species share the highest biomass value are classified as 'mixed forest', and pixels without any biomass are classified as 'no veg.'. Finally, (ΔB) is calculated per species as the final biomass (e.g., year 2011) minus the initial biomass (e.g., year 2001), either at the pixel- or landscape-level. All calculations were done per replicate.

8.2.2   Validation approaches

8.2.2.1   Visual validation

The module plots the above metrics as barplots showing landscape-level values (averaged across replicates for the simulated data) or boxplots showing pixel-level values. Plotting can be live and/or in the form of exported images (or both turned off completely).

8.2.2.2   Mean absolute deviation

Mean absolute deviance (MAD) values are calculated on landscape- and pixel-level species relative abundances and ΔB, and landscape-level species presences and dominance, per replicate and year (except for ΔB, which is integrated across years). Output tables with MAD values are exported as `landscapeMAD` and `pixelMAD`, and the module also produces visual inspection of these values as dot-and-whisker plots.

8.2.2.3   Sum of negative log-likelihood (SNLL)

To provide a measure of overall goodness of fit of the simulation model, given a given set of starting conditions and simulation mechanisms (i.e., the combination of inputs to *Biomass_core* but also other modules that may be associated in affecting vegetation dynamics), *Biomass_validationKNN* estimates sum of negative log-likelihoods (SNLL) of simulated species biomasses, ΔB (both at the

landscape and pixel-level), and species presences (at the landscape-level), with respect to their observed counterparts. More precisely, let ℓ be the log-likelihood function denoting the probability of observing x of X (a random variable following a continuous probability distribution $f(x)$), given a parameter θ:

$$\ell(\theta \mid x) = f(x) \tag{8.1}$$

In our case, θ is equivalent to the model's starting conditions and structure, X is the observed data with x being the simulated values, and $f(x)$ the continuous probability distribution of X. For each variable that we wanted to evaluate and for each simulation replicate, Equation (8.1) is applied to calculate the SNLL estimated for each value of x at the pixel or landscape-level, i:

$$-\sum_{i=1}^{N} \ell(\theta \mid x_i) \tag{8.2}$$

where N is equal to total number of pixels. At the landscape scale $N = 1$.

For species biomass and species presences, we draw the probability of observing x_i (a vector of species biomasses/presences in pixel/landscape i) from a multinomial density distribution ($f(x_i) = \mathsf{Multi}(n_i, \mathrm{p}_i)$), where $n_i = \sum_{j=1}^{K} X_{i,j}$ (X being the observed values of biomass of $j = 1, ..., K$ species in a pixel/landscape i) and p_i is the vector of simulated values $x_{i,j}$.

The computation of SNLL for ΔB is still under development. We have implemented the following, approach: For ΔB, we draw the probability of observing $x_{i,j}$ (the simulated ΔB of $j = 1, ..., K$ species in a pixel/landscape i) from a multivariate Gaussian distribution, $f(x_i) = \mathcal{N}(\mu_i, \mathrm{M}_i)$, where μ_i is the vector of observed mean ΔB for each species $j = 1, ..., K$, and M is the observed $K * K$ variance-covariance matrix of species ΔB. Unfortunately this is presenting problems, due to M not being strictly positive definite.

After calculating SNLL across pixels (or for a landscape), values are averaged across replicates for an overall model estimate and exported in the `logLikelihood` table.

We refer to the Wikipedia pages on the multinomial distribution and on the multivariate Gaussian distribution for a good summary of these two distributions and their use in SNLL estimation.

8.2.3 Initialization, inputs and parameters

Biomass_validationKNN initializes itself and prepares all inputs provided that it has access to outputs of simulations from *Biomass_core*, and internet access to retrieve the observed kNN datasets used for validation. Future users should run *Biomass_validationKNN* with defaults and inspect what the objects are

like before supplying their own data, or alternative data URLs. Alternatively, users may develop their own validation modules using *Biomass_validationKNN* as a template.

8.2.3.1   Input objects

Biomass_validationKNN requires the following input data layers: land-cover change (change type and year), fire perimeters, % species cover, stand age and stand biomass. By default, the module will take these from National Forest Inventory kNN layers for years 2001 and 2011. We recommend that the user supplies layers used to initialise the simulation as the starting input layers (2001 if that is the starting point) to guarantee that they match. Table 8.2.3.1 shows the full list of input objects used by the module.

List of *Biomass_validationKNN* input objects and their description.

objectName

objectClass

desc

sourceURL

allCohortData

data.table

All `cohortData` tables saved during the simulation, particularly for the validation years. If not supplied, the module will attempt to retrieve them using the 'simulationOutputs' table

NA

biomassMap

RasterLayer

total biomass raster layer in study area (in g/m^2), filtered for pixels covered by `cohortData`. Only used to calculate total no. of pixels being simulated If not supplied, will default to `rawBiomassMapStart`

NA

firePerimeters

sf

A map of fire perimeters in the study area that can be used to exclude pixels that have been burnt during the validation period. If burnt pixels are not to be excluded Provide an empty `sf` object with the same properties as the default. Defaults to the latest Canadian Wildland Fire Information System National

Burned Area Composite, subset to fires occuring up to last validation year (inclusively). Source URL determined by `fireURL`

NA

fireURL

character

A URL to a fire database, such as the Canadian National Fire Database, that is a zipped shapefile with fire polygons, an attribute (i.e., a column) named 'Year'. If supplied (omitted with NULL or NA), this will be used to 'update' age pixels on `standAgeMap` with 'time since fire' as derived from this fire polygons map

https://cwfis.cfs.nrcan.gc.ca/downloads/nfdb/fire_poly/current_version/NFDB_poly.zip

pixelGroupMapStk

RasterStack

A stack of `pixelGroupMap`s saved during the simulation, particularly for the validation years. If not supplied, the module will attempt to make it using the 'simulationOutputs' table

NA

rawBiomassMapStart

RasterLayer

observed total biomass raster layer in study area at the first year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in ton/ha). See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2001-attributes_attributs-2001/NFI_MODIS250m_2001_kNN_Structure_Biomass_TotalLiveAboveGround_v1.tif

rawBiomassMapEnd

RasterLayer

observed total biomass raster layer in study area at the last year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2011 (in ton/ha) See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2011-attributes_attributs-2011/NFI_MODIS250m_2011_kNN_Structure_Biomass_TotalLiveAboveGround_v1.tif

rasterToMatch

RasterLayer

A raster of the `studyArea` in the same resolution and projection as `rawBiomassMapStart`. This is the scale used for all outputs for use in the simulation.

NA

rstLCChange

RasterLayer

A mask-type map of land cover changes in the study area that can be used to exclude pixels that have been disturbed during the validation period. If disturbed pixels are not to be excluded Provide an empty sf object with the same properties as the default. Defaults to Canada's forest change map between 1985-2011 (CFS), filtered for years 2001-2011 (inclusively) and all disturbances collapsed (map only has values of 1 and NA). See `P(sim)$LCChangeYr` parameter to change the period of disturbances, and https://opendata.nfis.org/mapserver/ nfis-change_eng.html for more information.

https://opendata.nfis.org/downloads/forest_change/C2C_change_type_ 1985_2011.zip

rstLCChangeYr

RasterLayer

An OPTIONAL map of land cover change years in the study area used to exclude pixels that have been disturbed during the validation period. It defaults to Canada's forest change year national map between 1985-2011 (CFS). If `P(sim)$LCChangeYr` is not NULL, this layer is used to filted disturbed pixels that fall within the years specified by `P(sim)$LCChangeYr`. If `P(sim)$LCChangeYr` is `NULL` this layer is not used. See https://opendata.nfis.org/mapserver/nfis-change_eng.html for more information.

https://opendata.nfis.org/downloads/forest_change/C2C_change_year_ 1985_2011.zip

simulationOutputs

data.table

An OPTIONAL table listing simulation outputs (as passed to `spades()`, or `experiment`) that will be used to make`allCohortData`,`pixelGroupMapStk`, if these are not provided. </td>   <td style="text-align:left;"> NA </td>   </tr>   <tr>   <td style="text-align:left;"> speciesLayersStart </td>   <td style="text-align:left;"> RasterStack </td>   <td style="text-align:left;"> observed cover percentage raster layers by species in Canada species map, at the first year of the validation period. Defaults to the Canadian Forestry Service, National

```
Forest Inventory, kNN-derived species cover maps from 2001, using
a cover threshold of 10% - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6ac
for metadata </td>    <td style="text-align:left;"> http://ftp.maps.canada.ca/pub/nrcan_rncan/For
</td>   </tr>   <tr>    <td style="text-align:left;"> speciesLayersEnd
</td>    <td style="text-align:left;"> RasterStack </td>    <td
style="text-align:left;"> observed percent cover raster layers
by species in Canada used for validation at the last year of the
validation period. Defaults to the Canadian Forestry Service,
National Forest Inventory, kNN-derived species cover maps from
2011 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990
for metadata </td>    <td style="text-align:left;"> http://ftp.maps.canada.ca/pub/nrcan_rncan/For
</td>   </tr>   <tr>    <td style="text-align:left;"> sppColorVect
</td>    <td style="text-align:left;"> character </td>    <td
style="text-align:left;"> A named vector of colors to use for
plotting. The names must be in sim$sppEquiv[[sim$sppEquivCol]],
and should also contain a color for 'Mixed' </td>    <td style="text-align:left;">
NA </td>   </tr>   <tr>    <td style="text-align:left;"> sppEquiv
</td>    <td style="text-align:left;"> data.table </td>    <td
style="text-align:left;"> table of species equivalencies. SeeLandR::sppEquivalencies_CA'.
```

NA

standAgeMapStart

RasterLayer

observed stand age map in study area, at the first year of the validation period Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived biomass map from 2001 - see https://open.canada.ca/data/en/dataset/ ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2001-attributes_attributs-2001/NFI_ MODIS250m_2001_kNN_Structure_Stand_Age_v1.tif

standAgeMapEnd

RasterLayer

observed stand age raster layer in study area, at the last year of the validation period. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived stand age map from 2011. See https://open.canada.ca/data/en/ dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990

http://ftp.maps.canada.ca/pub/nrcan_rncan/Forests_Foret/canada-forests-attributes_attributs-forests-canada/2011-attributes_attributs-2011/NFI_ MODIS250m_2011_kNN_Structure_Stand_Age_v1.tif

studyArea

SpatialPolygonsDataFrame

Polygon to use as the study area. Must be provided by the user

NA

Of the inputs in Table 8.2.3.1, the following are particularly important and deserve special attention:

- **Spatial layers**

  - `biomassMap` – a map of simulated stand biomass (in g/m^2) filtered for the pixels where cohort dynamics were simulated. This corresponds to the `sim$biomassMap` object produced by *Biomass_borealDataPrep* or to the `sim$simulatedBiomassMap` produced by *Biomass_core.*

  - `firePerimeters` – a fire perimeters polygon map that should be used to exclude recently burned pixels from the analysis. If this is not desired the user needs to provide an empty `sf` object (e.g., `sf::st_polygon())` .

  - `rawBiomassMapStart` – raw biomass data used to initialize and parametrize `Biomass_core`. By default, the module uses the stand biomass map from KNN for the year 2001. The user must make sure this appropriate for their use case, or else supply the correct raster layer.

  - `rawBiomassMapEnd` – raw biomass data used to validate the model after several simualtion years. By default, the module uses the stand biomass map from KNN for the year 2011, which is compared with the 10th year of a simulation initialised using the KNN 2001 data. The user must make sure this appropriate for their use case, or else supply the correct raster layer.

  - `rstLCChange` – a binary raster layer with disturbed pixels that should be removed from the analyses. Can be combined with `rstLCChangeYr` to filter pixels disturbed in a given time period defined by `P(sim)$LCChangeYr`. Defaults to Canada's forest change national map between 1985-2011 (CFS).

  - `rstLCChangeYr` – a raster layer with year of disturbance. This is an optional layer that can be combined with `rstLCChange` and `P(sim)$LCChangeYr` to filter disturbed pixels by year of disturbance. Not used by default. Defaults to Canada's forest change year national map between 1985-2011 (CFS).

  - `speciesLayersStart` – same as `rawBiomassMapStart`, but with respect to species % cover data.

  - `speciesLayersEnd` – same as `rawBiomassMapEnd`, but with respect to species % cover data.

– `studyArea` – shapefile. A `SpatialPolygonsDataFrame` with a single polygon determining the where the simulation will take place. This is the only input object that **must be supplied by the user**.

- **Simulation-related objects**

  – `allCohortData` – OPTIONAL. A `data.table` containing all `cohortData` objects relevant for the validation (e.g., as many `cohortData` objects as simulation replicates times 2, for the beginning and end year). If not supplied, *Biomass_validationKNN* attempts to produce this object using the `cohortData` object file listed in `simulationOutputs`. Hence, the user must either supply **both** `allCohortData` and `pixelGroupMapStk` **or** `simulationOutputs`.

  – `pixelGroupMapStk` – OPTIONAL. As `allCohortData` but with respect to `pixelGroupMap` objects.

  – `simulationOutputs` – OPTIONAL. A `data.frame` that has the same structure as the `data.frame`'s specifying outputs to be saved in `spades(..., outputs = data.frame(...))`. We advise passing the same `data.frame` that was supplied to `spades` during the simulation call, but filtered by the relevant `cohortData` and `pixelGroupMap` objects and, potentially, with file paths corrected to match the current working directory (see Usage example). Only used if `allCohortData` and `pixelGroupMapStk` are not supplied.

  – `pixelGroupMap` – a raster layer with *pixelGroup* IDs per pixel. Pixels are always grouped based on identical *ecoregionGroup*, *speciesCode*, *age* and *B* composition, even if the user supplies other initial groupings (e.g., this is possible in the *Biomass_borealDataPrep* data module).

8.2.3.2 Parameters

Table 8.2.3.2 lists all parameters used in *Biomass_validationKNN* and their detailed information.

List of *Biomass_validationKNN* parameters and their description.

paramName

paramClass

default

min

max

paramDesc

coverThresh

integer

10

NA

NA

The minimum % cover a species needs to have (per pixel) in the study area to be considered present. Should be the same as the one used to obtain the species cover layers for simulation set up.

deciduousCoverDiscount

numeric

0.8418911

NA

NA

This was estimated with data from NWT on March 18, 2020 and may or may not be universal. Should be the same as the one used when preparing `cohortData` in the simulation set up.

LCChangeYr

integer

1900

NA

OPTIONAL. An integer or vector of integers of the validation period years, defining which years of land-cover changes (i.e. disturbances) should be excluded. `NULL` by default, which presumes no subsetting based on years is done internally (either the user supplies a pre-filtered `rstLCChange`, or no filtering is desired). If not `NULL` `rstLCChangeYr` is used to filter disturbed pixels within the specified years. See https://opendata.nfis.org/mapserver/nfis-change_eng.html for more information.

minCoverThreshold

numeric

5

0

100

Cover that is equal to or below this number will be omitted from the dataset Should be the same as the one used when preparing `cohortData` in the simulation set up.

obsDeltaAgeB

logical

TRUE

NA

NA

When TRUE, the observed changes in biomass and age (deltaB, deltaAge) between the two validation years will be plotted as maps and scatterplots

pixelGroupBiomassClass

numeric

100

NA

NA

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 Should be the same as the one used when preparing `cohortData` in the simulation set up.

sppEquivCol

character

Boreal

NA

NA

The column in `sim$sppEquiv` data.table to use as a naming convention

validationReps

integer

1, 2, 3,. . . .

NA

NA

The simulation repetitions for the validation. Defaults to 1:10. Set to NA if not using repetitions (i.e. only one run)

validationYears

integer

2001, 2011

NA

NA

The simulation years for the validation. Defaults to 2001 and 2011. Must select two years

.plotInitialTime

integer

1

NA

NA

If NA plotting is off completely (this includes saving).

.plots

character

object, png

NA

NA

Passed to `types` in Plots (see ?Plots). There are a few plots that are made within this module, if set. Note that plots (or their data) are saved in file.path(outputPath(sim), 'figures'). If `NA`, plotting is off completely (this includes plot saving).

.saveInitialTime

numeric

NA

NA

NA

This describes the simulation time at which the first save event should occur

.saveInterval

numeric

NA

NA

NA

This describes the simulation time interval between save events

.studyAreaName

character

NA

NA

NA

Human-readable name for the study area used. If `NA`, a hash of `studyArea` will be used.

.useCache

logical

init

NA

NA

Controls cache; caches the init event by default

Of the parameters listed in Table 8.2.3.2, the following are particularly important:

- `LCChangeYr` – integer. Optional parameter defining the years of disturbance that should be filtered out of the analysis using the `rstLCChangeYr` layer. This parameter is set to `NULL` by default, meaning that `rstLCChangeYr` will not be used.

- `sppEquivCol` – character. the column name in `speciesEquivalency` data.table that defines the naming convention to use throughout the simulation.

- `validationReps` – integer. which simulation replicates should be used for the validation.

- `validationYears` – integer. What simulation years should be used for the validation - the year number needs to match the observed data year. For instance, if the first observed data year is 2001, that must be the first simulation year.

8.2.4 Simulation flow

The general flow of *Biomass_validationKNN* processes is:

1. Preparation of all necessary objects, namely obtaining the observed data layers from online repositories (or if available stored local copies) and the compiling simulated data if the user has not done so previously (see Input objects).

2. Calculation of summary variables for validation, namely:

- relative biomass per species per pixel and across the landscape (per year and per replicate)

- changes in species biomass per pixel and across the landscape (per replicate), with respect to the first year.

- species dominance across the landscape

- species presences across the landscape

3. Calculation of validation statistics, namely mean absolute deviations (MAD) and sum of negative log-likelihoods (SNLL).

4. Assessment of the relationship between observed ΔB and observed ΔAge – this is an optional visual diagnostic of the observed data that produces scatterplots of ΔB ~ ΔAge of three types:

- With raw observed values of ΔB and ΔAge

- With ΔB and ΔAge calculated on observed data *after* pre-processing (i.e., the data clean-steps done in `Biomass_borealDataPrep`, which are also done to the observed data before validation)

- With the data shown in 2) above, but filtered by pixels where there was only a stand age increment corresponding to the number of years of between the two validation time points. This is not necessarily a *correct* filter, as stands may have suffered an age reduction due to the loss of old cohorts from background mortality (i.e., not coming from disturbances. However, if using the default input datasets, it is unlikely that this is a widespread phenomenon in only 10 years. We remind the user that disturbed pixels should be removed from the analyses when validating succession dynamics in the absence of disturbance - the default option.

5. Plots:

- Barplots of landscape-wide and pixel-level comparisons between observed and simulated data, with respect to relative biomass, dominance and presences.

- Boxplots of biomass changes (ΔB) in observed and simulated data, with respect to the first year.

- Maps of biomass and age changes (ΔB, ΔAge) with respect to the first year, in observed and simulated data.

All module default outputs are in the form of plots, but the user can chose to save any objects (see Table 8.1.4).

8.3 Usage example

8.3.1 Load `SpaDES` and other packages.

```
library(SpaDES)
library(SpaDES.install)
library(SpaDES.experiment)
library(future)
```

8.3.2 Get the modules

Because *Biomass_validationKNN* is meant to validate simulation outputs against observed data, we need to first run a simulation of forest dynamics with *Biomass_core*. To do that we get both modules' code from the PredictiveEcology GitHub repository. Notice that we are placing all module code, inputs and outputs in temporary directories.

```
tempDir <- tempdir()
spadesModulesDirectory <- file.path(tempDir, "modules")
paths <- list(inputPath = file.path(tempDir, "inputs"),
              cachePath = file.path(tempDir, "cache"),
              modulePath = spadesModulesDirectory,
              outputPath = file.path(tempDir, "outputs"))

getModule("PredictiveEcology/Biomass_core", modulePath = spadesModulesDirectory, overwrite = TRUE
getModule("PredictiveEcology/Biomass_validationKNN", modulePath = spadesModulesDirectory, overwri

## by default the repository branch name is appended to the module folder name.
## so we change the folder name to remove the "-master" suffix.
file.rename(c(file.path(spadesModulesDirectory, "Biomass_core-master"),
              file.path(spadesModulesDirectory, "Biomass_validationKNN-master")),
            c(file.path(spadesModulesDirectory, "Biomass_core"),
              file.path(spadesModulesDirectory, "Biomass_validationKNN")))
```

8.3.3 Setup simulation

```
times <- list(start = 2001, end = 2011)

studyArea <- Cache(randomStudyArea, size = 1e7) # cache this so it creates a random one only once
```

```r
# Pick the species you want to work with -- using the naming convention in "Boreal" co
speciesNameConvention <- "Boreal"
speciesToUse <- c("Pice_Gla", "Popu_Tre", "Pinu_Con")

sppEquiv <- LandR::sppEquivalencies_CA[get(speciesNameConvention) %in% speciesToUse]
# Assign a colour convention for graphics for each species
sppColorVect <- LandR::sppColors(sppEquiv, speciesNameConvention,
                                 newVals = "Mixed", palette = "Set1")

## Usage example
modules <- as.list("Biomass_core")
objects <- list(studyArea = studyArea, sppEquiv = sppEquiv, sppColorVect = sppColorVect

successionTimestep <- 20L

## keep default values for most parameters
## (ommitted from this list)
parameters <- list(
  Biomass_core = list(
    "sppEquivCol" = speciesNameConvention
    , "successionTimestep" = successionTimestep
    , ".plotInitialTime" = times$start
    , ".plotInterval" = 1L
    , ".plots" = "png"
    , ".saveInitialTime" = times$start
    , ".useCache" = "init"
    , ".useParallel" = FALSE
  )
)

outputs <- data.frame(expand.grid(objectName = "cohortData",
                                  saveTime = unique(seq(times$start, times$end, by = 1)
                                  eventPriority = 1,
                                  stringsAsFactors = FALSE))
outputs <- rbind(outputs, data.frame(objectName = "pixelGroupMap",
                                     saveTime = unique(seq(times$start, times$end, by =
                                     eventPriority = 1))
```

8.3.4   Run simulation

Here we run a simulation with three replicates using the `experiment2` function
of the `SpaDES.experiment` R package [10], which builds a folder structure where
simulation outputs are conveniently organised.

```
graphics.off()
mySimInit <- simInit(times = times,
                     params = parameters,
                     modules = modules,
                     objects = objects,
                     paths = paths,
                     outputs = outputs)

plan(sequential)
mySimExperiment <- experiment2(
  sim1 = mySimInit,
  clearSimEnv = FALSE,
  replicates = 3)
```

8.3.5   Validate simulation outputs with *Biomass_validationKNN*

Note that because we ran *Biomass_core* by itself using theoretical input data,
we can expect the validation to reveal that the module didn't do a great job at
reproducing observed patterns.

```
simulationOutputs <- lapply(mySimExperiment, FUN = function(x, localSimPaths) {
  oldPath <- dirname(outputPath(x)) ## exclude sim*_rep* folder
  DT <- as.data.table(outputs(x))
  DT[, file := sub(oldPath, localSimPaths$outputPath, file)]
  DT
}, localSimPaths = as.list(normPath(paths)))
simulationOutputs <- rbindlist(simulationOutputs)

validationPaths <- as.list(normPath(paths))
validationPaths$outputPath <- file.path(validationPaths$outputPath, "validation")

validationTimes <- list(start = 1, end = 1)
validationParams <- list(
  Biomass_validationKNN = list(
    "sppEquivCol" = params(mySimInit)$Biomass_core$sppEquivCol
    , "validationReps" = as.integer(1:3)  ## or length of simLists
    , "validationYears" = as.integer(c(2001, 2011))
    , ".plots" = c("png")
  )
)

## make an empty fire polygon object to bypass removing fire-disturbed pixels
noFires <- sf::st_polygon()
validationObjects <- list(
```

```
  "biomassMap" = mySimExperiment$sim1_rep1$biomassMap
  , "firePerimeters" = noFires
  , "rasterToMatch" = mySimExperiment$sim1_rep1$rasterToMatch
  , "rawBiomassMapStart" = mySimExperiment$sim1_rep1$biomassMap
  , "simulationOutputs" = simulationOutputs
  , "speciesLayersStart" = mySimExperiment$sim1_rep1$speciesLayers
  , "sppColorVect" = mySimExperiment$sim1_rep1$sppColorVect
  , "sppEquiv" = mySimExperiment$sim1_rep1$sppEquiv
  , "studyArea" = mySimExperiment$sim1_rep1$studyArea
)

mySimValidation <- simInitAndSpades(times = validationTimes
                                    , params = validationParams
                                    , modules = "Biomass_validationKNN"
                                    , objects = validationObjects
                                    , paths = validationPaths
                                    , .studyAreaName = SAname)
```

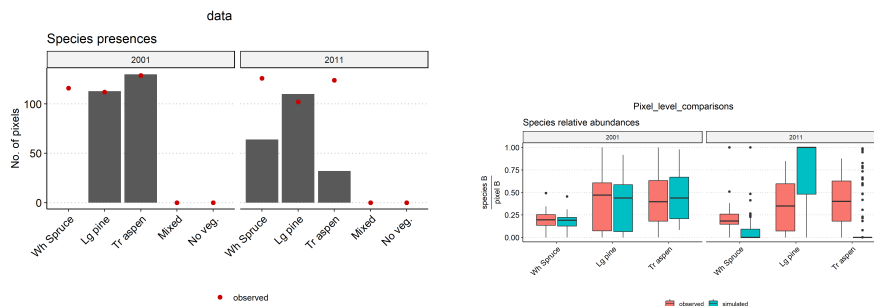Here are some of the output figures automatically produced by *Biomass_validationKNN*

Figure 8.1: *Biomass_validationKNN* automatically generates plots showing a visual comparison between simulated and observed species presences (right) across the landscape, and relative species biomass per pixel (left).
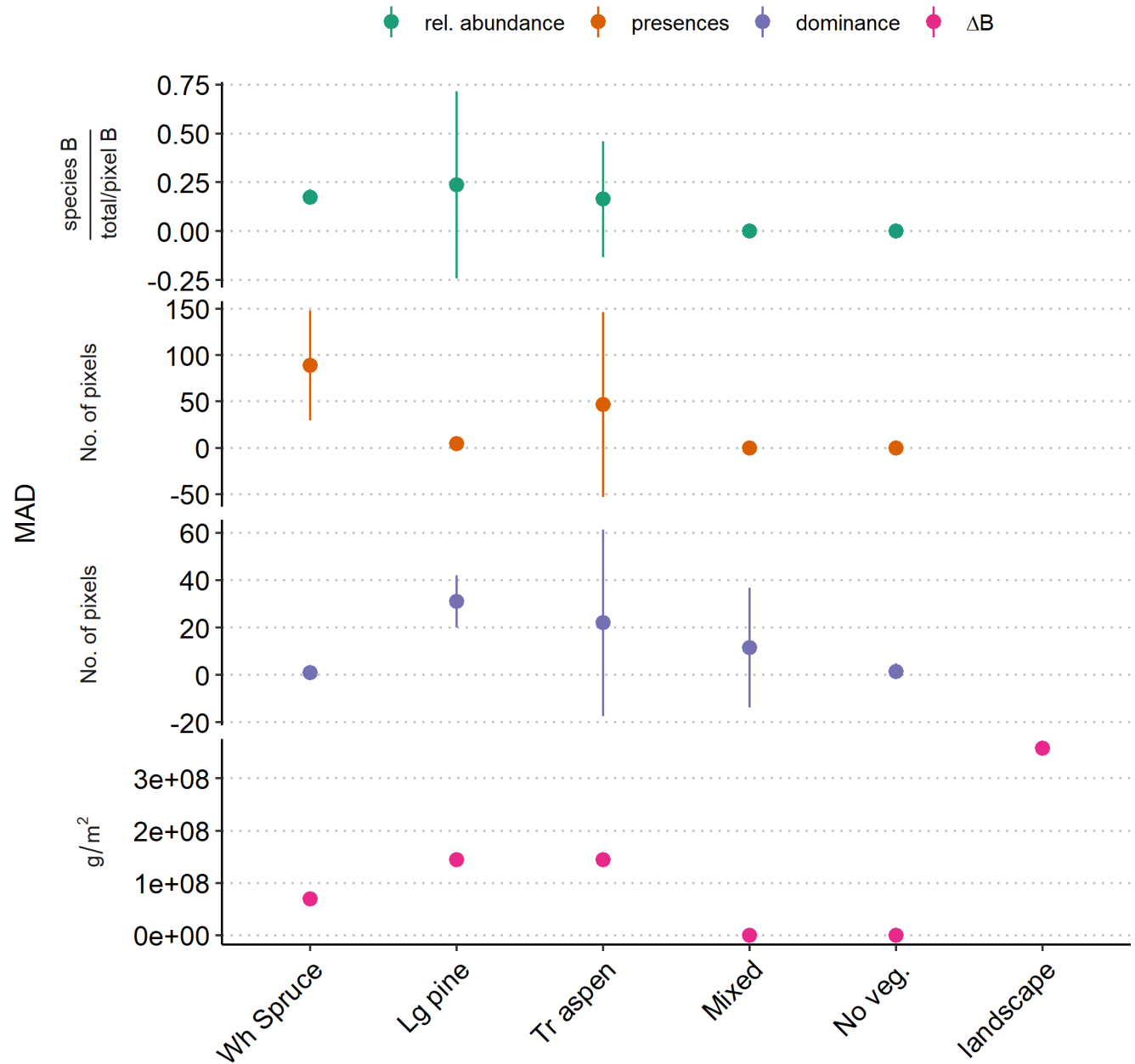
8.4   References

Figure 8.2: A plot of landscape-wide mean absolute deviations (MAD) from (top to bottom) observed mean relative abundance, no. of presences, no. of pixels where the species is dominant and ΔB.
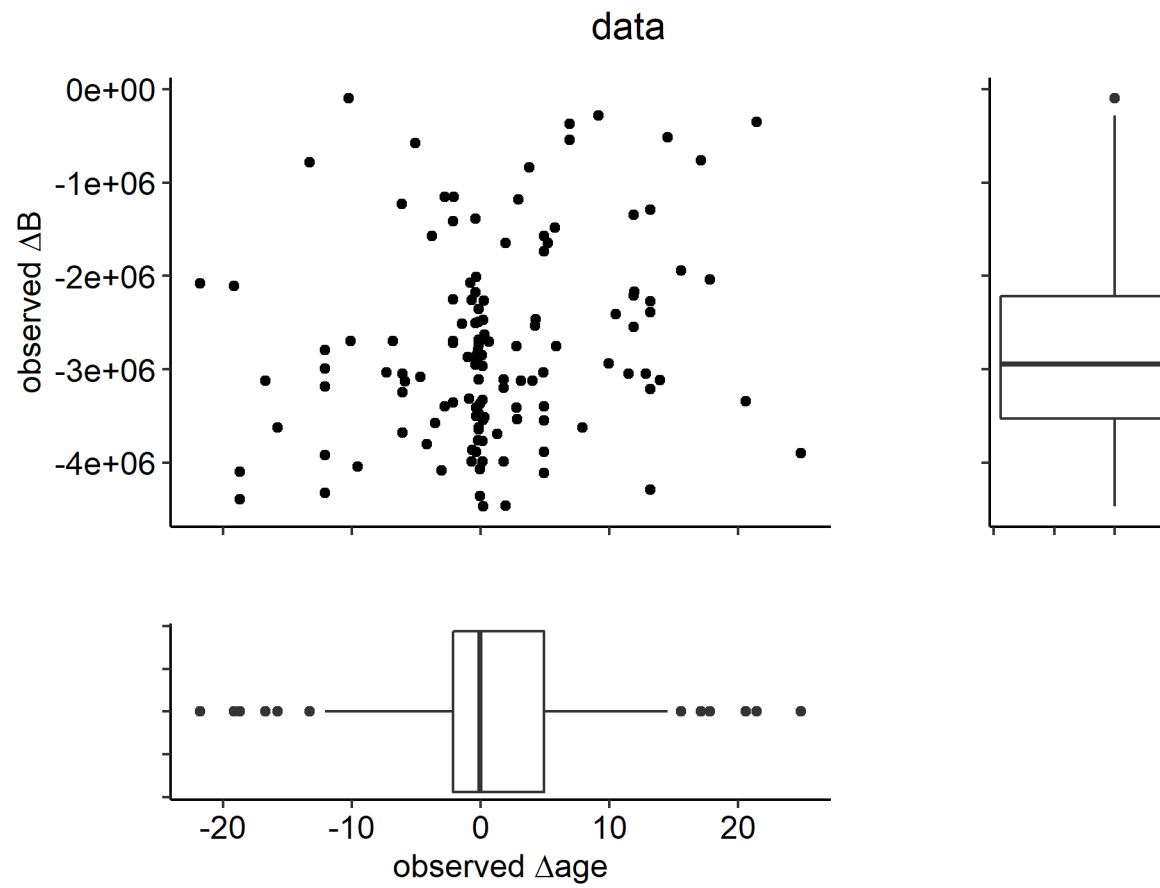
Figure 8.3: Diagnostic plot of observed changes in biomass and age ΔB and ΔAge, respectively).

Bibliography

[1]   Ceres Barros et al. "Empowering ecologists with a PERFICT workflow: seamlessly linking data, parameterization, prediction, validation and visualization".

[2]   Ceres Barros et al. "Empowering ecologists with a PERFICT workflow: seamlessly linking data, parameterization, prediction, validation and visualization".

[3]   A Beaudoin et al. *Species composition, forest properties and land cover types across Canada's forests at 250m resolution for 2001 and 2011*. 2017. DOI: 10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990. URL: http://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990.

[4]   P J Burton and S G Cumming. "Potential effects of climatic change on some western Canadian forests, based on phenological enhancements to a patch model of forest succession". In: *Water, Air and Soil Pollution* 82 (1995), pp. 401–414.

[5]   Alex M. Chubaty and Eliot J. B. McIntire. *SpaDES: Develop and Run Spatially Explicit Discrete Event Simulation Models*. 2019. URL: https://CRAN.R-project.org/package=SpaDES.

[6]   John Cosco. *COMMON ATTRIBUTE SCHEMA (CAS) FOR FOREST INVENTORIES ACROSS CANADA*. Feb. 2011, p. 117.

[7]   E. H. (Ted) Hogg, J. P. Brandt, and M. Michaelian. "Impacts of a regional drought on the productivity, dieback, and biomass of western Canadian aspen forests". In: *Canadian Journal of Forest Research* 38.6 (June 2008), pp. 1373–1384. ISSN: 0045-5067, 1208-6037. DOI: 10.1139/X08-001. URL: http://www.nrcresearchpress.com/doi/10.1139/X08-001.

[8]   M. C. Lambert, C. H. Ung, and F. Raulier. "Canadian national tree aboveground biomass equations". In: *Canadian Journal of Forest Research* 35 (2005), pp. 1996–2018. URL: http://cdnsciencepub.com/doi/abs/10.1139/x05-112.

[9]   Eliot J. B. McIntire and Alex M. Chubaty. *reproducible: A Set of Tools that Enhance Reproducibility Beyond Package Management*. 2020. URL: https://reproducible.predictiveecology.org, %20https://github.com/PredictiveEcology/reproducible.

[10]   Eliot J. B. McIntire and Alex M. Chubaty. *SpaDES.experiment: Simulation Experiments Within The SpaDES Ecosystem*. 2021. URL: https://github.com/PredictiveEcology/SpaDES.experiment.

[11]   Christian Messier, Sylvain Parent, and Yves Bergeron. "Effects of overstory and understory vegetation on the understory light environment in mixed boreal forests". In: *Journal of Vegetation Science* 9.4 (Aug. 1998), pp. 511–520. ISSN: 11009233. DOI: 10.2307/3237266. URL: http://doi.wiley.com/10.2307/3237266.

[12]   NRCan. *National Ecological Framework for Canada - Terrestrial Ecozones*. 2013.

[13]   Changhui Peng et al. "A drought-induced pervasive increase in tree mortality across Canada's boreal forests". In: *Nature Climate Change* 1.9 (Dec. 2011), pp. 467–471. ISSN: 1758-678X, 1758-6798. DOI: 10.1038/nclimate1293. URL: http://www.nature.com/articles/nclimate1293.

[14]   Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – Conceptual Description*. Apr. 20, 2012.

[15]   Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – User Guide*. July 19, 2011.

[16]   Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.

[17]   Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.

[18]   Robert M. Scheller and David J. Mladenoff. "A forest growth and biomass module for a landscape simulation model, LANDIS: design, validation, and application". In: *Ecological Modelling* 180.1 (Dec. 2004), pp. 211–229. DOI: 10.1016/j.ecolmodel.2004.01.022. URL: https://linkinghub.elsevier.com/retrieve/pii/S0304380004003837.

[19]   Shelley Stall et al. "Make scientific data FAIR". In: *Nature* 570.7759 (June 2019), pp. 27–29. DOI: 10.1038/d41586-019-01720-7. URL: http://www.nature.com/articles/d41586-019-01720-7.

[20]   Chhun-Huor Ung, Pierre Bernier, and Xiao-Jing Guo. "Canadian national biomass equations: new parameter estimates that include British Columbia data". In: *Canadian Journal of Forest Research* 38.5 (May 2008), pp. 1123–1132. ISSN: 0045-5067. DOI: 10.1139/X07-224. URL: https://cdnsciencepub-com.eu1.proxy.openathens.net/doi/full/10.1139/X07-224.

[21]   Hadley Wickham. "testthat: Get Started with Testing". In: *The R Journal* 3.1 (2011), p. 5. DOI: 10.32614/RJ-2011-002. URL: https://journal.r-project.org/archive/2011/RJ-2011-002/index.html.

[22] Mark D. Wilkinson et al. "The FAIR Guiding Principles for scientific data management and stewardship". In: *Scientific Data* 3.1 (Dec. 2016), p. 160018. ISSN: 2052-4463. DOI: 10.1038/sdata.2016.18. URL: http://www.nature.com/articles/sdata201618.

[23] Jian Yang et al. "An innovative computer design for modeling forest landscape change in very large spatial extents with fine resolutions". In: *Ecological Modelling* 222.15 (Aug. 10, 2011), pp. 2623–2630. DOI: 10.1016/j.ecolmodel.2011.04.032. URL: https://www.sciencedirect.com/science/article/pii/S0304380011002651.