

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

LandR Manual

v. 1.0.0



Contents

List of Figures	7
List of Tables	11
Preface	15
1 LandR <i>Biomass_core</i> Module	17
1.1 Module Overview	17
1.1.1 Quick links	17
1.1.2 Summary	18
1.1.3 Links to other modules	18
1.2 Module manual	20
1.2.1 General functioning	20
1.2.2 Initialisation, inputs and parameters	22
1.2.3 List of input objects	27
1.2.4 List of parameters	29
1.2.5 List of outputs	31
1.2.6 Simulation flow and module events	31
1.2.7 Differences between <i>Biomass_core</i> and the LANDIS-II Biomass Succession Extension model (LBSE)	34
1.3 Usage example	43
1.3.1 Set up R libraries	43
1.3.2 Get the module and module dependencies	44
1.3.3 Setup simulation	44
1.3.4 Run simulation	46
1.4 Appendix	47
1.4.1 Tables	47
1.5 References	47
2 LandR Data and Calibration Modules	55
3 LandR <i>Biomass_speciesData</i> Module	57

3.1	Module Overview	57
3.1.1	Quick links	57
3.1.2	Module summary	58
3.1.3	Links to other modules	58
3.2	Module manual	58
3.2.1	General functioning	58
3.2.2	List of input objects	59
3.2.3	List of parameters	61
3.2.4	List of outputs	62
3.2.5	Simulation flow and module events	62
3.3	Usage example	64
3.3.1	Load SpaDES and other packages.	64
3.3.2	Get module, necessary packages and set up folder directories	64
3.3.3	Setup simulation	65
3.3.4	Run module	66
3.4	References	67
4	LandR Biomass_borealDataPrep Module	69
4.1	Module Overview	69
4.1.1	Quick links	69
4.1.2	Summary	70
4.1.3	Links to other modules	70
4.2	Module manual	71
4.2.1	General functioning	71
4.2.2	Data acquisition and treatment	72
4.2.3	Parameter estimation/calibration	76
4.2.4	Agregating species	83
4.2.5	List of input objects	84
4.2.6	List of parameters	85
4.2.7	List of outputs	86
4.2.8	Simulation flow and module events	88
4.3	Usage example	90
4.4	References	90
5	LandR Biomass_speciesParameters Module	91
5.1	Module Overview	91
5.1.1	Quick links	91
5.1.2	Summary	92
5.1.3	Links to other modules	92

<i>Contents</i>	5
5.2 Module manual	93
5.2.1 General functioning	93
5.2.2 List of input objects	97
5.2.3 List of parameters	98
5.2.4 List of outputs	99
5.2.5 Simulation flow and module events	99
5.3 Usage example	101
5.3.1 Load SpaDES and other packages.	102
5.3.2 Get module, necessary packages and set up folder directories	102
5.3.3 Setup simulation	102
5.4 References	104
6 LandR Validation Modules	105
7 LandR <i>Biomass_validationKNN</i> Module	107
7.1 Module Overview	107
7.1.1 Quick links	107
7.1.2 Summary	108
7.1.3 Links to other modules	108
7.2 Module manual	109
7.2.1 General functioning	109
7.2.2 Validation approaches	111
7.2.3 List of input objects	113
7.2.4 List of parameters	115
7.2.5 List of outputs	118
7.2.6 Simulation flow and module events	118
7.3 Usage example	121
7.3.1 Load SpaDES and other packages.	121
7.3.2 Get the modules	121
7.3.3 Setup simulation	122
7.3.4 Run simulation	123
7.3.5 Validate simulation outputs with <i>Biomass_validationKNN</i>	124
7.4 References	127



List of Figures

- 1.1 *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). 18
- 1.2 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 $\frac{Biomass_{order2} - Biomass_{order1}}{Biomass_{order2}} * 100$ 36
- 1.3 Differences in the biomass 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. Values refer to the initial total biomass attributed to each species at the end of year 1. 37

1.4	Hashing design for <i>Biomass_core</i> . In the re-coded <i>Biomass_core</i> , the pixel group map was hashed based on the unique combination of species composition ('community map') and ecolocation map, and associated with a lookup table. The insert in the top-right corner was the original design that linked the map to the lookup table by pixel key.	40
1.5	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 <i>Biomass_core</i> were calculated as $\frac{Biomass_{LBSE} - Biomass_{Biomass_core}}{Biomass_{LBSE}} * 100$	42
1.6	Simulation efficiencies of LBSE and <i>Biomass_core</i> with increasing map size, in terms of a) mean running time across repetitions (left y-axis) and the ratio LBSE to <i>Biomass_core</i> running times (right y-axis and blue line), and b) running time scalability as the mean running time per 1000 pixels.	43
1.7	<i>Biomass_core</i> 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).	47
3.1	<i>Biomass_speciesData</i> automatically generates a plot of species dominance and number of presences in the study area when 'plotInitialTime=1' is passed as an argument.	67
4.1	Modelling biomass as a linear vs. exponential relationship. a) 'modelBiomass' as 'B ~ logAge * speciesCode + cover * speciesCode + (logAge + cover ecoregionGroup)'. b) 'modelBiomass' as 'logB ~ logAge * speciesCode + cover * speciesCode + (logAge + cover ecoregionGroup)'. Blue dots are marginal mean B values (back-transformed in b) cross ages with confidence intervals as the bars.	79
4.2	Thirty years of simulation with 'maxB' values estimated from a 'logB ~ ...' 'biomassModel' (see Fig. reffig:fig-biomassModelLogBtest). The steep increase in such little time is abnormal.	79

- 7.1 *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). 125
- 7.2 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 126
- 7.3 Diagnostic plot of observed changes in biomass and age ΔB and ΔAge , respectively). 126



List of Tables

1.1	Example of an invariant species traits table (the ‘species’ table object in the module), with species <i>Abies sp.</i> (Abie_sp), <i>Picea engelmannii</i> (Pice_eng), <i>Picea glauca</i> (Pice_gla), <i>Pinus sp.</i> (Pinu_sp), <i>Populus sp.</i> (Popu_sp) and <i>Pseudotsuga menziesii</i> (Pseu_men). Note that these are theoretical values.	24
1.2	Example of a spatio-temporally varying species traits table (the ‘speciesEcoregion’ table object in the module), with two ecolocations (called ‘ecoregionGroups’) and species <i>Abies sp.</i> (Abie_sp), <i>Picea engelmannii</i> (Pice_eng), <i>Picea glauca</i> (Pice_gla), <i>Pinus sp.</i> (Pinu_sp), <i>Populus sp.</i> (Popu_sp) and <i>Pseudotsuga menziesii</i> (Pseu_men). If a simulation runs for 10 year using this table, trait values from year 2 would be used during simulation years 2-10.	25
1.3	Example of a minimum relative biomass table (the ‘minRelativeB’ table object in the module), with two ecolocations (‘ecoregionGroups’) sharing the same values	26
1.4	Default species probability of germination values used by <i>Biomass_core</i> and <i>Biomass_borealDataPrep</i> . Columns X0-X5 are different site shade levels and each line has the probability of germination for each site shade and species shade tolerance combination.	26
1.5	List of <i>Biomass_core</i> input objects and their description. . . .	30
1.6	List of <i>Biomass_core</i> parameters and their description. . . .	32
1.7	List of <i>Biomass_core</i> output objects and their description. . .	33
1.8	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 ‘Calibrate-Mode’ was set to ‘yes’. Species starting ages are also shown.	48

1.9	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.	49
1.10	Randomly generated community combination no. 1 used in the recruitment comparison runs.	50
1.11	Randomly generated community combination no. 3 used in the recruitment comparison runs.	51
1.12	Invariant species traits table used in comparison runs. . .	52
1.13	Minimum relative biomass table used in comparison runs. XO-5 represent site shade classes from no-shade (0) to maximum shade (5). All ecolocations shared the same values. . .	52
1.14	Probability of germination for species shade tolerance and shade level combinations (called <i>sufficient light</i> table in LBSE and 'sufficientLight' input 'data.table' in LandR <i>Biomass_core</i>) used in comparison runs.	52
1.15	Species ecolocation table used in comparison runs. 'SEP' stands for species establishment probability, 'maxB' for maximum biomass and 'maxANPP' for maximum above-ground net primary productivity. Values were held constant throughout the simulation.	53
3.1	Example of species merging for simulation. Here the user wants to model <i>Abies balsamea</i> , <i>A. lasiocarpa</i> and <i>Pinus contorta</i> as separate species, but all <i>Picea spp.</i> as a genus-level group. For this, all six species are separately 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 <i>Picea spp.</i> have the same name. <i>Biomass_speciesData</i> will merge their % cover data into a single layer by summing their cover per pixel. . .	60
3.2	List of <i>Biomass_speciesData</i> input objects and their description.	60
3.3	List of <i>Biomass_speciesData</i> parameters and their description. . .	61
3.4	List of <i>Biomass_speciesData</i> output objects and their description.	62

3.5	List of species cover data downloaded by default by <i>Biomass_speciesData</i>	63
4.1	Example of species merging for simulation. Here the user wants to model <i>Abies balsamea</i> , <i>A. lasiocarpa</i> and <i>Pinus contorta</i> as separate species, but all <i>Picea spp.</i> as a genus-level group. 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 <i>Picea spp.</i> have the same name. <i>Biomass_borealDataPrep</i> will merge their % cover data into a single layer by summing their cover per pixel.	83
4.2	List of <i>Biomass_borealDataPrep</i> input objects and their description.	85
4.3	List of <i>Biomass_borealDataPrep</i> parameters and their description.	87
4.4	List of <i>Biomass_borealDataPrep</i> output objects and their description.	89
5.1	List of <i>Biomass_speciesParameters</i> input objects and their description.	98
5.2	List of <i>Biomass_speciesParameters</i> parameters and their description.	100
5.3	List of <i>Biomass_speciesParameters</i> output objects and their description.	101
7.1	List of <i>Biomass_validationKNN</i> input objects and their description.	115
7.2	List of <i>Biomass_validationKNN</i> parameters and their description.	117
7.3	List of <i>Biomass_validationKNN</i> output objects and their description.	119



Preface



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

The present manual is a “live” document, in that grows and changes 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., a LandR Biomass manual would describe only the LandR modules that are essential for the simulation of the vegetation components of forest succession, excluding disturbances or carbon cycling.

This manual does not cover the SpaDES toolkit, which we use to create and run LandR modules. To learn more about SpaDES go to the SpaDES webpage¹.

¹<https://spades.predictiveecology.org/>

1

LandR Biomass_core Module

Biomass core v. 1.3.9¹

1.0.0.1 Authors:

Yong Luo yluo1@lakeheadu.ca² [aut], Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca³ [aut, cre], Ceres Barros ceres.barros@ubc.ca⁴ [aut], Alex M. Chubaty achubaty@for-cast.ca⁵ [aut], Ian Eddy ian.eddy@nrcan-rncan.gc.ca⁶ [ctb], Jean Marchal jean.d.marchal@gmail.com⁷ [ctb]

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

1.1 Module Overview

1.1.1 Quick links

- [General functioning](#)
- [List of input objects](#)
- [List of parameters](#)
- [List of outputs](#)
- [Simulation flow and module events](#)

¹https://github.com/PredictiveEcology/Biomass_core

²<mailto:yluo1@lakeheadu.ca>

³<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

⁴<mailto:ceres.barros@ubc.ca>

⁵<mailto:achubaty@for-cast.ca>

⁶<mailto:ian.eddy@nrcan-rncan.gc.ca>

⁷<mailto:jean.d.marchal@gmail.com>

1.1.2 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. 1.1), in a spatially explicit manner and using a yearly time step. The model is based on the LANDIS-II Biomass Succession Extension v.3.2.1 [LBSE, 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 LBSE manual [16] for a detailed reading of the mechanisms implemented in the model.

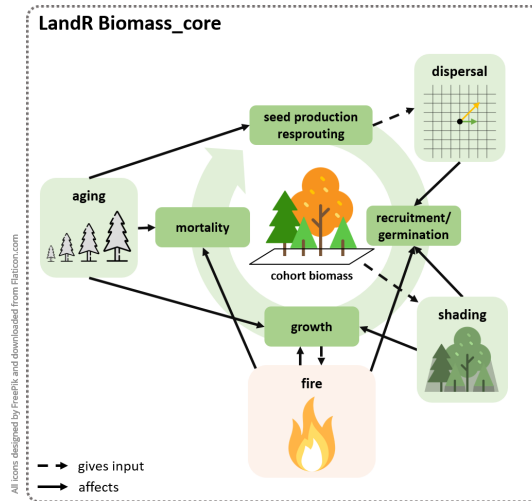


FIGURE 1.1: *Biomass_core* simulates tree cohort growth, mortality, recruitment and dispersal dynamics, as a function of cohort ageing and competition for light (shading) and space, as well as disturbances like fire (simulated using other modules).

1.1.3 Links to other modules

Biomass_core is intended to be used with data/calibration modules, disturbance modules and validation modules, amongst others. The following is a list of the modules most commonly used with *Biomass_core*. Not all are in the

LandR Manual⁸, but see each module's documentation (.Rmd file) available in its repository.

See here⁹ for all available modules and select *Biomass_core* from the drop-down menu to see linkages.

Data and calibration modules:

- *Biomass_speciesData*¹⁰: grabs and merges several sources of species cover data, making species percent cover (% cover) layers used by other LandR Biomass modules. Default source data spans the entire Canadian territory;
- *Biomass_borealDataPrep*¹¹: prepares all parameters and inputs (including initial landscape conditions) that *Biomass_core* needs to run a realistic simulation. Default values/inputs produced are relevant for boreal forests of Western Canada;
- *Biomass_speciesParameters*¹²: calibrates four-species level traits using permanent sample plot data (i.e., repeated tree biomass measurements) across Western Canada.

Disturbance-related modules:

- *Biomass_regeneration*¹³: simulates cohort biomass responses to stand-replacing fires (as in LBSE), including cohort mortality and regeneration through resprouting and/or serotiny;
- *Biomass_regenerationPM*¹⁴: like *Biomass_regeneration*, but allowing partial mortality. Based on the LANDIS-II Dynamic Fuels & Fire System extension [20];
- *fireSense*: climate- and land-cover-sensitive fire model simulating fire ignition, escape and spread processes as a function of climate and land-cover. Includes built-in parameterisation of these processes using climate, land-cover, fire occurrence and fire perimeter data. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*. See modules prefixed “*fireSense_*” at <https://github.com/PredictiveEcology/>;

⁸<https://landr-manual.predictiveecology.org/>

⁹https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

¹⁰https://github.com/PredictiveEcology/Biomass_speciesData

¹¹https://github.com/PredictiveEcology/Biomass_borealDataPrep

¹²https://github.com/PredictiveEcology/Biomass_speciesParameters

¹³https://github.com/PredictiveEcology/Biomass_regeneration

¹⁴https://github.com/PredictiveEcology/Biomass_regenerationPM

- *LandMine*¹⁵: wildfire ignition and cover-sensitive wildfire spread model based on a fire return interval input. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*;
- *scfm*¹⁶: spatially explicit fire spread module parameterised and modelled as a stochastic three-part process of ignition, escape, and spread. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*.

Validation modules:

- *Biomass_validationKNN*¹⁷: calculates two validation metrics (mean absolute deviation and sum of negative log-likelihoods) on species presences/absences and biomass-related properties across the simulated landscape. By default, it uses an independent dataset of species % cover and stand biomass for 2011, assuming that this is a second snapshot of the landscape.

1.2 Module manual

1.2.1 General functioning

Biomass_core is a forest landscape model based on the LANDIS-II Biomass Succession Extension v.3.2.1 model [LBSE, 16]. It is the core forest succession model of the LandR ecosystem of SpaDES modules. Similarly to 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. 1.1). Note that, by default, cohorts are unique combinations of species and age, but this can be changed via the `cohortDefinitionCols` parameter (see [List of parameters](#)).

Specifically, cohort growth is driven by both invariant (growth shape parameter, `growthcurve`) and spatio-temporally varying species traits (maximum biomass, `maxB`, and maximum annual net primary productivity, `maxANPP`), while background mortality (i.e., not caused by disturbances) depends only on invariant species traits (`longevity` and mortality shape parameter, `mortalityshape`). All these five traits directly influence the realised shape of species growth curves, by determining how fast they grow

¹⁵<https://github.com/PredictiveEcology/LandMine>

¹⁶<https://github.com/PredictiveEcology/scfm>

¹⁷https://github.com/PredictiveEcology/Biomass_validationKNN

(*growthcurve* and *maxANPP*), how soon age mortality starts with respect to longevity (*mortalityshape*) and the biomass a cohort can potentially achieve (*maxB*).

Cohort recruitment is determined by available “space” (i.e., pixel shade), invariant species traits (regeneration mode, *postfireregen*, age at maturity, *sexualmature*, shade tolerance, *shadetolerance*) and a third spatio-temporally varying trait (species establishment probability, *establishprob*, called SEP hereafter). The available “growing space” is calculated as the species’ *maxB* minus the occupied biomass (summed across other cohorts in the pixel). If there is “space”, a cohort can establish from one of three recruitment modes: serotiny, resprouting and germination.

Disturbances (e.g., fire) can cause cohort mortality and trigger post-disturbance regeneration. Two post-disturbance regeneration mechanisms have been implemented, following LBSE: serotiny and resprouting [16]. Post-disturbance mortality and regeneration only occur in response to fire and are simulated in two separate, but interchangeable modules, *Biomass_regeneration* and *Biomass_regenerationPM* that differ with respect to the level of post-fire mortality they simulate (complete or partial mortality, respectively).

Cohort germination (also called cohort establishment) occurs if seeds are available from local sources (the pixel), or via seed dispersal. Seed dispersal can be of three modes: ‘no dispersal’, ‘universal dispersal’ (arguably, only interesting for dummy case studies) or ‘ward dispersal’ [16]. Briefly, 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.

Finally, both germination and regeneration success depend on the species’ probability of germination in a given pixel (**probabilities of germination**).

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 above mentioned mechanisms implemented in *Biomass_core*. In a later section of this manual, we highlight existing **differences between *Biomass_core* and LBSE**, together with **comparisons between the two modules**.

1.2.2 Initialisation, inputs and parameters

To initialise and simulate forest dynamics in any given landscape, *Biomass_core* requires a number of inputs and parameters namely:

- **initial cohort biomass and age** values across the landscape;
- **invariant species traits** values;
- **spatio-temporally varying species traits** values (or just spatially-varying);
- **location- (ecolocation-) specific parameters**;
- and the **probabilities of germination** given a species' shade tolerance and site shade.

These are detailed below and in the **full list of input objects**. The *Biomass_borealDataPrep* module manual also provides information about the estimation of many of these traits/inputs from available data, or their adjustment using published values or our best knowledge of boreal forest dynamics in Western Canada.

Unlike the initialisation in LBSE¹⁸, *Biomass_core* initialises the simulation using data-derived initial cohort biomass and age. This information is ideally supplied by data and calibration modules like *Biomass_borealDataPrep* (**Links to other modules**), but *Biomass_core* can also initialise itself using theoretical data.

Similarly, although *Biomass_core* can create all necessary traits and parameters using theoretical values, for realistic simulations these should be provided by data and calibration modules, like *Biomass_borealDataPrep* and *Biomass_speciesParameters*. We advise future users and developers to become familiar with these data modules and then try to create their own modules (or modify existing ones) for their purpose.

1.2.2.1 Initial cohort biomass and age

Initial cohort biomass and age are derived from stand biomass (*biomassMap* raster layer), stand age (*standAgeMap* raster layer) and species % cover (*speciesLayers* raster layers) data (see Table 1.5) and formatted into the *cohortData* object. The *cohortData* table is a central simulation object that tracks the current year's cohort biomass, age, mortality (lost biomass) and

¹⁸in LBSE the initialisation consists in "iterat[ing] the number of time steps equal to the maximum cohort age for each site", beginning at 0 minus t (t = oldest cohort age) and adding cohorts at the appropriate time until the initial simulation time is reached (0) [16].

aboveground net primary productivity (ANPP) per species and pixel group (pixelGroup). At the start of the simulation, cohortData will not have any values of cohort mortality or ANPP.

Each pixelGroup is a collection of pixels that share the same ecolocation (coded in the ecoregionMap raster layer) and the same cohort composition. By default, an ecolocation is a combination of land-cover and ecological zonation (see ecoregionMap in the [full list of inputs](#)) and unique cohort compositions are defined as unique combinations of species, age and biomass. The cohortData table is therefore always associated with the current year's pixelGroupMap raster layer, which provides the spatial location of all pixelGroups, allowing to “spatialise” cohort information and dynamics (e.g., dispersal) on a pixel by pixel basis (see also [Hashing](#)).

The user, or another module, may provide initial cohortData and pixelGroupMap objects to start the simulation, or the input objects necessary to produce them: a study area polygon (studyArea), the biomassMap, standAgeMap, speciesLayers and ecoregionMap raster layers (see the [list of input objects](#) for more detail).

1.2.2.2 Invariant species traits

These are spatio-temporally constant traits that mostly influence population dynamics (e.g., growth, mortality, dispersal) and responses to fire (fire tolerance and regeneration).

By default, *Biomass_core* obtains trait values from available LANDIS-II tables (see Table 1.5), but traits can be adjusted/supplied by the user or by other modules. For instance, using *Biomass_borealDataPrep* will adjust some trait values for Western Canadian boreal forests [e.g., longevity values are adjusted following 3], while using *Biomass_speciesParameters* calibrates the growthcurve and mortalityshape parameters and estimates two additional species traits (inflationFactor and mANPPproportion) to calibrate maxB and maxANPP (respectively).

Table 1.1 shows an example of a table of invariant species traits. Note that *Biomass_core* (alone) requires all the columns Table 1.1 in to be present, with the exception of firetolerance, postfireregen, resproutprob, resproutage_min and resproutage_max, which are used by the post-fire regeneration modules (*Biomass_regeneration* and *Biomass_regenerationPM*).

Please see Scheller and Domingo [15, p.18] and Scheller and Miranda [16, p.16] for further detail.

TABLE 1.1: Example of an invariant species traits table (the ‘species’ table object in the module), with species *Abies sp.* (Abie_sp), *Picea engelmannii* (Pice_eng), *Picea glauca* (Pice_gla), *Pinus sp.* (Pinu_sp), *Populus sp.* (Popu_sp) and *Pseudotsuga menziesii* (Pseu_men). Note that these are theoretical values.

speciesCode	longevity	sexualmature	shadetolerance	firetolerance	postfireregen	resproutprob
Abie_sp	200	20	2.3	1	none	0.0
Pice_eng	460	30	2.1	2	none	0.0
Pice_gla	400	30	1.6	2	none	0.0
Pinu_sp	150	15	1.0	2	serotiny	0.0
Popu_sp	140	20	1.0	1	resprout	0.5
Pseu_men	525	25	2.0	3	none	0.0

1.2.2.3 Spatio-temporally varying species traits

These traits vary between species, by ecolocation and, potentially, by year if the year column is not omitted and several years exist (in which case last year’s values up to the current simulation year are always used). They are maximum biomass, `maxB`, maximum above-ground net primary productivity, `maxANPP`, and species establishment probability, `SEP` (called `establishprob` in the module). By default, *Biomass_core* assigns theoretical values to these traits, and thus we recommend using *Biomass_borealDataPrep* to obtain realistic trait values derived from data (by default, pertinent for Canadian boreal forest applications), or passing a custom table directly. *Biomass_speciesParameters* further calibrates `maxB` and `maxANPP` by estimating two additional invariant species traits (`inflationFactor` and `maxANPPproportion`; also for Western Canadian forests). See Table 1.2 for an example.

1.2.2.4 Ecolocation-specific parameters – minimum relative biomass

Minimum relative biomass (`minRelativeB`) is the only ecolocation-specific parameter used in *Biomass_core*. It is used to determine the shade level in each pixel (i.e., site shade) with respect to the total potential maximum biomass for that pixel (i.e., the sum of all `maxB` values in the pixel’s ecolocation). If relative biomass in the stand (with regards to the total potential maximum biomass) is above the minimum relative biomass thresholds, the pixel is assigned that threshold’s site shade value [16].

The shade level then influences the germination and regeneration of new cohorts, depending on their shade tolerance (see [Probabilities of germination](#)).

TABLE 1.2: Example of a spatio-temporally varying species traits table (the ‘speciesEcoregion’ table object in the module), with two ecolocations (called ‘ecoregionGroups’) and species *Abies* sp. (Abie_sp), *Picea engelmannii* (Pice_eng), *Picea glauca* (Pice_gla), *Pinus* sp. (Pinu_sp), *Populus* sp. (Popu_sp) and *Pseudotsuga menziesii* (Pseu_men). If a simulation runs for 10 year using this table, trait values from year 2 would be used during simulation years 2-10.

ecoregionGroup	speciesCode	establishprob	maxB	maxANPP	year
1_03	Abie_sp	1.000	8567	285	1
1_03	Pice_eng	0.983	10156	305	1
1_03	Popu_sp	0.737	8794	293	1
1_03	Pseu_men	1.000	17534	132	1
1_09	Abie_sp	0.112	1499	50	1
1_09	Pice_gla	0.302	3143	102	1
1_09	Pinu_sp	0.714	2569	86	1
1_09	Popu_sp	0.607	3292	110	1
1_09	Pseu_men	0.997	6020	45	1
1_03	Abie_sp	0.989	8943	225	2
1_03	Pice_eng	0.985	9000	315	2
1_03	Popu_sp	0.600	8600	273	2
1_03	Pseu_men	1.000	13534	142	2
1_09	Abie_sp	0.293	2099	45	2
1_09	Pice_gla	0.745	3643	90	2
1_09	Pinu_sp	0.500	2569	80	2
1_09	Popu_sp	0.670	3262	111	2
1_09	Pseu_men	1.000	6300	43	2

Site shade varies from X0 (no shade) to X5 (maximum shade). By default, *Biomass_core* uses the same minimum relative biomass threshold values across all ecolocations, adjusted from a publicly available LANDIS-II table¹⁹ to better reflect Western Canada boreal forest dynamics (see Table 1.3). *Biomass_borealDataPrep* does the same adjustment by default. As with other inputs, these values can be adjusted by using other modules or by passing user-defined tables.

¹⁹https://github.com/dcyr/LANDIS-II_IA_generalUseFiles

TABLE 1.3: Example of a minimum relative biomass table (the ‘minRelativeB’ table object in the module), with two ecolocations (‘ecoregionGroups’) sharing the same values

ecoregionGroup					
1_O3	0.15	0.25	0.5	0.75	0.85
1_O9	0.15	0.25	0.5	0.75	0.85

TABLE 1.4: Default species probability of germination values used by *Biomass_core* and *Biomass_borealDataPrep*. Columns X0-X5 are different site shade levels and each line has the probability of germination for each site shade and species shade tolerance combination.

species shade tolerance						
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

1.2.2.5 Probabilities of germination

A species’ probability of germination results from the combination of its shade tolerance level (an invariant species trait in the `species` table; Table ??(tab:invariantSpptraits)) and the site shade [defined by the amount of biomass in the pixel – see [minimum relative biomass parameter](#) and [16](#), p.14]. By default, both *Biomass_core* and *Biomass_borealDataPrep* use a publicly available LANDIS-II table (called `sufficientLight` in the module; Table [1.4](#)).

1.2.2.6 Other module inputs

The remaining module input objects either do not directly influence the basic mechanisms implemented in *Biomass_core* (e.g., `sppColorVect` and `studyAreaReporting` are only used for plotting purposes), are objects that keep track of a property/process in the module (e.g., `lastReg` is a counter of the last year when regeneration occurred), or define the study area for the simulation (e.g., `studyArea` and `rasterToMatch`).

The next section provides a complete list of all input objects, including those already mentioned above.

1.2.3 List of input objects

All of *Biomass_core*'s input objects have (theoretical) defaults that are produced automatically by the module²⁰. We suggest that new users run *Biomass_core* by itself supplying only a `studyArea` polygon, before attempting to supply their own or combining *Biomass_core* with other modules. This will enable them to become familiar with all the input objects in a theoretical setting.

Of the inputs listed in Table 1.5, 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 for consistency with original LBSE code, but we prefer to call them ecolocations to avoid confusion with the ecoregion-level classification of the National Ecological Classification of Canada (NECC)²¹. Ecolocations group pixels with similar biophysical conditions. By default, we use two levels of grouping in our applications: the first level being an ecological classification such as `ecodistricts` from the NECC, and the second level is a land-cover classification. Hence, these ecolocations contain relatively coarse scale regional information plus finer scale land cover information. The `ecoregionMap` layer must be defined as a categorical raster, 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 grouping) and `ecoregionGroup` (the full ecolocation “name” written as `<firstlevel_secondlevel>`). Note that if creating `ecoregionGroup`'s by combining two raster layers whose values are numeric (as in *Biomass_borealDataPrep*), the group label is a character combination of two numeric grouping levels. 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}`. However, the user is free to use any groupings they wish. Finally, note that all ecolocations (`ecoregionGroup`'s) are should be listed in the `ecoregion` table.

²⁰usually, default inputs are made when running the `.inputObjects` function (inside the module R script) during the `simInit` call and in the `init` event during the `spades` call – see `?SpaDES.core::events` and `SpaDES.core::simInit`

²¹<https://open.canada.ca/data/en/dataset/3ef8e8a9-8d05-4fea-a8bf-7f5023d2b6e1>

- `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 from `studyArea`, using `biomassMap` as the template for spatial resolution and projection.
- `studyArea` – 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 or another module**.

Species traits and other parameter tables

- `ecoregion` – a `data.table` listing all ecolocation “names” (*ecoregionGroup* column; see `ecoregionMap` above for details) and their state (active – yes – or inactive – no)
- `minRelativeB` – a `data.table` of minimum relative biomass values. See [Ecolocation-specific parameters – minimum relative biomass](#).
- `species` – a `data.table` of [invariant species traits](#).
- `speciesEcoregion` – a `data.table` of [spatio-temporally varying species traits](#).
- `sufficientLight` – a `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). See [Probabilities of germination](#).
- `sppEquiv` – a `data.table` of species name equivalences between various conventions. It must contain the columns *LandR* (species IDs in the 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. It 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 of the current year in g/m^2 .
- `mortality` – integer. Cohort dead biomass of the current year in g/m^2 . Usually filled with 0s in initial conditions.
- `aNPPAct` – integer. Actual aboveground net primary productivity of the current year in g/m^2 . B is the result of the previous year's B minus the current year's mortality plus aNPPAct. Usually filled with 0s in initial conditions. 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).

1.2.4 List of parameters

In addition to the above inputs objects, *Biomass_core* uses several parameters²² that control aspects like the simulation length, the “succession” time step, plotting and saving intervals, amongst others. 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 package²³ (or another similar package). These are not discussed in detail here, since climate effects are calculated externally to *Biomass_core* in `LandR.CS` functions and thus documented there.

A list of useful parameters and their description is listed below, while the

²²in `SpaDES` lingo parameters are “small” objects, such as an integer or boolean, that can be controlled via the `parameters` argument in `simInit`.

²³<https://github.com/ianmseddy/LandR.CS>

TABLE 1.5: List of *Biomass_core* input objects and their description.

objectName	objectClass	desc
biomassMap	RasterLayer	total biomass raster layer in study area
cceArgs	list	a list of quoted objects used by the 'grow'
cohortData	data.table	'data.table' with cohort-level informatio
ecoregion	data.table	ecoregion look up table
ecoregionMap	RasterLayer	ecoregion map that has mapcodes matc
lastReg	numeric	an internal counter keeping track of wh
minRelativeB	data.frame	table defining the relative biomass cut p
pixelGroupMap	RasterLayer	a raster layer with 'pixelGroup' IDs per p
rasterToMatch	RasterLayer	a raster of the 'studyArea' in the same re
species	data.table	a table of invariant species traits with th
speciesEcoregion	data.table	table of spatially-varying species traits
speciesLayers	RasterStack	percent cover raster layers of tree specie
sppColorVect	character	A named vector of colors to use for plot
sppEquiv	data.table	table of species equivalencies. See 'Land
studyArea	SpatialPolygonsDataFrame	Polygon to use as the study area. Must b
studyAreaReporting	SpatialPolygonsDataFrame	multipolygon (typically smaller/unbuffe
sufficientLight	data.frame	table defining how the species with diff
treedFirePixelTableSinceLastDisp	data.table	3 columns: 'pixelIndex', 'pixelGroup', an

full set of parameters is in Table 1.6. 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` parameter 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`.

Plotting and saving - `.plots` – activates/deactivates plotting and defines type of 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);

1.2.5 List of outputs

The main outputs of *Biomass_core* are the `cohortData` and `pixelGroupMap` containing cohort information per year (note that they are not saved by default), visual outputs of species level biomass, age and dominance across the landscape and the simulation length, and several maps of stand biomass, mortality and reproductive success (i.e, new biomass) on a yearly basis.

However, any of the objects changed/output by *Biomass_core* (listed in Table 1.7) can be saved via the `outputs` argument in `simInit`²⁴.

1.2.6 Simulation flow and module events

Biomass_core itself does not simulate disturbances or their effect on vegetation (i.e., post-disturbance mortality and regeneration). Should disturbance and post-disturbance mortality/regeneration modules be used (e.g., *LandMine* and *Biomass_regeneration*), the user should make sure that post-disturbance effects occur *after* the disturbance, but *before* dispersal and background vegetation growth and mortality (simulated in *Biomass_core*). Hence, the disturbance itself should take place either at the very beginning or at the very end of each simulation time step to guarantee that it happens immediately before post-disturbance effects are calculated.

The general flow of *Biomass_core* processes with and without disturbances is:

1. Preparation of necessary objects for the simulation – either by

²⁴see `?SpaDES.core::outputs`

TABLE 1.6: List of *Biomass_core* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
calcSummaryBGM	character	end	NA	NA	A character vector describing w
calibrate	logical	FALSE	NA	NA	Do calibration? Defaults to 'FAL
cohortDefinitionCols	character	pixelGro....	NA	NA	'cohortData' columns that deter
cutpoint	numeric	1e+10	NA	NA	A numeric scalar indicating how
gmcsGrowthLimits	numeric	66.66666....	NA	NA	if using 'LandR.CS' for climate-s
gmcsMortLimits	numeric	66.66666....	NA	NA	if using 'LandR.CS' for climate-s
gmcsMinAge	numeric	21	0	NA	if using 'LandR.CS' for climate-s
growthAndMortalityDrivers	character	LandR	NA	NA	package name where the followi
growthInitialTime	numeric	start(sim)	NA	NA	Initial time for the growth event
initialB	numeric	10	1	NA	initial biomass values of new ag
initialBiomassSource	character	cohortData	NA	NA	Currently, there are three option
keepClimateCols	logical	FALSE	NA	NA	include growth and mortality pr
minCohortBiomass	numeric	0	NA	NA	cohorts with biomass below this
mixedType	numeric	2	NA	NA	How to define mixed stands: 1 fo
plotOverstory	logical	FALSE	NA	NA	swap max age plot with overstor
seedingAlgorithm	character	wardDisp....	NA	NA	choose which seeding algorithm
spinupMortalityfraction	numeric	0.001	NA	NA	defines the mortality loss fracti
sppEquivCol	character	Boreal	NA	NA	The column in 'sim\$sppEquiv' d
successionTimestep	numeric	10	NA	NA	defines the simulation time step
vegLeadingProportion	numeric	0.8	0	1	a number that defines whether
.maxMemory	numeric	5	NA	NA	maximum amount of memory (
.plotInitialTime	numeric	start(sim)	NA	NA	Vector of length = 1, describing
.plotInterval	numeric	NA	NA	NA	defines the plotting time step. I
.plots	character	object	NA	NA	Passed to 'types' in 'Plots' (see '?I
.plotMaps	logical	TRUE	NA	NA	Controls whether maps should l
.saveInitialTime	numeric	NA	NA	NA	Vector of length = 1, describing
.saveInterval	numeric	NA	NA	NA	defines the saving time step. If
.studyAreaName	character	NA	NA	NA	Human-readable name for the s
.useCache	character	.inputOb....	NA	NA	Internal. Can be names of event
.useParallel	ANY	2	NA	NA	Used only in seed dispersal. If n

TABLE 1.7: List of *Biomass_core* output objects and their description.

objectName	objectClass	desc
activePixelIndex	integer	internal use. Keeps track of which pixels are active
activePixelIndexReporting	integer	internal use. Keeps track of which pixels are active in t
ANPPMap	RasterLayer	ANPP map at each succession time step (in g /m ²)
cohortData	data.table	'data.table' with cohort-level information on age, biom
ecoregionMap	RasterLayer	map with mapcodes match 'ecoregion' table and 'speci
inactivePixelIndex	logical	internal use. Keeps track of which pixels are inactive
inactivePixelIndexReporting	integer	internal use. Keeps track of which pixels are inactive in
lastFireYear	numeric	Year of the most recent fire year
lastReg	numeric	an internal counter keeping track of when the last rege
minRelativeB	data.frame	define the relative biomass cut points to classify stand
mortalityMap	RasterLayer	map of biomass lost (in g/m ²) at each succession time
pixelGroupMap	RasterLayer	updated community map at each succession time step
regenerationOutput	data.table	If 'P(sim)\$calibrate == TRUE', an summary of seed disp
reproductionMap	RasterLayer	Regeneration map (biomass gains in g/m ²) at each su
simulatedBiomassMap	RasterLayer	Biomass map at each succession time step (in g/m ²)
simulationOutput	data.table	contains simulation results by 'ecoregionGroup' (main
simulationTreeOutput	data.table	Summary of several characteristics about the stands, c
species	data.table	a table that has species traits such as longevity, shade t
speciesEcoregion	data.table	define the maxANPP, maxB and SEP change with both
speciesLayers	RasterStack	species percent cover raster layers, based on input 'spe
spinupOutput	data.table	Spin-up output. Currently deactivated.
summaryBySpecies	data.table	The total species biomass (in g/m ² as in 'cohortData'),
summaryBySpecies1	data.table	No. pixels of each leading vegetation type (used for plo
summaryLandscape	data.table	The averages of total biomass (in tonnes/ha , not g/m ²)
treedFirePixelTableSinceLastDisp	data.table	3 columns: 'pixelIndex', 'pixelGroup', and 'burnTime'. E
vegTypeMap	RasterLayer	Map of leading species in each pixel, colored according

data and calibration modules or by *Biomass_core* itself (during `simInit` and the `init` event²⁵);

2. Disturbances (OPTIONAL) – simulated by a disturbance module (e.g., *LandMine*);

²⁵`simInit` is a `SpaDES` function that initialises the execution of one or more modules by parsing and checking their code and executing the `.inputObjects` function(s), where the developer provides mechanisms to satisfy each module's expected inputs with default values.

3. Post-disturbance mortality/regeneration (OPTIONAL) – simulated by a regeneration module (e.g., *Biomass_regeneration*);
4. Seed dispersal (every `successionTimestep`; `Dispersal` event):
 - seed dispersal can be a slow process and has been adapted to occur every 10 years (default `successionTimestep`). The user can set it to occur more/less often, with the caveat that if using *Biomass_borealDataPrep* to estimate species establishment probabilities, these values are integrated over 10 years.
 - see Scheller and Domingo [14] for details on dispersal algorithms.
5. Growth and mortality (`mortalityAndGrowth` event):
 - unlike dispersal, growth and mortality always occur time step (year).
 - see Scheller and Mladenoff [18] for further detail.
6. Cohort age binning (every `successionTimestep`; `cohortAgeReclassification` event):
 - follows the same frequency as dispersal, collapsing cohorts (i.e., summing their biomass/mortality/aNPP) to ages classes with resolution equal to `successionTimestep`.
 - see Scheller and Miranda [16] for further detail.
7. Summary tables of regeneration (`summaryRegen` event), biomass, age, growth and mortality (`summaryBGM` event);
8. Plots of maps (`plotMaps` event) and averages (`plotAvgs` and `plotSummaryBySpecies` events);
9. Save outputs (`save` event).

... (repeat 2-9) ...

1.2.7 Differences between *Biomass_core* and the LANDIS-II Biomass Succession Extension model (LBSE)

1.2.7.1 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 significantly alter the 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 of other cohorts (i.e., their biomass gain and loss, respectively), 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 within-year sequential 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, 16]. Hence, in *Biomass_core* growth, mortality, recruitment 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 1.8 and Fig. 1.2). The calculation of competition, growth and mortality for all cohorts at the same time also circumvented this issue.

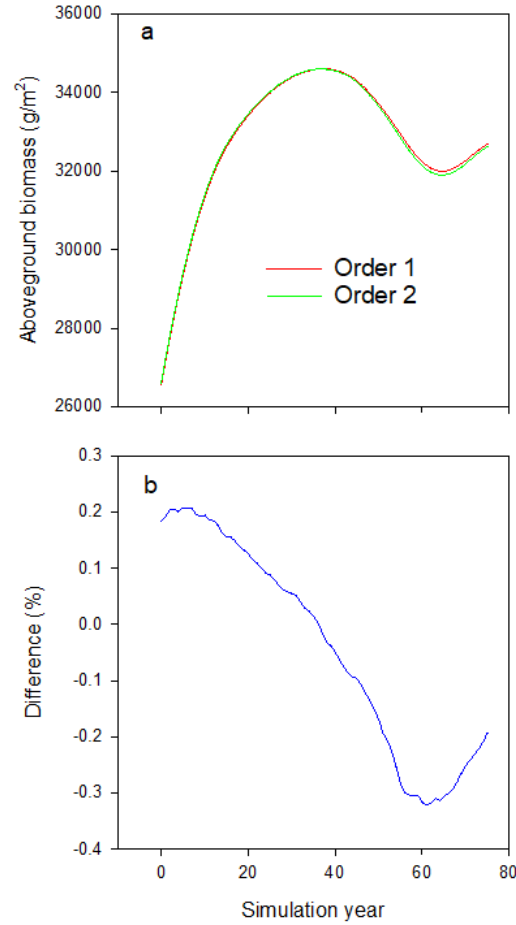


FIGURE 1.2: Differences in total landscape aboveground biomass when using two different input species orders for the same community. These simulations demonstrate how the sequential calculation of the competition index, combined with a lack of explicit species ordering affect the overall landscape aboveground 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 $\frac{Biomass_{order2} - Biomass_{order1}}{Biomass_{order2}} * 100$

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, p. 4], when the succession time step was set to 1. Additionally, together with the lack of explicit ordering, this generated different results in terms of the biomass assigned to each new cohort (e.g., Table 1.9 and Fig. 1.3). 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.

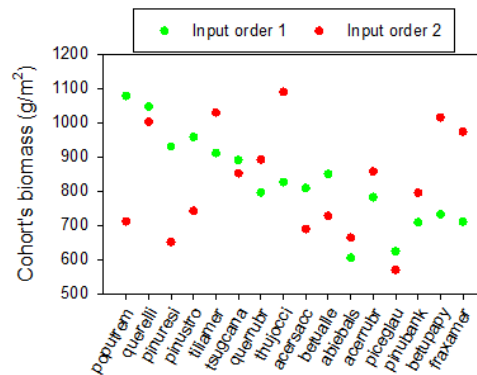


FIGURE 1.3: Differences in the biomass assigned to new cohorts, summed for each species across pixels, when using two different input species orders for the same community and when the succession time step is 1. These simulations demonstrate how the different summation of total cohort biomass for a succession 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. Values refer to the initial total biomass attributed to each species at the end of year 1.

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 pixel 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, since post-disturbance effects were separated background vegetation dynamics simulated by *Biomass_core*.

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 below 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 decreases growth but does not increase mortality, 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).

1.2.7.2 Other enhancements

In addition to the sixth changes in growth, mortality and regeneration mentioned above, we enhanced modularity by separating the components that govern vegetation responses to disturbances from *Biomass_core*, and implemented hashing, caching and testing to improve computational efficiency and insure performance.

1.2.7.2.1 Modularity

Unlike in LBSE, post-disturbance effects are not part of *Biomass_core* *per se*, but belong to two separate modules, used interchangeably (*Biomass_regeneration*²⁶ and *Biomass_regenerationPM*²⁷). 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

²⁶https://github.com/PredictiveEcology/Biomass_regeneration/blob/master/Biomass_regeneration.Rmd

²⁷https://github.com/PredictiveEcology/Biomass_regenerationPM/blob/master/Biomass_regenerationPM.Rmd

the moment). Again, this enables higher flexibility when swapping between different approaches to regeneration.

Climate effects on growth and mortality were also implemented a modular way. The effects of climate on biomass increase (growth) and loss (mortality) were written in functions grouped in two packages. The `LandR` R package contains default, “non-climate-sensitive” functions, while the `LandR.CS` R package contains the functions that simulate climate effects (CS stands for “climate sensitive”). Note that these functions do not simulate actual growth/mortality processes, but estimate modifiers that increase/decrease cohort biomass on top of background growth/mortality. *Biomass_core* uses the `LandR` functions by default (see `growthAndMortalityDrivers` parameter in the [full parameters list](#)). Should the user wish to change how climate effects on growth/mortality are calculated, they can provide new compatible functions (i.e., with the same names, inputs and outputs) via another R package.

1.2.7.2.2 Hashing

Our first strategy to improve simulation efficiency in *Biomass_core* was to use a hashing mechanism [24]. 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 composition (i.e., species, age and biomass composition), and tracked and stored simulation data for each *pixelGroup* (Fig. 1.4). 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.

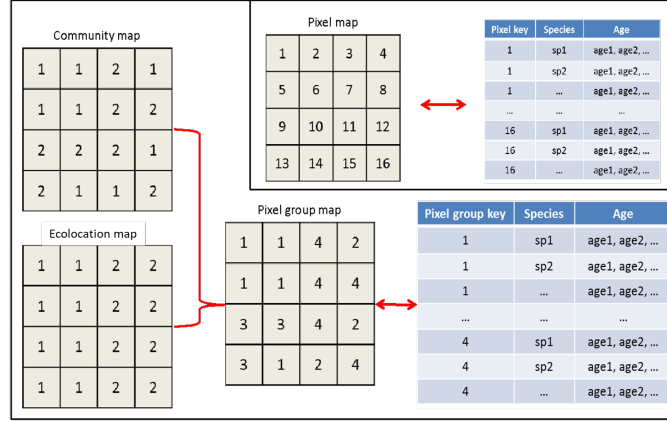


FIGURE 1.4: Hashing design for *Biomass_core*. In the re-coded *Biomass_core*, the pixel group map was hashed based on the unique combination of species composition ('community map') and ecolocation map, and associated with a lookup table. The insert in the top-right corner was the original design that linked the map to the lookup table by pixel key.

1.2.7.2.3 Caching

The second strategy aimed at improving model efficacy was the implementation of caching during 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 and calibration 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, 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.

1.2.7.2.4 Testing

Finally, we implemented code testing to facilitate bug detection by comparing the outputs of functions (etc.) to expected outputs [22]. We built and integrated code tests in *Biomass_core* and across all LandR modules and the LandR R package in the form of assertions, unit tests and integration tests. Assertions and unit tests are run automatically during simulations (but can be turned off) and evaluate individual code components (e.g., one function or an object's class). Integration tests evaluate if several coded processes are integrated correctly and are usually run manually. However, because we embedded assertions within the module code, R package dependencies of *Biomass_core*, such as the LandR R package and SpaDES, they also provide a means to test module integration. We also implemented GitHub Actions continuous integration (CI), which routinely test GitHub hosted packages (e.g., LandR) and modules. CRAN-hosted packages (e.g., SpaDES) are also automatically tested and checked on CRAN.

Finally, because *Biomass_core* (and all other LandR modules) code is hosted in public GitHub repositories, the module code is subject to the scrutiny of many users, who can identify issues and contribute to improve module code.

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

In the recoding of *Biomass_core*, we used integration tests to ensure similar outputs of each demographic process (namely, growth, mortality and recruitment) to the outputs from its counterpart in LBSE. 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 1.10-1.11). The remaining input parameters were taken from a LANDIS-II training course (Tables 1.12-1.15), and contained species attributes information of 16 common tree species in boreal forests and 2 ecoregions. We ran simulations for 1000 years, with a succession time step of 10 and three replicates, 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. 1.5), 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. 1.5 of this appendix).

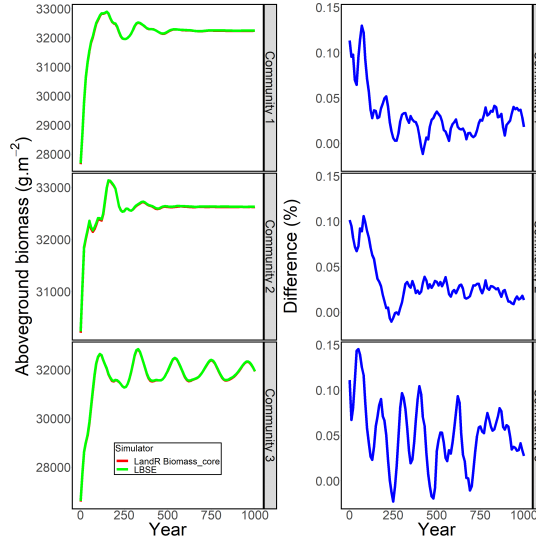


FIGURE 1.5: Visual comparison of simulation outputs for three randomly generated initial communities (left panels) and difference between those outputs (right panels). The % difference between LBSE and *Biomass_core* were calculated as $\frac{Biomass_{LBSE} - Biomass_{Biomass_core}}{Biomass_{LBSE}} * 100$

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 pixels. 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. 1.6a). 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. 1.6a). *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. 1.6b).

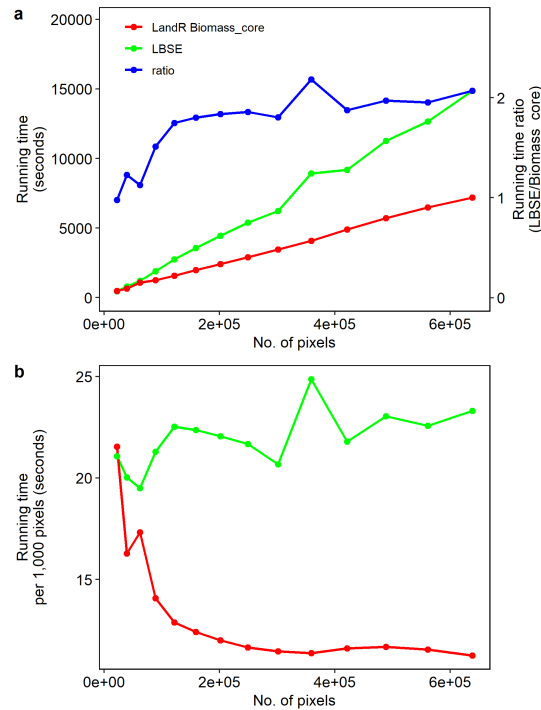


FIGURE 1.6: Simulation efficiencies of LBSE and *Biomass_core* with increasing map size, in terms of a) mean running time across repetitions (left y-axis) and the ratio LBSE to *Biomass_core* running times (right y-axis and blue line), and b) running time scalability as the mean running time per 1000 pixels.

1.3 Usage example

1.3.1 Set up R libraries

```
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 controlled at one level above the source code

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

```

1.3.2 Get the module and module dependencies

We can use the `SpaDES.install::getModule` function to download the module to the module folder specified above. Alternatively, see `SpaDES-modules` repository²⁸ to see how to download this and other `SpaDES` modules, or fork/clone from its GitHub repository²⁹ directly.

After downloading the module, it is important to make sure all module R package dependencies are installed in their correct version. `SpaDES.install::makeSureAllPackagesInstalled` takes care of this for any module in the `paths$modulePath`.

```

SpaDES.install::getModule("PredictiveEcology/Biomass_core",
modulePath = paths$modulePath,
    overwrite = TRUE)

## make sure all necessary packages are installed:
SpaDES.install::makeSureAllPackagesInstalled(paths$modulePath)

```

1.3.3 Setup simulation

Here we setup a simulation in a random study area, using any species within the `LandR::sppEquivalencies_CA` table that can be found there

²⁸<https://github.com/PredictiveEcology/SpaDES-modules>

²⁹https://github.com/PredictiveEcology/Biomass_core/

(*Biomass_core* will retrieve species % cover maps and filter present species). We also define the colour coding used for plotting, the type of plots we want to produce and choose to output cohortData tables every year – note that these are not pixel-based, so to “spatialise” results *a posteriori* the pixel-BrouMap must also be saved.

Please see the lists of **input objects**, **parameters** and **outputs** for more information.

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

studyArea <- Cache(randomStudyArea, size = 1e+07) # cache this
so it creates a random one only once on a machine

# Pick the species you want to work with - using the naming
# convention in 'Boreal' column of
# LandR::sppEquivalencies_CA
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 (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()
```

1.3.4 Run simulation

`simInitAndSpades` is a wrapper function that runs both `simInit` (which initialises all modules) and `spades` (which runs all modules, i.e., their events), to which pass all the necessary setup objects created above.

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

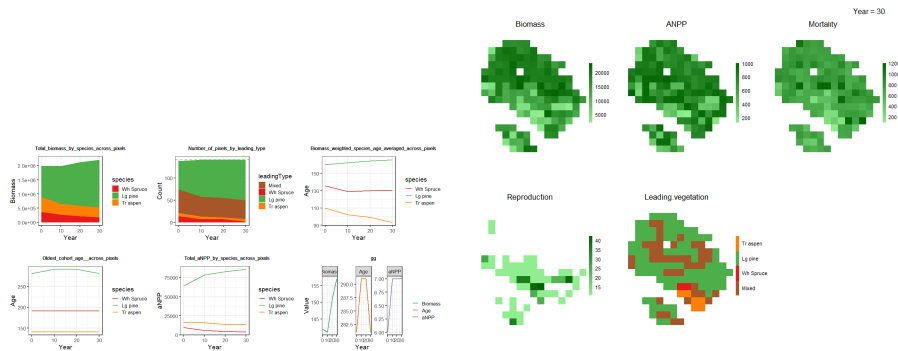


FIGURE 1.7: *Biomass_core* automatically generates simulation visuals of species dynamics across the landscape in terms of total biomass, number of presences and age and productivity (above), as well as yearly plots of total biomass, productivity, mortality, reproduction and leading species in each pixel (below).

1.4 Appendix

1.4.1 Tables

1.5 References

TABLE 1.8: 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	Community	Input order	Age	Processing
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 1.9: 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	Community	Input order	Age	Processing
1	abiebals	1	poputrem	1	pinustro	1	thujocci
1	acerrubr	1	querelli	1	poputrem	1	tiliamer
1	acersacc	1	pinuresi	1	acerrubr	1	querelli
1	betualle	1	pinustro	1	pinubank	1	querrubr
1	betupapy	1	tiliamer	1	betualle	1	betupapy
1	fraxamer	1	tsugcana	1	piceglau	1	fraxamer
1	piceglau	1	querrubr	1	pinuresi	1	tsugcana
1	pinubank	1	thujocci	1	acersacc	1	abiebals
1	pinuresi	1	acersacc	1	querelli	1	acerrubr
1	pinustro	1	betualle	1	querrubr	1	pinubank
1	poputrem	1	abiebals	1	thujocci	1	pinustro
1	querelli	1	acerrubr	1	tiliamer	1	poputrem
1	querrubr	1	piceglau	1	tsugcana	1	pinuresi
1	thujocci	1	pinubank	1	abiebals	1	acersacc
1	tiliamer	1	betupapy	1	betupapy	1	betualle
1	tsugcana	1	fraxamer	1	fraxamer	1	piceglau

TABLE 1.10: 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
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 1.11: 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
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 1.12: Invariant species traits table used in comparison runs.

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

TABLE 1.13: 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 1.14: 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 1.15: 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	0.90	886	26580
1	acerrubr	1.00	1175	35250
1	acersacc	0.82	1106	33180
1	betualle	0.64	1202	36060
1	betupapy	1.00	1202	36060
1	fraxamer	0.18	1202	36060
1	piceglau	0.58	969	29070
1	pinubank	1.00	1130	33900
1	pinuresi	0.56	1017	30510
1	pinustro	0.72	1090	38150
1	poputrem	1.00	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

LandR Data and Calibration Modules

The LandR ecosystem of SpaDES modules has a variety of data and/or calibration 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.



3

LandR Biomass_speciesData Module

Biomass speciesData v. 1.0.0 ¹

Get help Report issues ²

3.0.0.1 Authors:

Ceres Barros ceres.barros@ubc.ca³ [aut, cre], Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca⁴ [aut], Alex M. Chubaty achubaty@for-cast.ca⁵ [aut]

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

3.1 Module Overview

3.1.1 Quick links

- [General functioning](#)
- [List of input objects](#)
- [List of parameters](#)
- [List of outputs](#)
- [Simulation flow and module events](#)

¹https://github.com/PredictiveEcology/Biomass_speciesData.git

²https://github.com/PredictiveEcology/Biomass_speciesData/issues

³<mailto:ceres.barros@ubc.ca>

⁴<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

⁵<mailto:achubaty@for-cast.ca>

3.1.2 Module summary

LandR *Biomass_speciesData* (hereafter *Biomass_speciesData*) downloads and pre-processes species percent (% cover) data layers used by other LandR data modules (e.g., *Biomass_borealDataPrep*) and by the LandR forest simulation module *Biomass_core*.

3.1.3 Links to other modules

Biomass_speciesData is intended to be used with any LandR modules that require species % cover raster layers (see examples below). See here⁶ for all available modules in the LandR ecosystem and select *Biomass_speciesData* from the drop-down menu to see potential linkages.

- *Biomass_borealDataPrep*⁷: prepares all parameters and inputs (including initial landscape conditions) that *Biomass_core* needs to run a realistic simulation. Default values/inputs produced are relevant for boreal forests of Western Canada. Used downstream from *Biomass_speciesData*;
- *Biomass_core*⁸: core forest dynamics simulation module. Used downstream from *Biomass_speciesData*.

3.2 Module manual

3.2.1 General functioning

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

Currently, the module can access the Canadian National Forest Inventory (NFI) forest attributes kNN dataset [the default; Beaudoin et al. [2]], the

⁶https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

⁷https://github.com/PredictiveEcology/Biomass_borealDataPrep

⁸https://github.com/PredictiveEcology/Biomass_core

Common Attribute Schema for Forest Resource Inventories dataset [CAS-FRI; Cosco [6]], 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** and access to the other datasets must be granted by module developers and data owners, and requires a Google account. 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 the [list of parameters](#))

When multiple data sources are used, the module will replace lower quality data with higher quality data following the order specified in the `types` parameter.

When multiple species of a given data source are to be grouped, % cover is summed across species of the same group within each pixel. Please see the `sppEquiv` in the [list of input objects](#) for information on how to define species groups.

The module can also exclude species % cover layers if they don't have a minimum % cover value in at least one pixel. The user should still inspect where species is deemed present (e.g., in how many pixels in total), as it is possible that some datasets only have a few pixels where the species is present, but with reported high % cover. 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. 3.1).

3.2.2 List of input objects

Below is the full list of input objects that `Biomass_speciesData` requires (Table 3.2). Of these, the only input that **must** be provided (i.e., `Biomass_speciesData` does not have a default for) is `studyAreaLarge`.

Of the inputs in Table 3.2, the following are particularly important and deserve special attention:

- `studyAreaLarge` – the polygon defining the area for which species cover data are desired. It can be larger (but never smaller) than the study area used in the simulation of forest dynamics (i.e., `studyArea` object in `Biomass_core`), in which case it should fully cover it.
- `sppEquiv` – a table of correspondences between different species naming

TABLE 3.1: 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 spp.* as a genus-level group. For this, all six species are separately 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 spp.* 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
* <i>Abies balsamea</i> *	Abie_Bal	Abie_Bal	* <i>Abies balsamea</i> *
* <i>Abies lasiocarpa</i> *	Abie_Las	Abie_Las	* <i>Abies lasiocarpa</i> *
* <i>Picea engelmannii</i> *	Pice_Eng	Pice_Spp	* <i>Picea spp.</i> *
* <i>Picea glauca</i> *	Pice_Gla	Pice_Spp	* <i>Picea spp.</i> *
* <i>Picea mariana</i> *	Pice_Mar	Pice_Spp	* <i>Picea spp.</i> *
* <i>Pinus contorta</i> *	Pinu_Con	Pinu_Con	* <i>Pinus contorta</i> *

TABLE 3.2: List of *Biomass_speciesData* input objects and their description.

objectName	objectClass	desc
rasterToMatchLarge	RasterLayer	a raster of 'studyAreaLarge' in the same resolution and projection
sppColorVect	character	A named vector of colors to use for plotting. The names are the species names
sppEquiv	data.table	table of species equivalencies. See 'LandR::sppEquiv' for details
studyAreaLarge	SpatialPolygonsDataFrame	Polygon to use as the parametrisation study area. Must be a valid polygon
studyAreaReporting	SpatialPolygonsDataFrame	multipolygon (typically smaller/unbuffered than 'studyAreaLarge')

conventions. This table is used across several LandR modules, including *Biomass_core*. It is particularly important here because it will determine whether and which species (and their cover layers) are merged. 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 National Forest Inventory), but the same name (e.g., "Pice_Spp") in the column chosen for the naming convention used throughout the simulation (defined by the `sppEquivCol` parameter). See Table 3.1 for an example.

TABLE 3.3: 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
dataYear	numeric	2001	NA	NA	Passed to 'paste0('prepSpeciesLayers_', type
sppEquivCol	character	Boreal	NA	NA	The column in 'sim\$sppEquiv' data.table to
types	character	KNN	NA	NA	The possible data sources. These must corre
vegLeadingProportion	numeric	0.8	0	1	a number that defines whether a species is
.plotInitialTime	numeric	NA	NA	NA	This describes the simulation time at which
.plotInterval	numeric	NA	NA	NA	This describes the simulation time interval
.saveInitialTime	numeric	NA	NA	NA	This describes the simulation time at which
.saveInterval	numeric	NA	NA	NA	This describes the simulation time interval
.studyAreaName	character	NA	NA	NA	Human-readable name for the study area u
.useCache	logical	init	NA	NA	Controls cache; caches the init event by def
.useParallel	numeric	16	NA	NA	Used in reading csv file with fread. Will be

3.2.3 List of parameters

Table 3.3 lists all parameters used in *Biomass_speciesData* and their detailed information. All these parameters have default values specified in the module's metadata.

Of these parameters, 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 (*speciesLayers*). Note that this will affect what species have data for an eventual simulation and the user will need to adjust simulation parameters accordingly (e.g., species in trait tables will need to match the species in *speciesLayers*).
- **types** – character. Which % cover data sources are to be used (see **General functioning**). 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 in *types*. For instance, if *types* == c("KNN", "CASFRI", "ForestInventory"), KNN is assumed to be the lowest quality data set and ForestInventory the highest, hence 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.

TABLE 3.4: List of *Biomass_speciesData* output objects and their description.

objectName	objectClass	desc
speciesLayers	RasterStack	biomass percentage raster layers by species in Canada species map
treed	data.table	Table with one logical column for each species, indicating whether there w
numTreed	numeric	a named vector with number of pixels with non-zero cover values for each
nonZeroCover	numeric	A single value indicating how many pixels have non-zero cover

3.2.4 List of outputs

The module produces the outputs in Table 3.4, and automatically saves the processed species cover layers in the output path defined in `getPaths(sim)$outputPath`.

3.2.5 Simulation flow and module events

Biomass_speciesData initialises 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")`⁹.

The module defaults to processing cover data for 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 3.5; see `?LandR::sppEquivalencies_CA` for more information):

Biomass_speciesData only runs two events, the `init` event where all species cover layers are processed and a plotting event (`initPlot`) that plots the final layers.

The general flow of *Biomass_speciesData* processes is:

1. Download (if necessary) 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 \geq `coverThresh` are excluded.

⁹Raw data layers downloaded by the module are saved in `'dataPath(sim)'`, which can be controlled via `'options(reproducible.destinationPath = ...)'`.

TABLE 3.5: List of species cover data downloaded by default by *Biomass_speciesData*.

Species	Generic name
* <i>Abies balsamea</i> *	Balsam Fir
* <i>Abies lasiocarpa</i> *	Fir
* <i>Acer negundo</i> *	Boxelder maple
* <i>Acer pensylvanicum</i> *	Striped maple
* <i>Acer saccharum</i> *	Sugar maple
* <i>Acer spicatum</i> *	Mountain maple
* <i>Acer spp.</i> *	Maple
* <i>Alnus spp</i> *	Alder
* <i>Betula alleghaniensis</i> *	Swamp birch
* <i>Betula papyrifera</i> *	Paper birch
* <i>Betula populifolia</i> *	Gray birch
* <i>Betula spp.</i> *	Birch
* <i>Fagus grandifolia</i> *	American beech
* <i>Fraxinus americana</i> *	American ash
* <i>Fraxinus nigra</i> *	Black ash
* <i>Larix laricina</i> *	Tamarack
* <i>Larix lyallii</i> *	Alpine larch
* <i>Larix occidentalis</i> *	Western larch
* <i>Larix spp.</i> *	Larch
* <i>Picea engelmannii</i> *	Engelmann's spruce
* <i>Picea glauca</i> *	White.Spruce
* <i>Picea mariana</i> *	Black.Spruce
* <i>Picea spp.</i> *	Spruce
* <i>Pinus albicaulis</i> *	Whitebark pine
* <i>Pinus banksiana</i> *	Jack pine
* <i>Pinus contorta</i> *	Lodgepole pine
* <i>Pinus monticola</i> *	Western white pine
* <i>Pinus resinosa</i> *	Red pine
* <i>Pinus spp.</i> *	Pine
* <i>Populus balsamifera v. balsamifera</i> *	Balsam poplar
* <i>Populus trichocarpa</i> *	Black cottonwood
* <i>Populus grandidentata</i> *	White poplar
* <i>Populus spp.</i> *	Poplar
* <i>Populus tremuloides</i> *	Trembling poplar
* <i>Tsuga canadensis</i> *	Eastern hemlock
* <i>Tsuga spp.</i> *	Hemlock

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 replace 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 (`initPlot` event). A summary of number of pixels with forest cover are calculated (`treedand numTreed` output objects; see [list of outputs](#)).

3.3 Usage example

This module can be run stand-alone, but it only compiles species % cover data into layers used by other modules.

3.3.1 Load spaDES and other packages.

```
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))
```

3.3.2 Get module, necessary packages and set up folder directories


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

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

## make sure all necessary packages are installed:
makeSureAllPackagesInstalled(paths$modulePath)
```

3.3.3 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.

```
# 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 user:
# options(reproducible.inputPaths =
# 'E:/Data/LandR_related/') # to re-use datasets across
# projects
studyAreaLarge <- Cache(randomStudyArea, size = 1e+07, cacheRepo
= paths$cachePath) # cache this so it creates a random one only
once on a machine

# Pick the species you want to work with -- here we use the
# naming convention in 'Boreal' column of
# LandR::sppEquivalencies_CA (default)
```

```

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))

```

3.3.4 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.

```

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 outputs of *Biomass_speciesData* (dominant species) in a randomly generated study area within Canada.

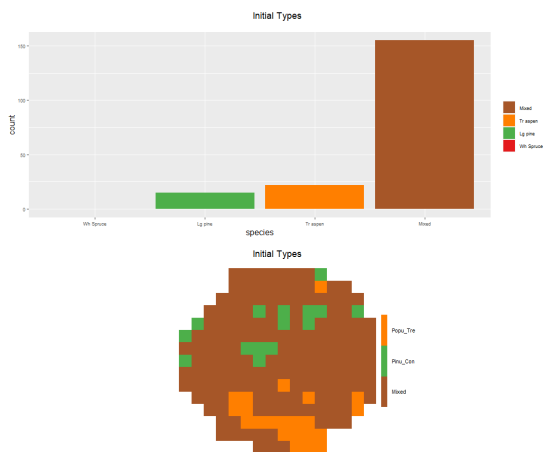


FIGURE 3.1: *Biomass_speciesData* automatically generates a plot of species dominance and number of presences in the study area when ‘.plotInitialTime=1’ is passed as an argument.



4

LandR Biomass_borealDataPrep Module

Biomass borealDataPrep v. 1.5.4 ¹

Get help Report issues ²

4.0.0.1 Authors:

Yong Luo yong.luo@canada.ca³ [aut], Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca⁴ [aut, cre], Ceres Barros ceres.barros@ubc.ca⁵ [aut], Alex M. Chubaty achubaty@for-cast.ca⁶ [aut]

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

4.1 Module Overview

4.1.1 Quick links

- General functioning
- List of input objects
- List of parameters
- List of outputs
- Simulation flow and module events

¹https://github.com/PredictiveEcology/Biomass_borealDataPrep.git

²https://github.com/PredictiveEcology/Biomass_borealDataPrep/issues

³<mailto:yong.luo@canada.ca>

⁴<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

⁵<mailto:ceres.barros@ubc.ca>

⁶<mailto:achubaty@for-cast.ca>

4.1.2 Summary

LandR *Biomass_borealDataPrep* (hereafter *Biomass_borealDataPrep*), prepares all necessary inputs for *Biomass_core* based on data available for forests across Canada forests, but focused on Western Canada boreal forest systems. Nevertheless, it provides a good foundation to develop other modules aimed at different geographical contexts. By keeping data preparation and parameter estimation outside of *Biomass_core*, we promote the modularity of the LandR-based model systems and facilitate interoperability with other parameter estimation procedures.

Specifically, it prepares and adjusts invariant and spatially varying species trait values, as well as ecolocation-specific parameters, probabilities of germination and initial conditions necessary to run *Biomass_core*. For this, *Biomass_borealDataPrep* requires internet access to retrieve default data⁷.

We advise future users to run *Biomass_borealDataPrep* with defaults and inspect what the input objects are like before supplying alternative data (or data URLs).

4.1.3 Links to other modules

Biomass_borealDataPrep is intended to be used with *Biomass_core*⁸, but can be linked with other data modules that prepare inputs. See here⁹ for all available modules in the LandR ecosystem and select *Biomass_borealDataPrep* from the drop-down menu to see potential linkages.

- *Biomass_core*¹⁰: core forest dynamics simulation module. Used downstream from *Biomass_borealDataPrep*;
- *Biomass_speciesData*¹¹: grabs and merges several sources of species cover data, making species percent cover (% cover) layers used by other LandR Biomass modules. Default source data spans the entire Canadian territory. Used upstream from *Biomass_borealDataPrep*;
- *Biomass_speciesParameters*¹²: calibrates four-species level traits using perma-

⁷Raw data layers downloaded by the module are saved in 'dataPath(sim)', which can be controlled via 'options(reproducible.destinationPath = ...)'.
⁸https://github.com/PredictiveEcology/Biomass_core
⁹https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem
¹⁰https://github.com/PredictiveEcology/Biomass_core
¹¹https://github.com/PredictiveEcology/Biomass_speciesData
¹²https://github.com/PredictiveEcology/Biomass_speciesParameters

nent sample plot data (i.e., repeated tree biomass measurements) across Western Canada. Used downstream from *Biomass_borealDataPrep*.

4.2 Module manual

4.2.1 General functioning

Biomass_borealDataPrep prepares all inputs necessary to run a realistic simulation of forest dynamics in Western Canadian 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:

1. **calculates species biomass per pixel** by multiplying the observed species % cover by the observed stand biomass and an adjustment factor, which can be statistically calibrated for the study area. 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 (see [Initial species age and biomass per pixel](#) and [Adjustment of species biomass](#));
2. prepares **invariant species traits** – these are spatio-temporally constant species traits that influence population dynamics (e.g., growth, mortality, dispersal) and responses to fire (see [Invariant species traits](#));
3. defines **ecolocations** – groupings of pixels with similar biophysical conditions. By default, ecolocations are defined as the spatial combination of ecodistricts of the National Ecological Framework for Canada, and the Land Cover of Canada 2010 map (see [Defining simulation pixels and ecolocations](#)). **Note that ecolocations are called `ecoregionGroup`'s across `LandR` modules.**
4. prepares **ecolocation-specific parameters** and **probabilities of germination** – only one ecolocation-specific parameter is used, the [minimum relative biomass thresholds](#), which defines the level

- of shade in a pixel. Together the level of shade and the **probabilities of germination** influence germination success in any given pixel;
5. estimates **spatio-temporally varying species traits** – species traits that can vary by ecolocation and in time. These are maximum biomass (`maxB`), maximum above-ground net primary productivity (`maxANPP`; see **Maximum biomass and maximum aboveground net primary productivity**) and species establishment probability (`SEP`, called `establishprob` in the module traits table; see **Species establishment probability**). **By default, *Biomass_borealDataPrep* estimates temporally constant values of `maxB`, `maxANPP` and `SEP`;**
 6. creates **initial landscape conditions** – *Biomass_borealDataPrep* performs data-based landscape initialisation, by creating the species cohort table (`cohortData`) and corresponding map (`pixelGroupMap`; both used to initialise and track cohorts across the landscape) based on observed stand age and species biomass (after data clean-up and dealing with mismatches).

As *Biomass_core* only simulates tree species dynamics, *Biomass_borealDataPrep* prepares all inputs and estimates parameters in pixels within forested land-cover classes (see **Defining simulation pixels and ecolocations**).

If a `studyAreaLarge` is supplied, the module uses it for parameter estimation to account for larger spatial variability.

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

4.2.2 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* 23] at the national-scale.

If no other inputs are supplied, *Biomass_borealDataPrep* will create raster layer

versions `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).

4.2.2.1 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 of Canada 2010 v1 map¹³, a raster-based database that distinguishes several forest and non-forest land-cover types. Pixels with classes 1 to 6 are included as forested pixels (see parameter `forestedLCCCclasses`).

When the land-cover raster (`rstLCC`) includes transient cover types (e.g., recent burns) the user may pass a vector of transient class IDs (via the parameter `LCCCclassesToReplaceNN`) that will be reclassified into a “stable” forested class (defined via the parameter `forestedLCCCclasses`). 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 Canada¹⁴ (a broad-scale polygon layer that captures sub-regional variation) (`ecoregionLayer`) and the above land cover (`rstLCC`), but the user can change this by supplying different ecozonation or land-cover layers.

4.2.2.2 Species cover

Species percent cover (% cover) raster layers (`speciesLayers`) can be automatically obtained and pre-processed by *Biomass_borealDataPrep*. The module ensures that:

¹³<http://www.cec.org/north-american-environmental-atlas/land-cover-2010-modis-250m/>

¹⁴<https://open.canada.ca/data/en/dataset/3ef8e8a9-8d05-4fea-a8bf-7f5023d2b6e1>

1. all data have the same geospatial properties (extent, resolution);
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 [2] – hereafter ‘kNN species data’.

4.2.2.3 Initial species age and biomass per pixel

Stand age and aboveground stand biomass (hereafter ‘stand biomass’) are used to derive parameters and define initial species age and biomass across the landscape. These are also derived from MODIS satellite imagery from 2001 prepared by the NFI [2] 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 initial 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 Wild-fire Data Base¹⁵ 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 where 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 biomass is inflated in pixels with recent burnts. 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`.

4.2.2.4 Invariant species traits

Most invariant species traits are obtained from available species trait tables used in LANDIS-II applications in Canada's boreal forests (available in Dominic Cyr's GitHub repository¹⁶). Some are then adapted with minor adjustments to match Western Canadian boreal forests using published literature. Others (key growth and mortality traits) can be calibrated by *Biomass_speciesParameters* (see [Calibrating species growth/mortality traits using Biomass_speciesParameters](#)).

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

¹⁵<https://cwfis.cfs.nrcan.gc.ca/datamart>

¹⁶https://github.com/dcy/LANDIS-II_IA_generalUseFiles

`P(sim)$speciesTableAreas`). Most trait values do not vary across these ecozones for a given species, but when they do 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 engelmannii*, *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`).

4.2.2.5 Probabilities of germination

By default, *Biomass_borealDataPrep* uses the same probabilities of germination (called `sufficientLight` in the module) as *Biomass_core*. These are obtained from publicly available LANDIS-II table¹⁷.

4.2.3 Parameter estimation/calibration

4.2.3.1 Adjustment of initial species biomass

Biomass_borealDataPrep estimates initial values of species aboveground biomass (`B`) 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 Northwest Territories

¹⁷https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt

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'), *speciesCode* (i.e. species ID) and land cover ('lcc'). 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.

4.2.3.2 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 biomass (*B*) to species age and cover, while accounting for variation between 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 *r*²) and by the entire model (conditional *r*²) – both of which are printed as messages.

Because the model can take a while to fit, by default we sample pixels within each species and ecolocation combination (sample size defined by the `P(sim)$subsetDataBiomassModel` parameter).

If convergence issues occur and `P(sim)$fixModelBiomass == TRUE`, the module attempts to refit the model by re-sampling the data, re-fitting *lmer* with the *bobyqa* optimizer, and re-scaling the continuous predictors (by default, *cover* and *logAge*). These steps are tried additively until the convergence issue is resolved. If the module is still unable to solve the converge issue a message is printed and the module uses the last fitted model.

Note that convergence issues are not usually problematic for the estimation of coefficient values, only for estimation of their standard errors. However, the user should always inspect the final model (especially if not converged) and make sure that the problems are not significant and that the fitted model meets residual assumptions. For this, the user should make sure model objects are exported to the `simList` using the `exportModels` parameter.

Alternative model calls/formulas 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).

Another consideration to add with respect to the estimation of `maxB`, is that we are choosing a linear model to relate $B \sim \log(\text{age}) + \text{cover}$. This is not ideal from an ecological point of view, as biomass is unlikely to vary linearly with age or cover, and more likely to saturate beyond a certain high value of cover and follow a hump-shaped curve with age (i.e., reaching maximum values for a given age, and then starting to decrease as trees approach longevity). Also, fitting a linear model can lead to negative B values at young ages and low cover. However, our tests revealed that a linear mixed effects model was not producing abnormal estimates of B at maximum values of age and cover (hence, `maxB` estimates), while allowing to leverage on the powerful statistical machinery of `lme4`.

Finally, we highlight that modelling $\log(B)$ is NOT an appropriate solution, because it will wrongly assume an *exponential* relationship between $B \sim \log(\text{age}) + \text{cover}$, leading to a serious overestimation of `maxB` (Fig. 4.1) and steep increases in species biomasses during the first years of the simulation (Fig. 4.2).

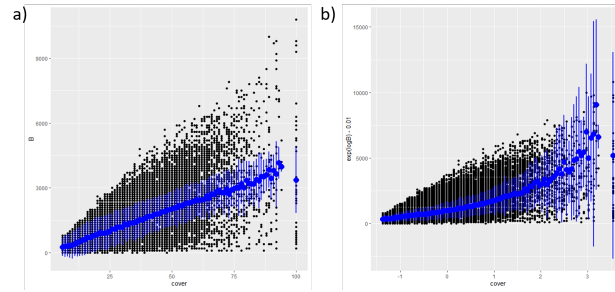


FIGURE 4.1: Modelling biomass as a linear vs. exponential relationship. a) ‘modelBiomass’ as ‘B ~ logAge * speciesCode + cover * speciesCode + (logAge + cover | ecoregionGroup)’. b) ‘modelBiomass’ as ‘logB ~ logAge * speciesCode + cover * speciesCode + (logAge + cover | ecoregionGroup)’. Blue dots are marginal mean B values (back-transformed in b) cross ages with confidence intervals as the bars.

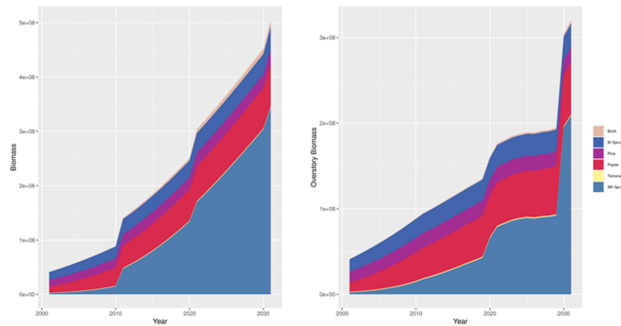


FIGURE 4.2: Thirty years of simulation with ‘maxB’ values estimated from a ‘logB ...’ ‘biomassModel’ (see Fig. reffig:fig-biomassModelLogBtest). The steep increase in such little time is abnormal.

After the biomass model is fit, maxB is predicted by species and ecolocation combination, for maximum species cover values (100%) and maximum log-age (the log of species longevity). When using *Biomass_speciesParameters*, maxB is calibrated so that species can achieve the maximum observed biomass during the simulation (see [Calibrating species growth/mortality traits using Biomass_speciesParameters](#)).

maxANPP is calculated as $\text{maxB} * \text{mANPPproportion} / 100$, where mANPPproportion defaults to 3.33, unless calibrated by *Biomass_speciesParameters* (see

Calibrating species growth/mortality traits using *Biomass_speciesParameters*). The default value, 3.33, comes from an inversion of the rationale used to calculate maxB in Scheller and Mladenoff [18]. There, the authors estimated maxANPP using the model PnET-II (and then adjusted the values manually) and from these estimates calculated maxB by multiplying the estimated maxANPP by 30.

4.2.3.3 Species establishment probability

Species establishment probability (SEP, `establishprob` in the module) 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 following a binomial distribution with a logit link function. 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 and predicted values are integrated over the length of the succession time step (`successionTimestep` parameter) as:

$$\text{integratedSEP} = 1 - (1 - \text{estimatedSEP})^{e^{\text{successionTimestep}}} \quad (4.1)$$

This is important, since seed establishment only occurs once at every `P(sim)$successionTimestep`, and thus the probabilities of seed establishment need to be temporally integrated to reflect the probability of a seed establishing in this period of time.

Finally, since the *observed* species cover used to fit `coverModel` is a result of both seed establishment and resprouting/clonal growth, the final species establishment probabilities are calculated as a function of the temporally integrated probabilities and species' probabilities of resprouting (`resproutprob`, in the species table) (bounded between 0 and 1):

$$SEP = integratedSEP * (1 - resproutprob) \quad (4.2)$$

if $SEP > 1$, then:

$$SEP = 1 \quad (4.3)$$

if $SEP < 0$, then:

$$SEP = 0 \quad (4.4)$$

4.2.3.4 Ecolocation-specific parameter – minimum relative biomass

Minimum relative biomass (`minRelativeB`) is a spatially-varying parameter used to determine the shade level in each pixel. Each shade class (X0-X5) is defined by a minimum relative biomass threshold compared to the pixel's current relative biomass, which is calculated as the sum of pixel's total biomass divided by the total potential maximum biomass in that pixel (the sum of all `maxB` for the pixel's ecolocation).

Since we found no data to base the parametrisation of the shade class thresholds, default values are based on publicly available values used in LANDIS-II applications in Canada's boreal forests (available in Dominic Cyr's GitHub repository¹⁸), and all ecolocations share 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 adjusted 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], which are likely driven by higher moisture limitation in the west [7, 13].

This adjustment can be bypassed by either supplying a `minRelativeB` table, or an alternative function call to `P(sim)$minRelativeBFunction` (which by default is `LandR::makeMinRelativeB`).

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

¹⁸https://github.com/dcyrr/LANDIS-II_IA_generalUseFiles

4.2.3.5 Calibrating species growth/mortality traits using *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 by *Biomass_borealDataPrep* (see [Maximum biomass and maximum above-ground 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 a new species trait, *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 second new species trait, *inflationFactor*, and recalibrates *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*.

TABLE 4.1: 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 spp.* as a genus-level group. 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 spp.* have the same name. *Biomass_borealDataPrep* will merge their % cover data into a single layer by summing their cover per pixel.

Species	KNN	Boreal	Modelled as
* <i>Abies balsamea</i> *	Abie_Bal	Abie_Bal	* <i>Abies balsamea</i> *
* <i>Abies lasiocarpa</i> *	Abie_Las	Abie_Las	* <i>Abies lasiocarpa</i> *
* <i>Picea engelmannii</i> x <i>glauca</i> *		Pice_Eng_Gla	* <i>Picea engelmannii</i> x <i>glauca</i> *
* <i>Picea engelmannii</i> x <i>glauca</i> *		Pice_Eng_Gla	* <i>Picea engelmannii</i> x <i>glauca</i> *
* <i>Picea engelmannii</i> *	Pice_Eng	Pice_Spp	* <i>Picea spp.</i> *
* <i>Picea glauca</i> *	Pice_Gla	Pice_Spp	* <i>Picea spp.</i> *
* <i>Picea mariana</i> *	Pice_Mar	Pice_Spp	* <i>Picea spp.</i> *
* <i>Pinus contorta</i> var. <i>contorta</i> *		Pinu_Con	* <i>Pinus contorta</i> var. <i>contorta</i> *
* <i>Pinus contorta</i> *	Pinu_Con	Pinu_Con	* <i>Pinus contorta</i> *

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

In cases where there is insufficient 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**) and inflationFactor and manPPproportion take default values of 1 and 3.33.

4.2.4 Agregating species

Biomass_borealDataPrep will use the input table `sppEquiv` and the parameter `P(sim)$sppEquivCol` to know what species identities will be used for the simulation (see full list of **input objects** and **parameters** for details). The user can use this table and parameter to define grouping that “merge” species that have their own invariant trait values (see **Invariant species traits**) (e.g. genus-level group or a functional group). To do so, the user must repeat the name of the species group in `sppEquivCol` column of the `sppEquiv` table as many times as the species being grouped:

When groups contain species with different (invariant) trait values, the minimum value across all species is used. As for the default species % cover layers,

Biomass_borealDataPrep proceeds in the same way as *Biomass_speciesData* and sums cover across species of the same group per pixel.

4.2.5 List of input objects

Below are the full lists of input objects (Table 4.2) 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.

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):

TABLE 4.2: List of *Biomass_borealDataPrep* input objects and their description.

objectName	objectClass	desc
cloudFolderID	character	The google drive location where cloudCache will s
columnsForPixelGroups	character	The names of the columns in 'cohortData' that de
ecoregionLayer	SpatialPolygonsDataFrame	A 'SpatialPolygonsDataFrame' that characterizes
ecoregionRst	RasterLayer	A raster that characterizes the unique ecological
rstLCC	RasterLayer	A land classification map in study area. It must b
rasterToMatch	RasterLayer	A raster of the 'studyArea' in the same resolution
rasterToMatchLarge	RasterLayer	A raster of the 'studyAreaLarge' in the same resol
rawBiomassMap	RasterLayer	total biomass raster layer in study area. Defaults
speciesLayers	RasterStack	cover percentage raster layers by species in Cana
speciesTable	data.table	a table of invariant species traits with the followi
sppColorVect	character	named character vector of hex colour codes corre
sppEquiv	data.table	table of species equivalencies. See '?LandR::sppE
sppNameVector	character	an optional vector of species names to be pulled f
standAgeMap	RasterLayer	stand age map in study area. Defaults to the Can
studyArea	SpatialPolygonsDataFrame	Polygon to use as the study area. Must be supplie
studyAreaLarge	SpatialPolygonsDataFrame	multipolygon (potentially larger than 'studyArea')

“species”, “Area”, “longevity”, “sexualmature”, “shadetolerance”, “fire-tolerance”, “seeddistance_eff”, “seeddistance_max”, “resproutprob”, “resproutage_min”, “resproutage_max”, “postfireregen”, “leaflongevity”, “wooddecayrate”, “mortalityshape”, “growthcurve”, “leafLignin”, “hard-soft”. The columns names can be different but not their order. See Scheller and Miranda [17] and *Biomass_core* manual for further detail about these columns.

4.2.6 List of parameters

Table 4.3 lists all parameters used in *Biomass_borealDataPrep* and their detailed information. All have default values specified in the module's meta-data.

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 subsampling 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).

4.2.7 List of outputs

The module produces the following outputs (Table 4.4), which are key inputs of *Biomass_core*.

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*.

TABLE 4.3: List of *Biomass_borealDataPrep* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
biomassModel	call	lme4::lm....	NA	NA	Model and formula for esti
coverModel	call	glm, cbi....	NA	NA	Model and formula used fo
coverPctToBiomassPctModel	call	glm, I(l....	NA	NA	Model to estimate the relat
deciduousCoverDiscount	numeric	0.8418911	NA	NA	This was estimated with da
fitDeciduousCoverDiscount	logical	FALSE	NA	NA	If TRUE, this will re-estim
dataYear	numeric	2001	NA	NA	Used to override the defau
ecoregionLayerField	character		NA	NA	the name of the field used
exportModels	character	none	NA	NA	Controls whether models u
fireURL	character	https://....	NA	NA	A URL to a fire database, s
fixModelBiomass	logical	FALSE	NA	NA	should 'modelBiomass' be
forestedLCCClasses	numeric	1, 2, 3,....	0	NA	The classes in the 'rstLCC'
imputeBadAgeModel	call	lme4::lm....	NA	NA	Model and formula used fo
LCCClassesToReplaceNN	numeric		NA	NA	This will replace these clas
minCoverThreshold	numeric	5	0	100	Pixels with total cover that
minRelativeBFunction	call	LandR::m....	NA	NA	A quoted function that ma
omitNonTreedPixels	logical	TRUE	FALSE	TRUE	Should this module use on
overrideBiomassInFires	logical	TRUE	NA	NA	should B values be re-estim
pixelGroupAgeClass	numeric	params(s....	NA	NA	When assigning 'pixelGrou
pixelGroupBiomassClass	numeric	100	NA	NA	When assigning pixelGrou
rmImputedPix	logical	FALSE	NA	NA	Should 'sim\$imputedPixII
speciesUpdateFunction	list	LandR::s....	NA	NA	Unnamed list of (one or m
sppEquivCol	character	Boreal	NA	NA	The column in 'sim\$specie
speciesTableAreas	character	BSW, BP, MC	NA	NA	One or more of the Ecopro
subsetDataAgeModel	numeric	50	NA	NA	the number of samples to
subsetDataBiomassModel	numeric		NA	NA	the number of samples to
successionTimestep	numeric	10	NA	NA	defines the simulation tim
useCloudCacheForStats	logical	TRUE	NA	NA	Some of the statistical mo
.plotInitialTime	numeric	start(sim)	NA	NA	This is here for backwards
.plots	character	NA	NA	NA	This describes the type of
.plotInterval	numeric	NA	NA	NA	This describes the simulati
.saveInitialTime	numeric	NA	NA	NA	This describes the simulati
.saveInterval	numeric	NA	NA	NA	This describes the simulati
.seed	list		NA	NA	Named list of seeds to use
.studyAreaName	character	NA	NA	NA	Human-readable name for
.useCache	character	.inputOb....	NA	NA	Internal. Can be names of

- `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. XO-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.

4.2.8 Simulation flow and module events

Biomass_borealDataPrep initialises 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*.

The module runs only for one time step and contains 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);

TABLE 4.4: List of *Biomass_borealDataPrep* output objects and their description.

objectName	objectClass
biomassMap	RasterLayer
cohortData	data.table
ecoregion	data.table
ecoregionMap	RasterLayer
imputedPixID	integer
pixelGroupMap	RasterLayer
pixelFateDT	data.table
minRelativeB	data.frame
modelCover	data.frame
modelBiomass	data.frame
rawBiomassMap	RasterLayer
species	data.table
speciesEcoregion	data.table
studyArea	SpatialPolygonsDataFrame
sufficientLight	data.frame

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.

[steps 1-9 are part of the `init` event. Before step 1, the data is downloaded when during the run of the `.inputObjects` function]

10. (OPTIONAL) Plots of `maxB`, `maxANPP` and `SEP` maps (`plot` event);
11. (OPTIONAL) Save outputs (`save` event)

4.3 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 repository¹⁹ and in **barros**.

4.4 References

¹⁹https://github.com/CeresBarros/LandRBiomass_publication

5

LandR Biomass_speciesParameters Module

Biomass speciesParameters v. 1.0.0¹

Get help Report issues²

5.0.0.1 Authors:

Ian Eddy ian.eddy@nrcan-rncan.gc.ca³ [aut, cre], Eliot McIntire eliot.mcintire@nrcan-rncan.gc.ca⁴ [aut], Ceres Barros ceres.barros@ubc.ca⁵ [ctb]

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

5.1 Module Overview

5.1.1 Quick links

- [General functioning](#)
- [List of input objects](#)
- [List of parameters](#)
- [List of outputs](#)
- [Simulation flow and module events](#)

¹https://github.com/PredictiveEcology/Biomass_speciesParameters.git

²https://github.com/PredictiveEcology/Biomass_speciesParameters/issues

³<mailto:ian.eddy@nrcan-rncan.gc.ca>

⁴<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

⁵<mailto:ceres.barros@ubc.ca>

5.1.2 Summary

LandR *Biomass_speciesParameters* (hereafter *Biomass_speciesParameters*) calibrates species growth and mortality trait values used in *Biomass_core*, by matching theoretical species' growth curves obtained with different trait values (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 trait values that allows a better match to the observed curves.

In particular, it directly calibrates the *growthcurve*, *mortalityshape* invariant species traits and two new traits *inflationFactor* and *mANPPproportion*, which are used to calibrate previously estimated species maximum biomass (*maxB*) and maximum aboveground net primary productivity (*maxANPP*) values (see [Parameter estimation/calibration](#)).

This module **will not** obtain other traits or parameters used in *Biomass_core* and so must be used in conjunction with another data/calibration module that does so (e.g., *Biomass_borealDataPrep*).

It can however be used stand-alone in an initial developmental phase for easier inspection of the statistical calibration procedure employed.

As of June 01, 2022, the *raw* PSP data used in this module is not freely available, and data sharing agreements must be obtained from the governments of SK, AB, and BC to obtain it. However, the *processed and anonymized* PSP data is provided via a Google Drive folder accessed automatically by the module.

A Google Account is therefore necessary to access the data used for calibration.

If you do not have a Google Account, or cannot access the data, please report an issue by clicking on the “Get help” link above.

5.1.3 Links to other modules

Biomass_speciesParameters is intended to be used with another data module, like *Biomass_borealDataPrep*, that prepares all other traits and parameters (including *maxB* and *maxANPP*) for *Biomass_core*. See here⁶ for all available modules in the LandR ecosystem and select *Biomass_speciesParameters* from the drop-down menu to see potential linkages.

⁶https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

- *Biomass_borealDataPrep*⁷: prepares all parameters and inputs (including initial landscape conditions) that *Biomass_core* needs to run a realistic simulation. Default values/inputs produced are relevant for boreal forests of Western Canada. Used upstream from *Biomass_speciesParameters*;
- *Biomass_core*⁸: core forest dynamics simulation module. Used downstream from *Biomass_speciesParameters*;

5.2 Module manual

5.2.1 General functioning

Tree cohort growth and mortality in *Biomass_core* are essentially determined by five parameters: the invariant species traits ‘growth curve’ (`growthcurve`), ‘mortality shape’, (`mortalityshape`) and `longevity`, and the spatio-temporally varying traits maximum biomass (`maxB`) and maximum aboveground net primary productivity (`maxANPP`).

All five traits strongly modulate the shape of species growth curves and so it is important that 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 should not be estimated on pure single species growth conditions, as their resulting dynamics will be different in a multi-species context.

Biomass_speciesParameters 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. This also provides a means to calibrate these traits using a dataset that is independent from the one used to derive initial landscape conditions and initial values of `maxB` and `maxANPP`.

While `longevity` is adjusted using published values (see *Biomass_borealDataPrep* manual), the remaining four parameters are

⁷https://github.com/PredictiveEcology/Biomass_borealDataPrep

⁸https://github.com/PredictiveEcology/Biomass_core

calibrated using the PSP data. Hence, *Biomass_speciesParameters* generally follows other data modules, like *Biomass_boreaDataPrep*, that prepare other traits such as `longevity`, `maxB` and `maxANPP`.

5.2.1.1 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 polygon (`studyAreaANPP`; see [List of input objects](#)).

The default PSP data were initially obtained from the National Forest Inventory (NFI), the Alberta Ministry of Agriculture, the Saskatchewan Ministry of the Environment, and the British Columbia Ministry of Forests, treated for errors and standardized into a single data set with the exact location and identifying attributes anonymized. We only share the randomized and anonymized dataset, as data sharing agreements must be met to access the raw data.

The data include individual species, diameter at breast height (DBH), and sometimes tree height 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 estimated by tree species, in g/m^2 , using either the DBH-only model or a DBH-height model from either Lambert, Ung, and Raulier [8] or Ung, Bernier, and Guo [21] (see `P(sim)$biomassModel` module parameter in [list of parameters](#)).

5.2.1.2 Simulated species data

The *Biomass_speciesFactorial* module was used to create a library of theoretical species curves (biomass accumulation curves, to be more precise) to which the empirical species curves derived from PSP-biomass are matched for each species trait combination in the study area. The library of curves was created by running several *Biomass_core* simulations with no reproduction, competition, disturbance, or dispersal effects, on the study area. Each simulation differed in the combination of species trait values that influence growth and mortality dynamics, namely: `growthcurve`, `mortalityshape`, `longevity`, `maxANPP` and maximum biomass (`maxBiomass`, not to be confused with the data-driven `maxB` which is later calibrated).

The values for `maxANPP` were explored via the `maxANPPproportion`, the ratio of `maxANPP` to `maxBiomass` (the parameter used for theoretical curves), as it reflects their relationship.

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 held constant at 5000.

This resulted in over 64,000,000 theoretical curves.

Results from these simulations were compiled into a table (cohortDataFactorial ; see [List of input objects](#)) that is accessed by Biomass_speciesParameters, so that the module can be run without needing to re-simulate the theoretical curves.

5.2.1.3 Parameter estimation/calibration

Biomass_speciesParameters calibrates growthcurve, mortalityshape and mANPPproportion by matching the theoretical species curves produced by Biomass_speciesFactorial (cohortDataFactorial object) against observed species growth curves from permanent sample plot (PSP) data.

Before fitting 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 fit the models using stands where the focal species is dominant (but not monocultures), 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%. This is, when calibrating *Populus tremuloides* traits, PSP data 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 fit using 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:

$$B \sim f_1(\text{standAge}) + (\sim 1 | \text{measureYear} + \text{plotID}) \quad (5.1)$$

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

5.2.1.3.1 Biomass-weighting

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):

$$\frac{\overline{\text{domSpeciesB}_1}}{\overline{\text{allDomSpeciesB}}} \quad (5.2)$$

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 trait (re-)calibration, and values remain unchanged.

After fitting each species GAMM, *Biomass_speciesParameters* compares it to the theoretical curves obtained with a `longevity` value that matches the focal species' longevity, 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 `maxANPPproportion` 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 data sets, 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.

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 ecoregion (ecological zone and land cover combination).

Biomass_speciesParameters uses the ratio between the potential maximum biomass (`maxBiomass`, always 5000) to the achieved maximum biomass in the theoretical curves, to rescale `maxB`. This ratio is called the `inflationFactor` and it is multiplied by `maxB` values previously estimated from data (e.g. by *Biomass_borealDataPrep*). This way, species 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:

$$\text{maxB} \times \frac{\text{manPPproportion}}{100} \quad (5.3)$$

where `maxB` is the already (re-)calibrated version. As already stated above, the final `maxANPP` value is then constrained between 2.0 and 5.0 by default.

In cases where insufficient PSP data prevent fitting the GAMMs and performing the calibration, `manPPproportion` defaults to 3.33 (the value used in LANDIS-II applications in Canada's boreal forests) and the `inflationFactor` to 1.

5.2.2 List of input objects

The full list of input objects required by the module is presented below (Table 5.1). The only input that **must** be provided is `studyAreaANPP` (the study area used extract the PSP data from). All other input objects have internal defaults, but the user may need to request access to their online files.

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

Spatial layers

- `studyAreaANPP` – shapefile. A `SpatialPolygonsDataFrame` with a single

TABLE 5.1: List of *Biomass_speciesParameters* input objects and their description.

objectName	objectClass	desc
factorialSpeciesTable	data.table	table with species traits for matching to 'red'
reducedFactorialCohortData	data.table	results of factorial species trait simulation. 7
PSPmeasure	data.table	merged PSP and TSP individual tree measur
PSPplot_sppParams	data.table	merged PSP and TSP plot data. Defaults to m
PSPgis	sf	Plot location sf object. Defaults to PSP data
species	data.table	a table of invariant species traits with the fo
speciesEcoregion	data.table	table of spatially-varying species traits ('max
sppEquiv	data.table	table of species equivalencies. See '?LandR::s
studyAreaANPP	SpatialPolygonsDataFrame	study area used to crop PSP data before buil

polygon determining the where the PSP should be subset to simulation will take place. This input object **must be supplied by the user or another module**.

Tables

- factorialSpeciesTable and reducedFactorialCohortData – a tables of species trait combinations and the theoretical species growth curve data (respectively).
- PSPmeasure, PSPplot and PSPgis – tree measurement, biomass growth and geographical data of the PSP datasets used to build observed species growth curves.
- 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 'ecoregionGroup' (the ecolocation ID). 'maxB' and 'maxANPP' values are (re-)calibrated by species.

5.2.3 List of parameters

The full list of parameters used by the module is presented below (Table 5.2), all of which have default values specified in the module's metadata.

Of these parameters, the following are particularly important:

Calibration parameters

- `GAMMiterations` and `GAMMknots` – control the number of iterations 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.

5.2.4 List of outputs

The module produces the following outputs (Table 4.4). Note that `species` and `speciesEcoregion` are modified versions of the inputted objects with the same name.

Tables

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

5.2.5 Simulation flow and module events

Biomass_speciesParameters initialises 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).

We advise future users to run *Biomass_speciesParameters* with defaults and inspect what the objects are like before supplying their own data. The user does not need to run *Biomass_speciesFactorial* to generate their own theoretical curves (unless they wish to), as the module accesses a pre-generated on-line library with these simulated data.

Note that this module only runs once (in one “time step”) and only executes one event (`init`). The general flow of *Biomass_speciesParameters* processes is:

1. Preparation of all necessary data and input objects that do not re-

TABLE 5.2: List of *Biomass_speciesParameters* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
.plots	character	screen	NA	NA	Used by Plots function, which can
.plotInitialTime	numeric	start(sim)	NA	NA	This describes the simulation time
.plotInterval	numeric	NA	NA	NA	This describes the simulation time
.saveInitialTime	numeric	NA	NA	NA	This describes the simulation time
.saveInterval	numeric	NA	NA	NA	This describes the simulation time
.useCache	character	.inputOb....	NA	NA	Should this entire module be run
speciesFittingApproach	character	pairwise	NA	NA	Either 'all', 'pairwise', 'focal' or 'sin
biomassModel	character	Lambert2005	NA	NA	The model used to calculate bioma
constrainGrowthCurve	numeric	0, 1	0	1	upper and lower bounds on range
constrainMortalityShape	numeric	15, 25	5	25	upper and lower bounds on morta
constrainMaxANPP	numeric	2, 5	1	10	upper and lower bounds on 'maxA
GAMMiterations	numeric	8	1	NA	number of iterations for GAMMs.
GAMMknots	numeric	3	NA	NA	the number of knots to use in the
minimumPlotsPerGamm	numeric	50	10	NA	minimum number of PSP plots be
minDBH	integer	0	0	NA	minimum diameter at breast heig
PSPdataTypes	character	all	NA	NA	Which PSP datasets to source, def
PSPperiod	numeric	1920, 2019	NA	NA	The years by which to subset samp
quantileAgeSubset	numeric	95	1	100	quantile by which to subset PSP d
sppEquivCol	character	default	NA	NA	The column in 'sim\$sppEquiv' dat
useHeight	logical	FALSE	NA	NA	Should height be used to calculate

quire 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. Theoretical curves are subset to those with longevity matching the species' longevity (in species table) and with growthcurve and mortalityshape values within the chosen boundaries ($P(sim)\$constrainGrowthCurve$, $P(sim)\$constrainMortalityShape$);
4. Calibrating $maxB$ and $maxANPP$

TABLE 5.3: List of *Biomass_speciesParameters* output objects and their description.

Ecolocation	Species	SEP	maxANPP	maxB
1	abiebals	0.90	886	26580
1	acerrubr	1.00	1175	35250
1	acersacc	0.82	1106	33180
1	betualle	0.64	1202	36060
1	betupapy	1.00	1202	36060
1	fraxamer	0.18	1202	36060
1	piceglau	0.58	969	29070
1	pinubank	1.00	1130	33900
1	pinuresi	0.56	1017	30510
1	pinustro	0.72	1090	38150
1	poputrem	1.00	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

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

5.3 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 access to the data may need to be granted.

A realistic usage example of this module and a few others can be found in this repository⁹ and in Barros et al. [1].

⁹https://github.com/CeresBarros/LandRBiomass_publication

5.3.1 Load SpaDES and other packages.

```
if (!require(Require)) {  
  install.packages("Require")  
  library(Require)  
}  
  
Require(c("PredictiveEcology/SpaDES.install", "SpaDES",  
"PredictiveEcology/SpaDES.core@development"),  
  install_githubArgs = list(dependencies = TRUE))
```

5.3.2 Get module, necessary packages and set up folder directories

```
tempDir <- tempdir()  
  
paths <- list(inputPath = normPath(file.path(tempDir,  
"inputs")),  
  cachePath = normPath(file.path(tempDir, "cache")),  
  modulePath = normPath(file.path(tempDir,  
    "modules")), outputPath = normPath(file.path(tempDir,  
    "outputs")))  
  
getModule(  
  "PredictiveEcology/Biomass_speciesParameters@79896a4e3b785e34e5f509798ab6c2204bb334d7"  
  ,  
  modulePath = paths$modulePath, overwrite = TRUE)  
  
## make sure all necessary packages are installed:  
makeSureAllPackagesInstalled(paths$modulePath)
```

5.3.3 Setup simulation

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

```

modules <- list("Biomass_speciesParameters")

# the purpose of this table is experiment with modify
# longevity - longevity is not estimated by the module but
# it is used in trait estimation.
inputSpecies <- data.table(species = c("Abie_bal", "Abie_las",
    "Betu_pap", "Lari_lar", "Pice_eng", "Pice_gla", "Pice_mar",
    "Pinu_ban", "Pinu_con", "Pseu_men", "Popu_tre"), longevity =
    c(300,
    300, 170, 170, 330, 250, 250, 175, 300, 600, 200),
    mortalityshape = 15,
    growthcurve = 0)
objects <- list(species = inputSpecies)

inputs <- list()
outputs <- 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
```

5.4 References

6

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.



7

LandR Biomass_validationKNN Module

Biomass validationKNN v. 0.0.2.9001¹

Get help Report issues²

7.0.0.1 Authors:

Ceres Barros ceres.barros@ubc.ca³ [aut, cre], Eliot McIntire eliot.mcintire@nrcan-rncan.gc.ca⁴ [aut]

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

7.1 Module Overview

7.1.1 Quick links

- General functioning
- List of input objects
- List of parameters
- List of outputs
- Simulation flow and module events

¹https://github.com/PredictiveEcology/Biomass_validationKNN

²https://github.com/PredictiveEcology/Biomass_validationKNN/issues

³<mailto:ceres.barros@ubc.ca>

⁴<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

7.1.2 Summary

LandR *Biomass_validationKNN* (hereafter *Biomass_validationKNN*) provides an approach to validate outputs from LandR Biomass (i.e., *Biomass_core* linked with other modules or not) simulations, using publicly available data for Canadian forests. It produces both a visual and statistical validation of *Biomass_core* outputs 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.

7.1.3 Links to other modules

Biomass_validationKNN is intended to be used with *Biomass_core* and any other modules that link to it and affect cohort biomass (e.g., disturbance modules and calibration modules may both affect resulting biomass). See here⁵ for all available modules in the LandR ecosystem and select *Biomass_validationKNN* from the drop-down menu to see potential linkages. By default, disturbed pixels are excluded from the validation, but the user can bypass this option. The following is a list of the modules commonly validated with *Biomass_validationKNN*.

- *Biomass_core*⁶: core forest dynamics simulation module. Used downstream from *Biomass_borealDataPrep*;

Data and calibration modules:

- *Biomass_speciesData*⁷: grabs and merges several sources of species cover data, making species percent cover (% cover) layers used by other LandR Biomass modules. Default source data spans the entire Canadian territory;
- *Biomass_borealDataPrep*⁸: prepares all parameters and inputs (including initial landscape conditions) that *Biomass_core* needs to run a realistic simulation. Default values/inputs produced are relevant for boreal forests of Western Canada;
- *Biomass_speciesParameters*⁹: calibrates four-species level traits using perma-

⁵https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

⁶https://github.com/PredictiveEcology/Biomass_core

⁷https://github.com/PredictiveEcology/Biomass_speciesData

⁸https://github.com/PredictiveEcology/Biomass_borealDataPrep

⁹https://github.com/PredictiveEcology/Biomass_speciesParameters

nent sample plot data (i.e., repeated tree biomass measurements) across Western Canada.

Disturbance-related modules:

- *Biomass_regeneration*¹⁰: simulates cohort biomass responses to stand-replacing fires (as in the LANDIS-II Biomass Succession Extension v.3.2.1), including cohort mortality and regeneration through resprouting and/or serotiny;
- *Biomass_regenerationPM*¹¹: like *Biomass_regeneration*, but allowing partial mortality. Based on the LANDIS-II Dynamic Fuels & Fire System extension [20];
- *fireSense*: climate- and land-cover-sensitive fire model simulating fire ignition, escape and spread processes as a function of climate and land-cover. Includes built-in parameterisation of these processes using climate, land-cover, fire occurrence and fire perimeter data. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*. See modules prefixed “*fireSense_*” at <https://github.com/PredictiveEcology/>;
- *LandMine*¹²: wildfire ignition and cover-sensitive wildfire spread model based on a fire return interval input. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*;
- *scfm*¹³: spatially explicit fire spread module parameterised and modelled as a stochastic three-part process of ignition, escape, and spread. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*.

7.2 Module manual

7.2.1 General functioning

Biomass_validationKNN 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

¹⁰https://github.com/PredictiveEcology/Biomass_regeneration

¹¹https://github.com/PredictiveEcology/Biomass_regenerationPM

¹²<https://github.com/PredictiveEcology/LandMine>

¹³<https://github.com/PredictiveEcology/scfm>

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 exclude disturbed pixels identified in *rstLCCChange* raster layer and in the fire perimeter data (*firePerimeters* layer). If this is not intended, the user can provide 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 for 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 pix-

els 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.

7.2.2 Validation approaches

7.2.2.1 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).

7.2.2.2 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. MAD values are calculated per replicate and year, except ΔB MAD values, which integrate 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.

7.2.2.3 Sum of negative log-likelihood (SNLL)

To provide a measure of overall goodness of fit of the model set-up that gave rise to the outputs, this is the set of starting conditions, parameters and simulation mechanisms that generated predictions (which includes the LandR modules used), *Biomass_validationKNN* estimates sum of negative log-likelihoods (SNLL) of species presences (at the landscape-level), simulated species biomasses, and ΔB (the latter two at the landscape and pixel levels), 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 :

$$\ell(\theta \mid x) = f(x) \quad (7.1)$$

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 (7.1) is applied to calculate the SNLL estimated for each value of x at the pixel or landscape-level, i :

$$-\sum_{i=1}^N \ell(\theta \mid x_i) \quad (7.2)$$

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

For species presences and species biomass, we draw the probability of observing x_i (a vector of species presences/biomasses in pixel/landscape i) from a multinomial density distribution ($f(x_i) = \text{Multi}(n_i, p_i)$), where $n_i = \sum_{j=1}^K X_{i,j}$ (X being the observed values of biomass of $j = 1, \dots, K$ 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. The following approach is currently implemented, but presents issues:

For ΔB , we draw the probability of observing $x_{i,j}$ (the simulated ΔB of $j = 1, \dots, K$ species in a pixel/landscape i) from a multivariate Gaussian distribution, $f(x_i) = \mathcal{N}(\mu_i, M_i)$, where μ_i is the vector of observed mean ΔB for each species $j = 1, \dots, K$, 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 the entire 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 distribution¹⁴ and on the multivariate Gaussian distribution¹⁵ for a good summary of these two distributions and their use in SNLL estimation.

¹⁴https://en.wikipedia.org/wiki/Multinomial_distribution

¹⁵https://en.wikipedia.org/wiki/Multivariate_normal_distribution#Density_function

7.2.3 List of input objects

The full list of input objects *Biomass_validationKNN* requires is presented below (Table 7.1). All have defaults except `studyArea`, which **must** be provided by the user, or another module.

Of these, the input spatial layers land-cover change (change type and year), fire perimeters, % species cover, stand age and stand biomass are obtained from National Forest Inventory kNN layers for years 2001 and 2011.

We strongly recommend that for the “starting point layers” (those suffixed with `*Start`, which by default correspond to 2001) the user supplies the same objects used as the starting input layers to initialise the simulation to make sure that they match.

Note that objects suffixed with `*Start` correspond to the same objects in the main simulation without this suffix (e.g. `rawBiomassMapStart` is `rawBiomassMap` in *Biomass_borealDataPrep*), whereas other objects like `studyArea` and `rasterToMatch` have the same names in the simulation and should be **exactly** the same object.

Of the inputs in Table 7.1, the following 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 initialise and parametrise *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 simulation years. By default, the module uses the kNN stand biomass map from 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)$LCChangeYr`. Defaults to Canada's forest change national map between 1985-2011 (CFS)¹⁶.
- `rstLCChangeYr` – a raster layer with year of disturbance. This is an optional layer that can be combined with `rstLCChange` and `P(sim)$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)¹⁷.
- `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.

¹⁶https://opendata.nfis.org/downloads/forest_change/C2C_change_type_1985_2011.zip

¹⁷https://opendata.nfis.org/downloads/forest_change/C2C_change_year_1985_2011.zip

TABLE 7.1: List of *Biomass_validationKNN* input objects and their description.

objectName	objectClass	desc
allCohortData	data.table	All ‘cohortData’ tables saved during the simulation,
biomassMap	RasterLayer	total biomass raster layer in study area (in g/m ²),
firePerimeters	sf	A map of fire perimeters in the study area that can
fireURL	character	A URL to a fire database, such as the Canadian Nati
pixelGroupMapStk	RasterStack	A stack of ‘pixelGroupMap’s saved during the simul
rawBiomassMapStart	RasterLayer	observed total biomass raster layer in study area at
rawBiomassMapEnd	RasterLayer	observed total biomass raster layer in study area at
rasterToMatch	RasterLayer	A raster of the ‘studyArea’ in the same resolution an
rstLCChange	RasterLayer	A mask-type map of land cover changes in the study
rstLCChangeYr	RasterLayer	An OPTIONAL map of land cover change years in th
simulationOutputs	data.table	An OPTIONAL table listing simulation outputs (as p
speciesLayersStart	RasterStack	observed cover percentage raster layers by species i
speciesLayersEnd	RasterStack	observed percent cover raster layers by species in C
sppColorVect	character	A named vector of colors to use for plotting. The na
sppEquiv	data.table	table of species equivalencies. See ‘LandR::sppEqui
standAgeMapStart	RasterLayer	observed stand age map in study area, at the first y
standAgeMapEnd	RasterLayer	observed stand age raster layer in study area, at the
studyArea	SpatialPolygonsDataFrame	Polygon to use as the study area. Must be provided

- *pixelGroupMap* – a raster layer with *pixelGroup* IDs per pixel. Pixels are 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).

7.2.4 List of parameters

Table 7.2 lists all parameters used in *Biomass_validationKNN* and their detailed information. All have default values specified in the module’s metadata

Of the parameters listed in Table 7.2, 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.

TABLE 7.2: 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
deciduousCoverDiscount	numeric	0.8418911	NA	NA	This was estimated with data from NWT on March 18, 2020 and may or may not be applicable elsewhere
LCChangeYr	integer		1900	NA	OPTIONAL. An integer or vector of integers of the validation period years, defining the years to use for validation
minCoverThreshold	numeric	5	0	100	Cover that is equal to or below this number will be omitted from the dataset Should be less than coverThresh
obsDeltaAgeB	logical	TRUE	NA	NA	When TRUE, the observed changes in biomass and age (deltaB, deltaAge) between years are used
pixelGroupBiomassClass	numeric	100	NA	NA	When assigning 'pixelGroup' membership, this defines the resolution of biomass to use
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 validation
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 function
.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

7.2.5 List of outputs

The module produces the following outputs (Table 7.3):

7.2.6 Simulation flow and module events

Biomass_validationKNN initialises 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 validation¹⁸.

The module then compiles 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 [list of input objects](#) for more detail.

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. We expect the number of validation modules to increase as other validation approaches are developed based on project needs.

The general flow of *Biomass_validationKNN* processes is (note that this module only runs once, i.e. in one “time step”):

1. Preparation of all necessary objects, namely obtaining the observed data layers from on-line repositories (or if available stored local copies) and the compiling simulated data if the user has not done so previously (see [list of input objects](#)) – (`init` event).
2. Calculation of summary variables for validation (`calculateValidVars` event), 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

¹⁸Raw data layers downloaded by the module are saved in ‘`dataPath(sim)`’, which can be controlled via ‘`options(reproducible.destinationPath = ...)`’.

TABLE 7.3: List of *Biomass_validationKNN* output objects and their description.

objectName	objectClass	desc
logLikelihood	data.table	A table of negative sum log-likelihood values calculated for different repetitions. At the moment, log-likelihood values are calculated for pixel-level, species presences and dominance (landscape-level) and pixel-level. For biomass and count data (presences/dominance, we use multinomial distribution, and for deltaB a multivariate Gaussian distribution is still under development.
landscapeMAD	data.table	Mean absolute deviance values calculated on landscape-level relative presences and dominance, and deltaB, per repetition and year (except for deltaB, which is integrated across years)
landscapeVars	data.table	A table containing observed and simulated landscape-averaged variables (by year and repetition, 'rep', in the case of simulated data), namely: species relative abundance ('relAbund'), species presences ('count'), species dominance (as in relative abundance; 'countDom') and species changes in biomass ('deltaB'). Observed data rows are labelled as 'observed' in 'dataType' column. Pixels with ≥ 2 species with max(B) and pixels with no B are classified as 'noDoms' respectively in the 'speciesCode' column - note that this is 'vegType' in the input data.
pixelCohortData	data.table	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'), average biomass-weighted pixel age ('pixelAge'), species relative abundance ('relativeAbund'), species dominance (the species with max(B), 'vegType'), and species biomass ('landscapeB'). Observed data columns are suffixed with 'C' for cohort. Pixels with ≥ 2 species with max(B) (i.e. 'noDoms' ≥ 2) are classified as 'noDoms' in the 'speciesCode' column.
pixelMAD	data.table	Mean absolute deviance values calculated on pixel-level relative abundance, species presences and dominance, and deltaB, per repetition and year (except for deltaB, which is integrated across years)
pixelVars	data.table	The same as 'landscapeVars', but variables are calculated at the pixel-level
rstDisturbedPix	RasterLayer	Raster of pixel IDs (as a mask) that have been disturbed by fire or other events during the validation period. These pixels are excluded from the validation
rawBiomassMapStart	RasterLayer	observed total biomass raster layer in study area at the first year of the validation period to exclude pixels that were disturbed during the validation period
rawBiomassMapEnd	RasterLayer	observed total biomass raster layer in study area at the last year of the validation period to exclude pixels that were disturbed during the validation period
speciesLayersStart	RasterStack	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	RasterStack	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	RasterLayer	observed stand age map in study area, at the first year of the validation period to exclude pixels that were disturbed during the validation period
standAgeMapEnd	RasterLayer	observed stand age map in study area, at the last year of the validation period to exclude pixels that were disturbed during the validation period

- species presences across the landscape
3. Calculation of validation statistics (`validationStats` event), namely mean absolute deviations (MAD) and sum of negative log-likelihoods (SNLL).
 4. Assessment of the relationship between observed ΔB and observed ΔAge (`obsDeltaMaps` event) – this is an optional visual diagnostic of the observed data that produces scatterplots of $\Delta B \sim \Delta \text{Age}$ 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 (`landscapeWidePlots`, `pixelLevelPlots` and `deltaBComparisons` events):
 - 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 7.3).

7.3 Usage example

7.3.1 Load SpaDES and other packages.

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

7.3.2 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()
paths <- list(inputPath = file.path(tempDir, "inputs"),
  cachePath = file.path(tempDir,
    "cache"), modulePath = file.path(tempDir, "modules"),
  outputPath = file.path(tempDir,
    "outputs"))

getModule("PredictiveEcology/Biomass_core", modulePath =
  paths$modulePath,
  overwrite = TRUE)
getModule("PredictiveEcology/Biomass_validationKNN", modulePath
  = paths$modulePath,
  overwrite = TRUE)

## 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(paths$modulePath,
  "Biomass_core-master"),
```

```

file.path(paths$modulePath,
  "Biomass_validationKNN-master")),
c(file.path(paths$modulePath, "Biomass_core"),
file.path(paths$modulePath,
  "Biomass_validationKNN"))))

```

7.3.3 Setup simulation

```

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

studyArea <- Cache(randomStudyArea, size = 1e+07) # cache this
so it creates a random one only once on a machine

# Pick the species you want to work with -- using the
# naming convention in 'Boreal' column of
# LandR::sppEquivalencies_CA
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 (omitted 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 = 1)),
  eventPriority = 1))

```

7.3.4 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)

```

7.3.5 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*

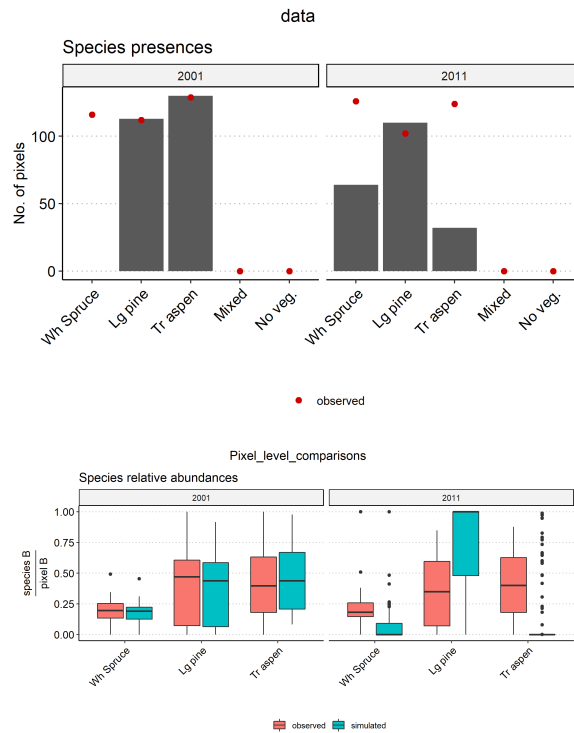


FIGURE 7.1: *Biomass_validationKNN* automatically generates plots showing a visual comparison between simulated and observed species presences (right) across the landscape, and relative species biomass per pixel (left).

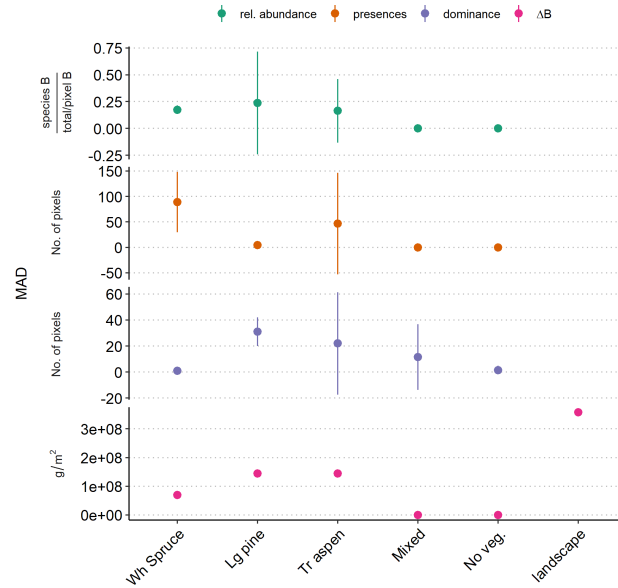


FIGURE 7.2: A plot of landscape-wide mean absolute deviations (MAD) from (top to bottom) observed mean relative abundance, no. of presences, no. of pixels where the species is dominant and ΔB .

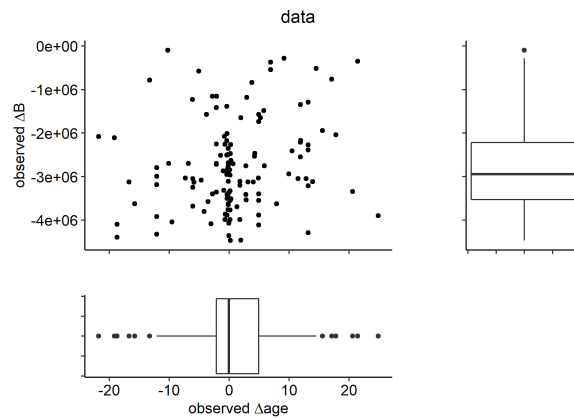


FIGURE 7.3: Diagnostic plot of observed changes in biomass and age ΔB and ΔAge , respectively).

7.4 References



Bibliography

- [1] Ceres Barros et al. “Empowering ecologists with a PERFICT workflow: seamlessly linking data, parameterization, prediction, validation and visualization”. in review.
- [2] 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-87A2-6ACED147A990¹⁹. URL: <http://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990>.
- [3] 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.
- [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/X08-001²⁰. 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 Re-*

¹⁹<https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990>

²⁰<https://doi.org/10.1139/X08-001>

- search 35 (2005), pp. 1996–2018. URL: <http://cdnsiencepub.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: 10.2307/3237266²¹. 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/nclimate1293²². 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.2004.01.022²³. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304380004003837>.

²¹<https://doi.org/10.2307/3237266>

²²<https://doi.org/10.1038/nclimate1293>

²³<https://doi.org/10.1016/j.ecolmodel.2004.01.022>

- [19] Shelley Stall et al. “Make scientific data FAIR”. In: *Nature* 570.7759 (June 2019), pp. 27–29. DOI: 10.1038/d41586-019-01720-7²⁴. URL: <http://www.nature.com/articles/d41586-019-01720-7>.
- [20] Brian R. Sturtevant et al. *LANDIS-II Dynamic Fire System Extension v3.0 – User Guide*. 2018.
- [21] 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/X07-224²⁵. URL: <https://cdnsiencepub-com.eu1.proxy.openathens.net/doi/full/10.1139/X07-224>.
- [22] Hadley Wickham. “testthat: Get Started with Testing”. In: *The R Journal* 3.1 (2011), p. 5. DOI: 10.32614/RJ-2011-002²⁶. URL: <https://journal.r-project.org/archive/2011/RJ-2011-002/index.html>.
- [23] 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.2016.18²⁷. URL: <http://www.nature.com/articles/sdata201618>.
- [24] 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.2011.04.032²⁸. URL: <https://www.sciencedirect.com/science/article/pii/S0304380011002651>.

²⁴<https://doi.org/10.1038/d41586-019-01720-7>

²⁵<https://doi.org/10.1139/X07-224>

²⁶<https://doi.org/10.32614/RJ-2011-002>

²⁷<https://doi.org/10.1038/sdata.2016.18>

²⁸<https://doi.org/10.1016/j.ecolmodel.2011.04.032>