

Alex M. Chubaty; Ceres Barros; Ian M. S. Eddy; Eliot J. B. McIntire; David W. Andison

LandWeb Manual



Contents

List of Tables	9
List of Figures	13
Overview	17
0.1 Background	17
0.2 The LandWeb Model	17
0.2.1 Data preparation	18
0.2.2 Vegetation dynamics	19
0.2.3 Wildfire dynamics	20
0.2.4 Summary maps and statistics	20
0.2.5 LandWeb app	20
0.3 Previous Manual Versions	20
1 Getting started	21
1.1 Prerequisites	21
1.1.1 Docker	21
1.1.2 Development tools	22
1.1.3 Geospatial libraries	24
1.1.4 git, Git Kraken, and GitHub	26
1.1.5 R and Rstudio	27
1.2 Getting the code	29
1.3 Project directory structure	30
1.4 Updating the code	30
1.4.1 Using GitKraken	31
1.4.2 Using the command line	31
1.5 Data requirements	32
1.6 Getting help	32
2 Running LandWeb	33
2.1 Model setup and configuration	33
2.1.1 Select a study area	33

2.1.2	Select a scenario	40
2.1.3	Replication	41
2.2	Running the model	41
2.2.1	Interactive R session	41
2.2.2	Commandline interface	41
2.3	Post-processing analyses	42
2.4	Advanced setup	43
2.4.1	Customizing model run configuration	43
2.4.2	Cache backend	43
2.4.3	Speeding up disk-based operations	43
2.5	Additional Resources	44
Study areas		45
3	<i>LandWeb_preamble</i> Module	47
3.1	Module Overview	47
3.1.1	Module summary	47
3.2	Parameters	47
3.3	Data dependencies	48
3.3.1	Input data	48
3.3.2	Output data	48
3.4	Links to other modules	48
Vegetation submodel		49
4	<i>LandR_Biomass_speciesData</i> Module	51
4.1	Module Overview	51
4.1.1	Module summary	51
4.1.2	Module inputs and parameters at a glance	51
4.1.3	Events	53
4.1.4	Module outputs	53
4.1.5	Links to other modules	53
4.1.6	Getting help	53
4.2	Module manual	54
4.2.1	Detailed description	54
4.2.2	Initialization, inputs and parameters	55
4.2.3	Simulation flow	58
4.3	Usage example	58
4.3.1	Load SpaDES and other packages.	58

<i>Contents</i>	5
4.3.2 Get module, necessary packages and set up folder di- rectories	59
4.3.3 Setup simulation	59
4.3.4 Run module	60
4.4 References	61
5 LandR <i>Biomass_borealDataPrep</i> Module	63
5.1 Module Overview	63
5.1.1 Quick links	63
5.1.2 Summary	63
5.1.3 Links to other modules	64
5.2 Module manual	65
5.2.1 General functioning	65
5.2.2 Data acquisition and treatment	66
5.2.3 Parameter estimation/calibration	71
5.2.4 Aggregating species	77
5.2.5 List of input objects	79
5.2.6 List of parameters	89
5.2.7 List of outputs	98
5.2.8 Simulation flow and module events	101
5.3 Usage example	102
5.4 References	102
6 LandR <i>Biomass_core</i> Module	105
6.1 Module Overview	105
6.1.1 Quick links	105
6.1.2 Summary	106
6.1.3 Links to other modules	107
6.2 Module manual	108
6.2.1 General functioning	108
6.2.2 Initialisation, inputs and parameters	110
6.2.3 List of input objects	117
6.2.4 List of parameters	127
6.2.5 List of outputs	138
6.2.6 Simulation flow and module events	142
6.2.7 Differences between <i>Biomass_core</i> and the LANDIS-II Biomass Succession Extension model (LBSE)	143
6.3 Usage example	152
6.3.1 Set up R libraries	152
6.3.2 Get the module and module dependencies	153

6.3.3	Setup simulation	154
6.3.4	Run simulation	156
6.4	Appendix	157
6.4.1	Tables	157
6.5	References	169
7	LandR Biomass_regeneration Module	171
7.1	Module Overview	171
7.1.1	Module summary	171
7.1.2	General flow of Biomass_regeneration processes - fire disturbances only	173
7.1.3	Module inputs and parameters	173
7.1.4	Module outputs	173
7.1.5	Links to other modules	174
7.2	Getting help	174
	Wildfire submodel	175
8	LandMine Module	177
8.1	Module Overview	177
8.1.1	Module summary	177
8.1.2	Model Differences	177
8.1.3	Known Species	178
8.1.4	Module inputs and parameters	179
8.1.5	Module outputs	179
8.2	Usage	180
8.2.1	Package dependencies	180
8.2.2	Module usage	180
8.3	Testing the burn algorithm	182
8.4	Optimizing parameters	186
8.5	Manual inspection of optimization results	190
8.5.1	Original (2018) version	190
8.5.2	Current (2022) version	194
8.5.3	Cleaning up	195
8.5.4	Code and data availability	195
8.5.5	Links to other modules	195
	Model outputs	197
8.6	Currently Selected Spatial Area	197
8.7	Large Patches Data for study region	197

<i>Contents</i>	7
8.8 Leading Vegetation Cover Data for study region	197
8.9 Simulation Rasters (cropped to study region)	198
8.10 Additional R Data Files (advanced users)	198
9 burnSummaries Module	199
9.1 Module Overview	199
9.1.1 Module summary	199
9.1.2 Module inputs and parameters	199
9.1.3 Events	200
9.1.4 Plotting	200
9.1.5 Saving	200
9.1.6 Module outputs	200
9.1.7 Links to other modules	200
9.1.8 Getting help	200
10 LandWeb_output Module	201
10.1 Module Overview	201
10.1.1 Module summary	201
10.1.2 Module inputs and parameters	201
10.1.3 Module outputs	201
10.1.4 Links to other modules	202
10.1.5 Getting help	202
11 timeSinceFire Module	203
11.1 Module Overview	203
11.1.1 Module summary	203
11.1.2 Module inputs and parameters	203
11.1.3 Events	204
11.1.4 Module outputs	205
11.1.5 Code and data availability	205
11.1.6 Links to other modules	205
12 LandWeb_summary Module	207
12.1 Module Overview	207
12.1.1 Module summary	207
12.1.2 Module inputs and parameters	207
12.1.3 Module outputs	208
12.1.4 Links to other modules	208
12.1.5 Getting help	208
Web app	209

13 Modifying LandWeb	211
13.1 Example 1: adding new reporting polygons	211
13.1.1 National-scale polygons	211
13.1.2 Regional or study-area-specific polygons	212
13.2 Example 2: adding a new study area	214
13.3 Example 3: updating the LTHFC map	215
13.4 Contributing changes	216
References	217

List of Tables

1.1	LandWeb project directory structure	30
2.1	FMA polygon IDs (from Figure 1) and their corresponding FMA names.	37
2.2	Model study areas with corresponding FMA polygon IDs (from Figure 1).	39
4.1	List of <i>Biomass_speciesData</i> input objects and their description.	52
4.2	List of <i>Biomass_speciesData</i> parameters and their description.	52
4.3	List of <i>Biomass_speciesData</i> output objects and their description.	53
4.4	List of species cover data downloaded by default by <i>Biomass_speciesData</i>	56
4.5	List of <i>Biomass_speciesData</i> input objects and their description.	57
4.6	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</i> spp. as a genus-level group. For this, all six species are identified in the KNNcolumn, so that their % cover layers can be obtained, but in the Borealcolumn (which defines the naming convention used in the simulation in this example) all <i>Picea</i> spp. have the same name. <i>Biomass_speciesData</i> will merge their % cover data into a single layer by summing their cover per pixel.	57
4.7	List of <i>Biomass_speciesData</i> parameters and their description.	57

5.1	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</i> 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 <i>Picea</i> spp. have the same name. <i>Biomass_borealDataPrep</i> will merge their % cover data into a single layer by summing their cover per pixel. (continued below)	78
5.3	List of <i>Biomass_borealDataPrep</i> input objects and their description.	81
5.4	List of <i>Biomass_borealDataPrep</i> parameters and their description.	90
5.5	List of <i>Biomass_borealDataPrep</i> output objects and their description.	99
6.1	Example of an invariant species traits table (the <i>species</i> table object in the module), with species <i>Abies</i> sp. (<i>Abie_sp</i>), <i>Picea engelmannii</i> (<i>Pice_eng</i>), <i>Picea glauca</i> (<i>Pice_gla</i>), <i>Pinus</i> sp. (<i>Pinu_sp</i>), <i>Populus</i> sp. (<i>Popu_sp</i>) and <i>Pseudotsuga menziesii</i> (<i>Pseu_men</i>). Note that these are theoretical values. (continued below)	113
6.2	Table continues below	113
6.4	Example of a spatio-temporally varying species traits table (the <i>speciesEcoregion</i> table object in the module), with two ecolocations (called <i>ecoregionGroups</i>) and species <i>Abies</i> sp. (<i>Abie_sp</i>), <i>Picea engelmannii</i> (<i>Pice_eng</i>), <i>Picea glauca</i> (<i>Pice_gla</i>), <i>Pinus</i> sp. (<i>Pinu_sp</i>), <i>Populus</i> sp. (<i>Popu_sp</i>) and <i>Pseudotsuga menziesii</i> (<i>Pseu_men</i>). If a simulation runs for 10 year using this table, trait values from year 2 would be used during simulation years 2-10.	115
6.5	Example of a minimum relative biomass table (the <i>minRelativeB</i> table object in the module), with two ecolocations (<i>ecoregionGroups</i>) sharing the same values	116
6.6	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.	117

6.7	List of <i>Biomass_core</i> input objects and their description.	121
6.8	List of <i>Biomass_core</i> parameters and their description.	129
6.9	List of <i>Biomass_core</i> output objects and their description.	138
6.10	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 <i>Landis-log.txt</i> when <i>CalibrateMode</i> was set to 'yes'. Species starting ages are also shown. (continued below)	158
6.12	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 <i>Landis-log.txt</i> when <i>CalibrateMode</i> was set to 'yes'. Species starting ages are also shown. (continued below)	160
6.14	Randomly generated community combination no. 1 used in the recruitment comparison runs.	162
6.15	Randomly generated community combination no. 2 used in the recruitment comparison runs.	163
6.16	Randomly generated community combination no. 3 used in the recruitment comparison runs.	165
6.17	Invariant species traits table used in comparison runs. (continued below)	167
6.19	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.	168
6.20	Probability of germination for species shade tolerance and shade level combinations (called <i>sufficient light</i> table in LBSE and <i>sufficientlight</i> input data.table in LandR <i>Biomass_core</i>) used in comparison runs.	168
6.21	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.	168

7.1	List of <i>Biomass_regeneration</i> input objects and their description.	173
7.2	List of <i>Biomass_regeneration</i> parameters and their description.	173
7.3	List of <i>Biomass_regeneration</i> outputs and their description.	173
8.1	LandMine species codes.	178
8.2	List of <i>LandMine</i> input objects and their description.	179
8.3	List of <i>LandMine</i> parameters and their description.	179
8.4	List of <i>LandMine</i> outputs and their description.	179
9.1	List of <i>burnSummaries</i> input objects and their description.	199
9.2	List of <i>burnSummaries</i> parameters and their description.	200
9.3	List of <i>burnSummaries</i> outputs and their description.	200
10.1	List of <i>LandWeb_output</i> input objects and their description.	201
10.2	List of <i>LandWeb_output</i> parameters and their description.	202
10.3	List of <i>LandWeb_output</i> outputs and their description.	202
11.1	List of <i>timeSinceFire</i> input objects and their description.	204
11.2	List of <i>timeSinceFire</i> parameters and their description.	204
11.3	List of <i>timeSinceFire</i> outputs and their description.	205
12.1	List of <i>LandWeb_summary</i> input objects and their description.	207
12.2	List of <i>LandWeb_summary</i> parameters and their description.	208
12.3	List of <i>LandWeb_summary</i> outputs and their description.	208

List of Figures

1	LandWeb study area (blue) with mountain and boreal caribou ranges highlighted (pink).	18
1.1	Screenshot showing code commits in Git Kraken. The submodules pane is highlighted on the bottom left.	31
2.1	FMAs within the LandWeb study area, identified by polygon IDs, which are described in Table 1. Polygon fill colours are used only to differentiate neighbouring FMAs.	36
4.1	<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.	61
5.1	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.	73
5.2	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.	74
6.1	<i>Biomass_core</i> 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).	106

- 6.2 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 above-ground 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 $\frac{\text{Biomass}_{\text{order}2} - \text{Biomass}_{\text{order}1}}{\text{Biomass}_{\text{order}2}} * 100$ 145
- 6.3 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. 146
- 6.4 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. 149
- 6.5 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 $\frac{\text{Biomass}_{\text{LBSE}} - \text{Biomass}_{\text{Biomasscore}}}{\text{Biomass}_{\text{LBSE}}} * 100$ 151

List of Figures

15

- 6.6 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. 152
- 6.7 *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). 156



Overview

This manual is a live document which is automatically updated as changes are made to underlying model code and documentation.

0.1 Background

0.2 The LandWeb Model

LandWeb is the first large scale, data-driven approach to simulating historic natural range of variation (H/NRV) (<https://landweb.ca>). In developing the model, analyses, as well as the infrastructure to host data, we strove to implement a single, reproducible workflow to facilitate running simulations, analyses, and model reuse and future expansion. This tight linkage between data and simulation model is made possible via its implementation using the SpaDES family of packages [8] within the R Statistical Language and Environment [**R-base**]. For more information about SpaDES, see <https://spades.predictiveecology.org/>.

The LandWeb model integrates two well-used models for forest stand succession and wildfire simulation, implemented in the SpaDES simulation platform as a collection of submodels (implement as SpaDES modules). Vegetation dynamics are modeled using the LandR Biomass suite of modules, which reimplement the LANDIS-II Biomass Succession model [21, 16] in R. Wildfire dynamics are modeled using an implementation of LandMine [1, 2]. Simulations were run for the entire LandWeb study area, which spans most of the western Canadian boreal forest. A summary of the results are presented using a web app, which can be run locally.

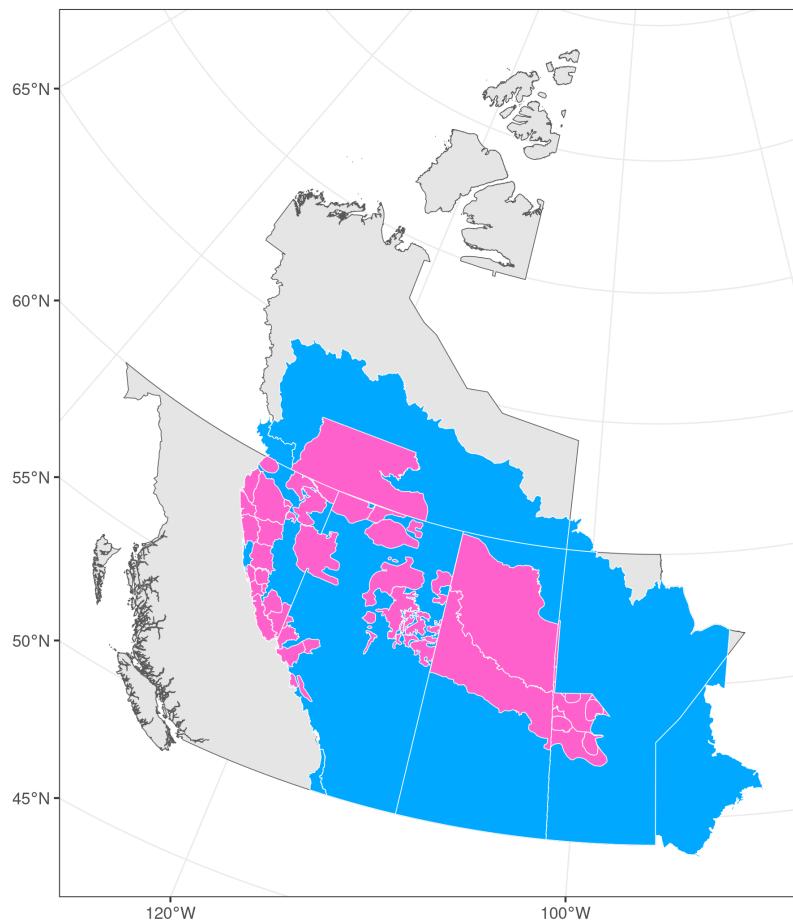


FIGURE 1: LandWeb study area (blue) with mountain and boreal caribou ranges highlighted (pink).

0.2.1 Data preparation

Input data were derived from multiple sources, including several publicly available as well as proprietary datasets.

Detailed descriptions of these sources are provided in the relevant sections of this manual.

0.2.1.1 Public data sources

- Land Cover Classification 2005 map (no longer available from Government of Canada's Open Data website);
- LANDIS-II species traits: https://github.com/dcyr/LANDIS-II_IA_generalUseFiles;
- LANDIS-II parameterization tables and data: <https://github.com/LANDIS-II-Foundation/Extensions-Succession-Archive/master/biomass-succession-archive/trunk/tests/v6.0-2.0/>;
- Canada biomass, stand volume, and species data [from 4]: <http://tree.pfc.forestry.ca>;
- National ecodistrict polygons: http://sis.agr.gc.ca/cansis/nsdb/ecostrat/district/ecodistrict_shp.zip;
- National ecoregion polygons: http://sis.agr.gc.ca/cansis/nsdb/ecostrat/region/ecoregion_shp.zip;
- National ecozone polygons: http://sis.agr.gc.ca/cansis/nsdb/ecostrat/zone/ecozone_shp.zip.

0.2.1.2 Proprietary data sources

All proprietary data used by for the model are stored in an access-controlled Google Drive location.

- biomass by species maps created by Pickell & Coops [15] resolution 100m x 100m from LandSat and kNN based on CASFRI;
- various reporting polygons used to summarize model results in the app.

To request access, please contact Alex Chubaty (achubaty@for-cast.ca¹).

0.2.2 Vegetation dynamics

Vegetation growth and succession are modeled using a re-implementation of the LANDIS-II Biomass model, a widely used and well-documented dynamic vegetation succession model [21, 16]. Our re-implemented model largely follows the original LANDIS-II source code (v 3.6.2; Scheller and Miranda [19]), but with some modifications with respect to species traits parameterization. This model simulates landscape-scale forest dynamics in a spatio-temporally explicit manner, using cohorts of tree species within each pixel. Multiple ecological processes are captured by the model, including vegetation growth, mortality, seed dispersal, and post-disturbance regeneration.

¹<mailto:achubaty@for-cast.ca>

This submodel is described in further detail in [Vegetation submodel](#).

0.2.3 Wildfire dynamics

Wildfire is simulated using a re-implementation of the fire submodel of An-dison's [1, 2] LandMine model of landscape disturbance.

This submodel is described in further detail in [Wildfire submodel](#).

0.2.4 Summary maps and statistics

Summaries are derived from simulation outputs, and consist of maps showing the time since fire as well as histogram summaries of 1) number of large patches (i.e., patches above the number of hectares specified by the user) contained within the selected spatial area; and 2) the vegetation cover within the selected spatial area. Histograms are provided for each spatial area by polygon, age class, and species. Authorized users can additionally overlay current stand conditions onto these histograms. Simulation outputs are summarized for several publicly available reporting polygons (including Alberta Natural Ecoregions and Caribou Ranges).

These are described in further detail in [Model outputs](#).

0.2.5 LandWeb app

Using the web app is described in [Web app](#).

0.3 Previous Manual Versions

If available, archived copies of previous manual versions are provided at the links below.

- LandR Manual v3.0.0² (*current*)

²[archive/pdf/LandWeb-manual-v3.0.0.pdf](#)

1

Getting started

1.1 Prerequisites

Minimum system requirements:

- Windows 10, macOS 10.13 High Sierra, or Ubuntu 20.04 LTS;
- 20 GB of storage space, plus additional storage for model outputs;
- 128 GB RAM to run the model over the full area (less for sub-areas);
- High-speed internet connection.

The following section provides details on installing prerequisite software for running LandWeb.

1.1.1 Docker

If you prefer to not use Docker, skip this subsection.

Due to idiosyncratic difficulties of installing multiple pieces of software and ensuring the correct versions are used throughout, we provide prebuilt Docker (<https://www.docker.com/>) images, which better provides a consistent and reproducible software environment for running the model.

Thus, using these images are preferred over ‘bare-metal’ installation.

Install Docker for your system following <https://docs.docker.com/get-docker/>.

Next, pull the image from Docker Hub:

```
## get the image
docker pull achubaty/landweb-standalone:latest

## launch a new container based on thi image
docker run -d -it \
```

```
-e GITHUB_PAT=$(cat ${HOME}/.Renviron | grep GITHUB_PAT | cut
-d '=' -f 2) \
-e PASSWORD='<mySecretPassword>' \
--memory=128g \
--cpus=32 \
-p 127.0.0.1:8080:8787 \
--name LandWeb \
achubaty/landweb-standalone:latest
```

Once the container is running, open your web browser and go to `localhost:8080`.

Login to the Rstudio session as user `rstudio` and password `<mySecretPassword>` (change this password when launching container above).

Once finished, you can stop and destroy the container:

```
docker stop LandWeb
docker rm LandWeb
```

1.1.2 Development tools

1.1.2.1 Windows

1. Download `Rtools` version 4.2 from <https://cran.r-project.org/bin/windows/Rtools/rtools42/rtools.html> and install it as `administrator`. `Rtools` provides the necessary compilers etc. to build and install R packages from source on Windows.
 - a. During installation, be sure to check the option to add `Rtools` to your PATH.
2. Download and install a proper text editor, e.g. Notepad++ (<https://notepad-plus-plus.org/downloads/>).

1.1.2.2 macOS

1.1.2.2.1 Xcode command line tools

To build software, you will need the Xcode command line tools¹, which include various compilers and git version control software.

```
xcode-select --install
```

1.1.2.2.2 homebrew package manager

Next, install homebrew which provides a package manager for macOS. This will facilitate software updates and will handle various package dependency issues automatically.

```
/bin/bash -c "$(curl -fsSL
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh
)"
```

1.1.2.3 Ubuntu Linux

```
sudo apt-get update

sudo apt-get -y install \
    build-essential \
    biber \
    ccache \
    cmake \
    curl \
    libarchive-dev \
    libcairo2-dev \
    libcurl4-openssl-dev \
    libgit2-dev \
    libglpk-dev \
    libgmp3-dev \
    libicu-dev \
    libjq-dev \
    libmagick++-dev \
```

¹<https://developer.apple.com/downloads/>

```
libnode-dev \
libpng-dev \
libprotobuf-dev \
libprotoc-dev \
libssh2-1-dev \
libssl-dev \
libxml2-dev \
libxt-dev \
make \
p7zip-full p7zip-rar \
pandoc pandoc-citeproc \
protobuf-compiler \
qpdf \
screen \
sysstat \
texinfo texlive-base texlive-bibtex-extra \
texlive-fonts-extra texlive-latex-extra texlive-xetex \
wget \
xauth \
xfonts-base \
xvfb \
zlib1g-dev
```

1.1.3 Geospatial libraries

In order to work with geospatial data, recent versions of GDAL, PROJ, and GEOS geospatial libraries need to be available on your system.

1.1.3.1 Windows

No additional should be needed, as recent versions of R geospatial packages include pre-bundled versions of GDAL, PROJ, and GEOS.

1.1.3.2 macOS

Use homebrew to install the required geospatial software libraries:

```
brew install pkg-config
brew install gdal
```

```
# brew install geos
# brew install proj
brew install udunits
```

1.1.3.3 Ubuntu Linux

The default Ubuntu 20.04 LTS package repositories ship older versions of the geospatial libraries we will be using, so we will need to add some additional repositories to get the latest versions.

```
## add GIS repository
sudo add-apt-repository ppa:ubuntugis-unstable/ppa
sudo apt-get update
```

Install additional system dependencies that serve as prerequisites for running the LandWeb model in R.

```
sudo apt-get -y install \
    gdal-bin \
    libgdal-dev \
    libgeos-dev \
    libproj-dev \
    libudunits2-dev \
    python3-gdal
```

Optionally, we install `mapshaper` geospatial library which is used to speed up polygon simplification.

```
## mapshaper installation
sudo apt-get remove -y libnode-dev

curl -sL https://deb.nodesource.com/setup_20.x | sudo -E bash -

sudo apt install nodejs
sudo npm install npm@latest -g
sudo npm install -g mapshaper
```

1.1.4 git, Git Kraken, and GitHub

git is the version control software used throughout this project, and is required to ‘checkout’ specific versions of the code as well as to make changes and ‘push’ these changes to the model code repository.

1. Install the latest version of git from <https://git-scm.com/downloads> or via your package manager.

Windows users should install *as administrator*. Use nano (instead of vi/vim) as the default text editor. For all other choices, use the recommended settings.

For macOS users, git is included with the Xcode command line tools.

2. Create a GitHub (<https://github.com>) account if you don’t already have one, and configure a Personal Access Token (PAT).

A GitHub (<https://github.com>) account is required to assist with package installation and accessing model code.

Several packages used by LandWeb are only available on GitHub. Because we will be installing several of these, we want to ensure we can do so without GitHub rate-limiting our requests. Without a PAT, some packages may *temporarily* fail to install, but can be retried a little later (usually 1 hour).

- a. Create a GitHub PAT following the instructions²;
- b. Be sure to uncheck all scopes.
- c. Copy this token and save it in a text file in your home directory called `.Renvironment`:

Run the following in an R session to edit `~/.Renvironment`:

```
install.packages("usethis")
usethis::edit_r_environ()
```

Paste the following into the `.Renvironment` file, using your copied token:

²<https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token>

```
GITHUB_PAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

If creating `~/.Renviron` using a text editor, be sure to save it *without* a `.txt` extension. Windows users in particular should ensure they edit this file via Rstudio per above, or use a proper text editor, e.g., Notepad++ (<https://notepad-plus-plus.org/downloads/>).

3. *Optional.* Install the latest version of GitKraken from <https://www.gitkraken.com/download/>.

The free version is sufficient to access the public repositories used in this project. However, the paid pro version is required to access private repositories.

1.1.5 R and Rstudio

1. Download and install R version 4.2.3.

Windows

1. Download R from <https://cran.r-project.org/bin/windows/base/R-4.2.3-win.exe>;
2. Install R *as administrator*.

macOS

1. Install rig (<https://github.com/r-lib/rig>) to manage multiple R installations.

```
brew tap r-lib/rig
brew install --cask rig

## e.g., M1/M2 mac users, install the arm version
rig install 4.2-arm64

## start Rstudio using a specific R version:
rig rstudio 4.2-arm64
```

Ubuntu Linux

1. Add the CRAN apt repository to get the required version of R.

```
## add R repository
sudo sh -c 'echo "deb https://cran.rstudio.com/bin/linux/ubuntu
focal-cran40/" > \
/etc/apt/sources.list.d/cran.list'
sudo apt-key adv --keyserver keyserver.ubuntu.com \
--recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9

sudo apt-get update
```

2. Install R version 4.2.3

To install previous versions of R see <https://github.com/achubaty/r-config/blob/master/using-multiple-R-versions-on-linux.Rmd> or use rig (<https://github.com/r-lib/ri>).

2. Download and install the latest version of Rstudio from <https://www.rstudio.com/products/rstudio/download/>.

Windows users should install Rstudio as *administrator*.

3. (optional) On Linux, configure ccache to speed up R package re-installation and updates³.

```
## configure ccache for R package installation
mkdir -p ~/.ccache
mkdir -p ~/.R
{ echo 'VER='; \
  echo 'CCACHE=ccache'; \
  echo 'CC=$(CCACHE) gcc$(VER)'; \
  echo 'CXX=$(CCACHE) g++$(VER)'; \
```

³http://dirk.eddelbuettel.com/blog/2017/11/27/#011_faster_package_installation_one

```
echo 'CXX11=$(CCACHE) g++$(VER)'; \  
echo 'CXX14=$(CCACHE) g++$(VER)'; \  
echo 'FC=$(CCACHE) gfortran$(VER)'; \  
echo 'F77=$(CCACHE) gfortran$(VER)'; } >>  
~/R/Makevars  
{ echo 'max_size = 5.0G'; \  
echo 'sloppiness = include_file_ctime'; \  
echo 'hash_dir = false'; } >> ~/.ccache/ccache.conf
```

1.2 Getting the code

All modules are written in R and all model code was developed collaboratively using GitHub (<https://github.com>), with each module contained in its own repository. Code that is shared among modules was bundled into R packages, and hosted in on GitHub repositories. All package code is automatically and regularly tested using cross-platform continuous integration frameworks to ensure the code is reliable and free of errors.

```
mkdir -p ~/GitHub  
cd ~/GitHub  
  
## get development branch (app and deploy are private  
submodules)  
git clone --recurse-submodules \  
-j8 https://github.com/PredictiveEcology/LandWeb
```

Windows users should ensure the GitHub/ directory is accessible at both ~GitHub and ~Documents/GitHub by creating a directory junction:

```
mklink /J C:\\\\Users\\\\username\\\\Documents\\\\GitHub  
C:\\\\Users\\\\username\\\\GitHub
```

TABLE 1.1: LandWeb project directory structure

directory	description
R/	additional R helper scripts
cache/	all per-run and per-study area cache files stored here
docker/	Dockerfiles, scripts, and documentation
docs/	rendered model and app documentation
inputs/	all model data inputs stored here
m/	module code (git submodules)
manual/	raw files for generating documentation manual
outputs/	all per-run model outputs stored here
renv/	project package management directory

1.3 Project directory structure

Model code is organized by the following directories and summarized in the table below.

NOTE: it may be useful to store data in a different location, but to map this location back to the e.g., `cache/`, `inputs/`, and/or `outputs/` directories using symbolic links. See R's `?file.link` to set these up on your machine.

1.4 Updating the code

After having cloned the LandWeb code repository, users can keep up-to-date using their preferred graphical git tools (*e.g.*, GitKraken) or from the command line.

1.4.1 Using GitKraken

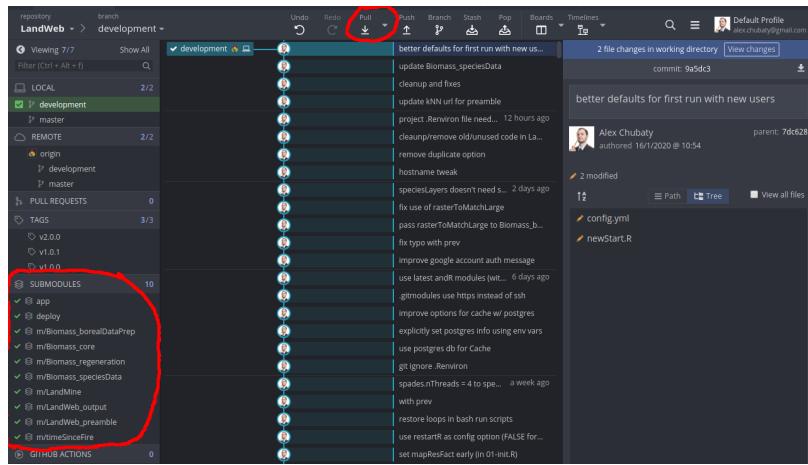


FIGURE 1.1: Screenshot showing code commits in Git Kraken. The submodules pane is highlighted on the bottom left.

1. Open the LandWeb repo, and after a few moments you will see the commit history update to reflect the latest changes on the server.
2. ‘Pull’ in the latest changes to this repo, noting that the status of the git submodules (left hand side) may change.
3. If any submodules have changed status, for each one, right-click and select ‘Update’.

1.4.2 Using the command line

WARNING: experienced git users only!

```
git pull
git submodule update
```

1.5 Data requirements

In order to access and use the proprietary data in LandWeb simulations, you will need to be granted access to the shared Google Drive directory. During first-run of the model, all required data will be downloaded to the `inputs/` directory.

To request access, please contact Alex Chubaty (achubaty@for-cast.ca⁴).

1.6 Getting help

- <https://github.com/PredictiveEcology/LandWeb/issues>

⁴<mailto:achubaty@for-cast.ca>

2

Running LandWeb

1. Launch Rstudio and open the LandWeb Rstudio project (`LandWeb.Rproj`);
2. Open the file `00-global.R` and run each line in sequence, responding to any prompts as required.

Before you can run the model, you first need to install the packages required for the project by restoring from the project's snapshot file.

```
options(renv.config.mran.enabled = FALSE)
renv::restore()
```

2.1 Model setup and configuration

The default settings for study area, model version, and scenario are defined in `01a-globalvars.R`. These defaults are defined as ‘dot-variables’ (e.g., `.studyAreaName`) and can be set externally to the main script (e.g., if `.studyAreaName` is defined before running `01a-globalvars.R`, then the user-set value will be used, rather than the default defined in that script).

Advanced setup and model run customization is described in [Advanced setup](#).

2.1.1 Select a study area

The model can be run over the entire study area, for certain individual provinces (currently only AB, SK, MB, NWT), or groups of predefined FMAs (see Fig. 1 and Table 1).

2.1.1.1 FMA boundaries

Currently, only a subset of the FMAs within the LandWeb study area are pre-defined to be run on their own (i.e., without needing to run the model over the entire study area; see Table 2).

```
## Saving 6.5 x 4.5 in image
```

FMA boundaries within the LandWeb study area

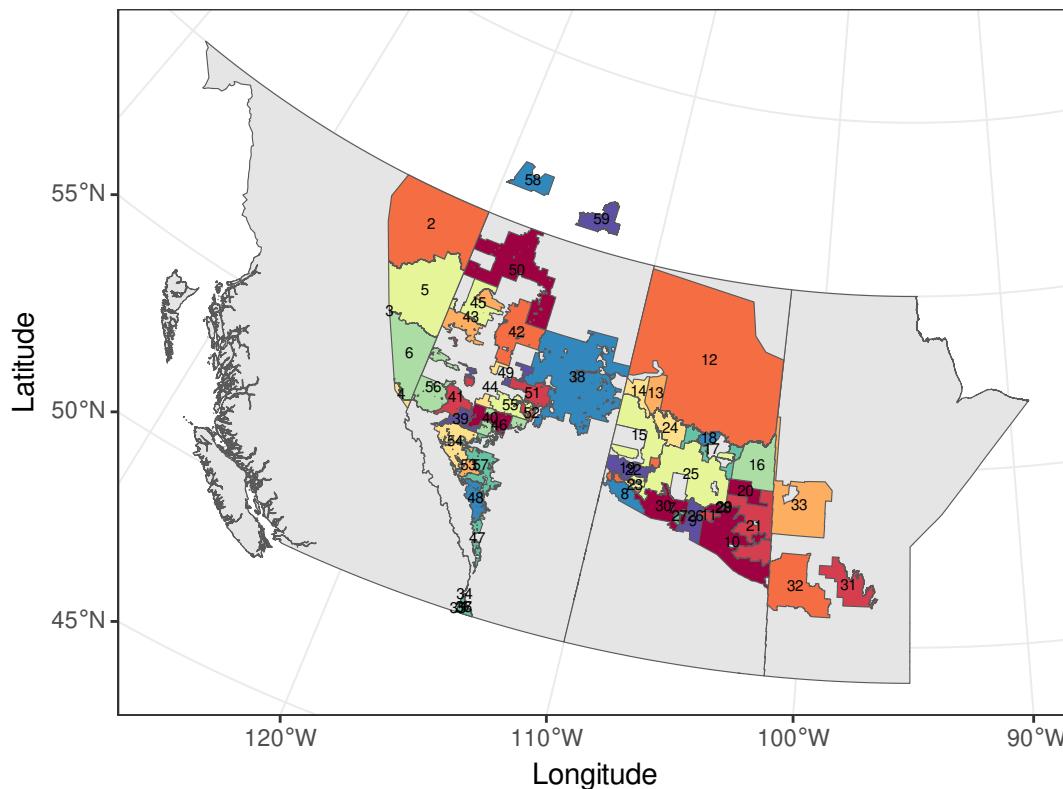


FIGURE 2.1: FMAs within the LandWeb study area, identified by polygon IDs, which are described in Table 1. Polygon fill colours are used only to differentiate neighbouring FMAs.

TABLE 2.1: FMA polygon IDs (from Figure 1) and their corresponding FMA names.

ID	Name
1	Cranbrook
2	Fort Nelson
3	Mackenzie
4	Prince George
5	Fort St. John
6	Dawson Creek
7	Island Forests
8	Turtleford
9	Prince Albert
10	Kelvington
11	Island Forests
12	Northern Reconnaissance
13	Turnor East
14	Turnor West
15	Mistik
16	Mee-Toos
17	Kitsaki Zelensky
18	Nemeiben
19	Meadow Lake Fringe
20	Suggi Lowlands
21	Pasquia-Porcupine
22	Meadow Lake OSB
23	L M Wood Products
24	North West
25	Prince Albert FMA
26	Island Forests
27	Island Forests
28	Island Forests
29	Island Forests
30	Spiritwood
31	INTERLAKE
32	Mountain
33	SASKATCHEWAN RIVER

TABLE 2.1: FMA polygon IDs (from Figure 1) and their corresponding FMA names. (*continued*)

ID	Name
34	Cranbrook
35	Cranbrook
36	Cranbrook
37	Cranbrook
38	ALPAC Forest Products Incorporated
39	ANC Timber Ltd.
40	Blue Ridge Lumber Inc.
41	Canadian Forest Products Ltd.
42	Daishowa-Marubeni International Ltd. (East)
43	Daishowa-Marubeni International Ltd. (West)
44	West Fraser Mills Ltd. and Tolko Industries Ltd.
45	Manning Diversified Forest Products Ltd.
46	Millar Western Forest Products Ltd.
47	Spray Lake Sawmills (1980) Ltd.
48	Sundre Forest Products Inc.
49	Tolko Industries Ltd. (High Prairie)
50	Tolko Industries Ltd., Footner Forest Products Ltd. and La Crete Sawmills Ltd.
51	Tolko Industries Ltd., Vanderwell Contractors (1971) Ltd. and West Fraser Mills Ltd. (Slave Lake)
52	Vanderwell Contractors (1971) Ltd.
53	West Fraser Mills Ltd. (Edson)
54	West Fraser Mills Ltd. (Hinton)
55	West Fraser Mills Ltd. (Slave Lake)
56	Weyerhaeuser Company Limited (Grande Prairie)
57	Weyerhaeuser Company Limited (Pembina Timberland)
58	Fort Providence
59	Fort Resolution

2.1.1.2 Choosing a study area

The model can be run on any of several pre-defined study areas summarized in the table below. To select one of these predefined study areas, set `.studyAreaName` to use one of the following, corresponding to the polygon IDs in the map above.

TABLE 2.2: Model study areas with corresponding FMA polygon IDs (from Figure 1).

studyAreaName	ID	Description
ANC	39	ANC Timber Ltd.
Blueridge	40	Blueridge Lumber Inc.
DMI	42, 43	Mercer Peace River Pulp Ltd. (formerly DMI)
Edson	53	West Fraser Mills Ltd. (Edson)
FMANWT	59	Fort Resolution
FMANWT2	58	Fort Providence
LP_BC	5, 6	Lousiana Pacific (British Columbia)
LP_MB	32	Lousiana Pacific (Manitoba)
Manning	45	Manning Diversified Forest Products Ltd.
MillarWestern	46	Millar Western Forest Products Ltd.
Mistik	15	Mistik
MPR	42, 43	Mercer Peace River Pulp Ltd. (formerly DMI)
Sundre	48	Sundre Forest Products Inc.
Tolko_AB_N	50	Tolko Industries Ltd. (Alberta North)
Tolko_AB_S	44, 49, 51	Tolko Industries Ltd. (Alberta South)
Tolko_SK	22	Tolko (Saskatchewan)
Vanderwell	51, 52	Vanderwell Contractors (1971) Ltd.
WestFraser_N	44, 51, 55	West Fraser Mills Ltd. (Slave Lake)
WestFraser_S	53, 54	West Fraser Mills Ltd. (Edson + Hinton)
WeyCo_GP	56	Weyerhauser Company Ltd. (Grand Prairie)
WeyCo_PT	57	Weyerhauser Company Ltd. (Pembina Timberland)
WeyCo_SK	21	Weyerhauser Company Ltd. (Pasquia-Porcupine)

To run LandWeb over an entire province use one of `provAB`, `provMB`, `provNWT`, or `provSK`.

To run the entire LandWeb study area, use `LandWeb`.

2.1.2 Select a scenario

In version 2.0.0 of the LandWeb model, seed dispersal distances needed to be adjusted to ensure sufficient regeneration following fire. These adjustments cause the model to behave more like a state-transition model, rather than a process-based one.

Version 3.0.0 relaxes these parameter forcings to behave like the standard LANDIS-II model.

The dispersal scenario is principally set via `.version` but can be overridden is set via `.dispersalType`:

Dispersal Scenario	Description
<code>default (v3)</code>	default LANDIS-II dispersal
<code>aspen</code>	limit seed dispersal to deciduous only
<code>high (v2)</code>	high seed dispersal of all species (used for v2.0.0 runs)
<code>none</code>	no seed dispersal (all species)

Additionally, v2.0.0 of the model uses adjusted fire return intervals (FRI) and log-adjusted rates of spread (ROS). Version 3.0.0 uses the LandMine defaults.

The fire scenario is principally set via `.version` but can be overridden by specifying `.ROStype`:

Fire Scenario	Description
<code>default (v3)</code>	default rate of spread values
<code>burny</code>	Increases the flammability of non-forest types to facilitate fire spread in landscapes with discontinuous fuels.
<code>equal</code>	Set all rates of fire spread equal to each other (no vegetation differences)
<code>log (v2)</code>	Reduce the rates of spread but keep the magnitude of vegetation differences

To define a scenario to run, select one dispersal scenario and one fire scenario from the tables above. All LandWeb v2.0.0 runs from 2019 were run

using `.dispersalType = "high"` and `.ROStype = "log"`. Using v3.0.0 of the model, `.dispersalType = "default"` and `.ROStype = "default"`.

2.1.3 Replication

To run multiple replicates of a given run, set `.rep` to an integer corresponding to the replicate id. All replicate runs use a different random seed, and this seed is saved as on output for reuse in the event that a replicate needs to be rerun. To rerun a replicate using a different seed, be sure to delete that run's `seed.rds` file. The seed used is also saved in human-readable `seed.txt` file.

2.2 Running the model

NOTE: The first time the model is run, it will automatically download additional data and install additional R packages, which can take some time to complete.

2.2.1 Interactive R session

When working in an R session, be sure to set the working directory to the LandWeb project directory. The first time running the model, open the file `00-global.R`, and step through each line to ensure any prompts etc. are answered correctly.

```
source("00-global.R")
```

When authenticating with Google Drive, be sure to check the box to allow access to files.

2.2.2 Commandline interface

In addition to running the model in an interactive R session, we provide a command line interface to run replicates of the model for the study areas defined above (i.e., batch mode).

For example, to run replicate number 7 of the model at 250m resolution using FRI multiple of 1 for Alberta FMU L11, use:

```
cd ~/GitHub/LandWeb

## ./run_fmu.sh <FMU> <FRI> <RES> <REP>
./run_fmu.sh L11 1 250 7
```

FMU command line runs do not use modified dispersal nor fire scenarios, and thus only require the study area, replicate, and FRI multiple to be defined.

To run replicate number 7 of the model for the entire province of Alberta, use:

```
cd ~/GitHub/LandWeb

## ./run_fma.sh <FMU> <REP>
./run_fma.sh provAB 7
# ./run_fma_win.sh provAB 7 ## if on Windows!
```

FMA command line runs use highDispersal_logROS scenarios at 250m resolution, and thus only require the study area and replicate to be defined.

To run the entire LandWeb study area, only a replicate number needs to be passed. For example:

```
## ./run_landweb.sh <REP>
./run_landweb.sh 7
```

LandWeb command line run set the dispersal and fire scenarios as above, and the pixel resolution (.pixelSize) at 250m, and thus only require the study area and replicate to be defined.

2.3 Post-processing analyses

After having run several reps of the model on a given study area, results are combined in subsequent post-processing analyses to generate the following outputs for each set of reporting polygons within the study area:

- boxplots of leading vegetation cover;
- histograms of leading vegetation cover;
- histograms of large patches.

To run processing, use `.mode = "postprocess"` and be sure to set the number of replicates run in the config.

2.4 Advanced setup

2.4.1 Customizing model run configuration

See `SpaDES.config/R/config.landweb.R1`.

(TODO)

2.4.2 Cache backend

Simulation caching is provided by the `reproducible` and `SpaDES.core` packages, and is enabled by default.

The default cache uses a SQLite database backend and stores cache files in `cache/`. However, other database backends can also be used, and advanced users running multiple parallel simulations may wish to set up and use a PostgreSQL database for this cache.

See <https://github.com/PredictiveEcology/SpaDES/wiki/Using-alternate-database-backends-for-Cache> and ensure the following options are added to your user-specific config in `02a-user-config.R`:

```
reproducible.cacheSaveFormat = "qs",
reproducible.conn = SpaDES.config::dbConnCache("postgresql"),
```

2.4.3 Speeding up disk-based operations

Caching and other disk-based file operations benefit from using an solid state drive (SSD) instead of a conventional spinning hard drive. Advanced users can move their `cache/` directory to an SSD mountpoint and create a

¹<https://github.com/PredictiveEcology/SpaDES.config/blob/development/R/config.landweb.R>

symlink to this location in the project directory. Likewise, users can configure a scratch path for temporary raster file operations to point to an SSD location.

2.5 Additional Resources

Resources for (re)learning R and spatial data:

- <https://r-spatial.org>

Study areas

Lorem ipsum ... (TODO)



3

LandWeb_preamble Module

(ref:LandWeb_preamble) *LandWeb_preamble*

3.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca¹ [aut, cre], Alex M. Chubaty achubaty@for-cast.ca² [aut], Ceres Barros cbarros@mail.ubc.ca³ [aut]

3.1 Module Overview

3.1.1 Module summary

Set up study areas and parameters for LandWeb simulations.

3.2 Parameters

Provide a summary of user-visible parameters.

¹mailto:eliot.mcintire@nrcan-rncan.gc.ca

²mailto:achubaty@for-cast.ca

³mailto:cbarros@mail.ubc.ca

parameterName	parameterClass	default	min	max	parameterDescription
bufferDist	numeric	20000	20000	20000	Study area buffer distance (m) used to make 'studyArea'.
bufferDistLarge	numeric	50000	50000	50000	Study area buffer distance (m) used to make 'studyAreaLarge'.
percentageLarge	logical	FALSE	NA	NA	TRUE levels all species to remove, setting 'percentage_min' to zero, 'percentage_max' to 100, and 'reservoirProb' to 1.0.
percentageMin	numeric	0	0.5	1	Percentage of species to remove.
percentageMax	numeric	1	0.5	0	Percentage of species to remove.
plotStyle	character	default	NA	NA	One of 'super', 'high', 'none', or 'default'.
plotTime	character	NA	NA	NA	Plot time.
plotSize	numeric	350	NA	NA	Pixel size in metres. Should be one of 25, 125, 50, 25.
ROSType	character	default	NA	NA	Rate of spread preset to use. One of 'burn', 'square', 'log', or 'default'.
treeClassesT20	character	NA	NA	NA	The tree classes in the T20? layer that will become 'trees' from the perspective of LandWeb biomass (e.g., 'Birch').
treeClassesT20Replace	numeric	14, 25, 56	0	99	The tree classes in the T20? layer that will become 'trees' from the perspective of LandWeb biomass (e.g., 'Birch').
plotInitialTime	numeric	0	NA	NA	This describes the simulation time at which the first plot event should occur.
plotIntervalTime	numeric	NA	NA	NA	This describes the simulation time at which the next plot event should occur.
plots	character	object	NA	NA	Passed to 'tspars' in 'Plot' (see 'Plot'). There are a few plots that are made within this module, if any. Note that plots (or their data) seeing will ONLY occur at 'Vndomin'. If NA, plotting is turned off completely (this includes plot saving).
savelInitialTime	numeric	NA	NA	NA	This describes the simulation time at which the first save event should occur.
savelIntervalTime	numeric	NA	NA	NA	This describes the simulation time at which the next save event should occur.
sslVerify	integer	64	NA	NA	Passed to 'httpConfig\$sslVerify' when downloading FNN (NFI) datasets. Set to 0 if necessary to bypass checking the SSL certificate (this may be necessary when NFI's website SSL certificate is not correctly configured).
studyAreaName	character	NA	NA	NA	Human-readable name for the study area used. If NA, a hash of studyAreaLayer will be used.
useCache	logical	FALSE	NA	NA	Should this entire module be run with caching activated? This is generally intended for data-type modules, where stochasticity and time are not relevant.

objectName	objectClass	desc	sourceURL
canProvs	SpatialPolygonsDataFrame	Canadian provincial boundaries shapefile	NA

Section	Topic	Description
1	Introduction	Provides an overview of the project, its purpose, and the scope of the report.
2	Methodology	Describes the research methods and data collection process used in the study.
3	Data Analysis	Presents the results of the analysis, including key findings, trends, and insights derived from the data.
4	Conclusion	Summarizes the main conclusions drawn from the study and their implications.
5	Recommendations	Offers practical suggestions for future actions or improvements based on the findings.
6	Appendices	Includes additional supporting information such as tables, figures, and references.

3.3 Data dependencies

3.3.1 Input data

Description of the module inputs.

3.3.2 Output data

Description of the module outputs.

3.4 Links to other modules

Originally developed for use with the LandR Biomass suite of modules, with LandMine fire model. ## References

Vegetation submodel

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 subsequent chapters.



4

LandR Biomass_speciesData Module

This documentation is work in progress. Please report any discrepancies or omissions at https://github.com/PredictiveEcology/Biomass_speciesData/issues.

4.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca¹ [aut, cre], Alex M. Chubaty achubaty@for-cast.ca² [aut], Ceres Barros cbarros@mail.ubc.ca³ [aut]

4.1 Module Overview

4.1.1 Module summary

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

4.1.2 Module inputs and parameters at a glance

Below are the full list of input objects (Table 4.1) and parameters (Table 4.2) that *Biomass_speciesData* expects. Of these, the only input that **must** be provided (i.e., *Biomass_speciesData* does not have a default for) is `studyAreaLarge`.

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

¹<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

²<mailto:achubaty@for-cast.ca>

³<mailto:cbarros@mail.ubc.ca>

TABLE 4.1: List of *Biomass_speciesData* input objects and their description.

objectName	desc
rasterToMatchLarge	a raster of 'studyAreaLarge' in the same resolution and projection the simulation's. Defaults to the using the Canadian Forestry Service, National Forest Inventory, kNN-derived stand biomass map.
rawBiomassMap	total biomass raster layer in study area. Only used to create 'rasterToMatchLarge' if necessary. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in tonnes/ha), unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.
sppColorVect	A named vector of colors to use for plotting. The names must be in sim\$sppEquiv[[sim\$sppEquivCol]], and should also contain a color for 'Mixed'
sppEquiv	table of species equivalencies. See 'LandR::sppEquivalencies_CA'.
sppNameVector	an optional vector of species names to be pulled from 'sppEquiv'. Species names must match 'P(sim)\$sppEquivCol' column in 'sppEquiv'. If not provided, then species will be taken from the entire 'P(sim)\$sppEquivCol' column in 'sppEquiv'. See 'LandR::sppEquivalencies_CA'.
studyAreaLarge	Polygon to use as the parametrisation study area. Must be provided by the user. Note that 'studyAreaLarge' is only used for parameter estimation, and can be larger than the actual study area used for LandR simulations (e.g. larger than 'studyArea' in LandR Biomass_core).
studyAreaReporting	multipolygon (typically smaller/unbuffered than 'studyAreaLarge' and 'studyArea' in LandR Biomass_core) to use for plotting/reporting. If not provided, will default to 'studyAreaLarge'.

TABLE 4.2: List of *Biomass_speciesData* parameters and their description.

paramName	paramDesc
coverThresh	The minimum % cover a species needs to have (per pixel) in the study area to be considered present
dataYear	Passed to 'pasteO('prepSpeciesLayers_', types)' function to fetch data from that year (if applicable). Defaults to 2001 as the default kNN year.
sppEquivCol	The column in 'sim\$sppEquiv' data.table to group species by and use as a naming convention. If different species in, e.g., the kNN data have the same name in the chosen column, their data are merged into one species by summing their % cover in each raster cell.
types	The possible data sources. These must correspond to a function named pasteO('prepSpeciesLayers_', types). Defaults to 'KNN' to get the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from year 'dataYear', using the 'LandR::prepSpeciesLayers_KNN' function (see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for details on these data). Other currently available options are 'ONFRI', 'CASFRI', 'Pickell' and 'ForestInventory', which attempt to get proprietary data - the user must be granted access first. A custom function can be used to retrieve any data, just as long as it is accessible by the module (e.g., in the global environment) and is named as pasteO('prepSpeciesLayers_', types).
.vegLeadingProportion	a number that defines whether a species is leading for a given pixel. Only used for plotting.
.plotInitialTime	This describes the simulation time at which the first plot event should occur
.plotInterval	This describes the simulation time interval between plot events
.plots	Passed to 'types' in 'Plots' (see ?Plots'). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at 'end(sim)'. If 'NA', plotting is turned off completely (this includes plot saving).
.saveInitialTime	This describes the simulation time at which the first save event should occur
.saveInterval	This describes the simulation time interval between save events
.sslVerify	Passed to 'httr::config(ssl_verify/peer = P(sim)\$sslVerify)' when downloading KNN (NFI) datasets. Set to 0L if necessary to bypass checking the SSL certificate (this may be necessary when NFI's website SSL certificate is not correctly configured).
.studyAreaName	Human-readable name for the study area used. If NA, a hash of 'studyAreaLarge' will be used.
.useCache	Controls cache; caches the init event by default
.useParallel	Used in reading csv file with fread. Will be passed to data.table::setDTthreads.

TABLE 4.3: List of *Biomass_speciesData* output objects and their description.

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

4.1.3 Events

Biomass_speciesData only runs two events:

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

4.1.4 Module outputs

The module produces the following outputs (Table 4.3):

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

4.1.5 Links to other modules

Intended to be used with other LandR data modules (e.g., *Biomass_borealDataPrep*) that require species cover data and the LandR forest simulation *Biomass_core* module. You can see all potential module linkages within the LandR ecosystem here⁴. Select *Biomass_speciesData* from the drop-down menu to see linkages.

4.1.6 Getting help

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

⁴https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

4.2 Module manual

4.2.1 Detailed description

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

Currently, the module can access the Canadian Forest Inventory forest attributes kNN dataset [the default; [5](#)], the Common Attribute Schema for Forest Resource Inventories [CASFRI; [9](#)] dataset, the Ontario Forest Resource Inventory (ONFRI), a dataset specific to Alberta compiled by Paul Pickell, and other Alberta forest inventory datasets. However, **only the NFI kNN data are freely available** – access to the other datasets must be granted by module developers and data owners, and a Google account is required. Nevertheless, the module is flexible enough that any user can use it to process additional datasets, provided that an adequate R function is passed to the module (see `types` parameter details in [Parameters](#))

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

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` input in [Input objects](#) for information on how species groups are defined.

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

4.2.2 Initialization, inputs and parameters

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

The module defaults to processing cover data 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 4.4; see `?LandR::sppEquivalencies_CA` for more information):

4.2.2.1 Input objects

Biomass_speciesData requires the following input data layers

Of the inputs in Table 4.5, the following are particularly important and deserve special attention:

- `studyAreaLarge` – the polygon defining the area for which species cover data area desired. It can be larger (but never smaller) than the study area used in the simulation of forest dynamics (i.e., `studyArea` object in *Biomass_core*).
- `sppEquiv` – a table of correspondences between different species naming conventions. This table is used across several LandR modules, including *Biomass_core*. It is particularly important here because it will determine whether and how species (and their cover layers) are merged, if this is desired by the user. For instance, if the user wishes to simulate a generic *Picea spp.* that includes, *Picea glauca*, *Picea mariana* and *Picea engelmannii*, they will need to provide these three species names in the data column (e.g., `kNN` if obtaining forest attribute kNN data layers from the Canadian Forest Inventory), but the same name (e.g., “*Pice_Spp*”) in the column chosen for the naming convention used throughout the simulation (the `sppEquivCol` parameter); see Table 4.6 for an example).

4.2.2.2 Parameters

Table 4.7 lists all parameters used in *Biomass_speciesData* and their detailed information.

Of the parameters listed in Table 4.7, the following are particularly important:

- `coverThresh` – integer. Defines a minimum % cover value (from 0-100) that

TABLE 4.4: 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 saccharinum</i> *	Silver 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>Fraxinus spp.</i> *	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 x glauca</i> *	Engelmann's spruce
* <i>Picea engelmannii x glauca</i> *	Engelmann's spruce
* <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</i> *	Balsam poplar
* <i>Populus balsamifera v. balsamifera</i> *	Balsam poplar
* <i>Populus tremuloides</i> *	Black cottonwood

TABLE 4.5: List of *Biomass_speciesData* input objects and their description.

TABLE 4.6: 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_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 x glauca</i> *	Pice_Eng_Gla	Pice_Spp	* <i>Picea spp.</i> *
* <i>Picea engelmannii x glauca</i> *	Pice_Eng_Gla	Pice_Spp	* <i>Picea spp.</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> *

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

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

TABLE 4.7: List of *Biomass_speciesData* parameters and their description.

.....

with overlapping values from *CASFRI* layers and values from *KNN* and *CASFRI* layers are replaced with overlapping values of *ForestInventory* layers.

4.2.3 Simulation flow

The general flow of *Biomass_speciesData* processes is:

1. Download (if necessary) of and spatial processing of species cover layers from the first data source listed in the `types` parameter. Spatial processing consists in sub-setting the data to the area defined by `studyAreaLarge` and ensuring that the spatial projection and resolution match those of `rasterToMatchLarge`. After spatial processing, species layers that have no pixels with values \geq to the `coverThresh` parameter are excluded.
2. If more than one data source is listed in `types`, the second set of species cover layers is downloaded and processed as above.
3. The second set of layers is assumed to be the highest quality dataset and used to 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. A summary of number of pixels with forest cover are calculated (`treedand numTreed` output objects; see [Module outputs](#)).

4.3 Usage example

4.3.1 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))
```

4.3.2 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)
```

4.3.3 Setup simulation

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

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

```
# 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))

```

4.3.4 Run module

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

```

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.

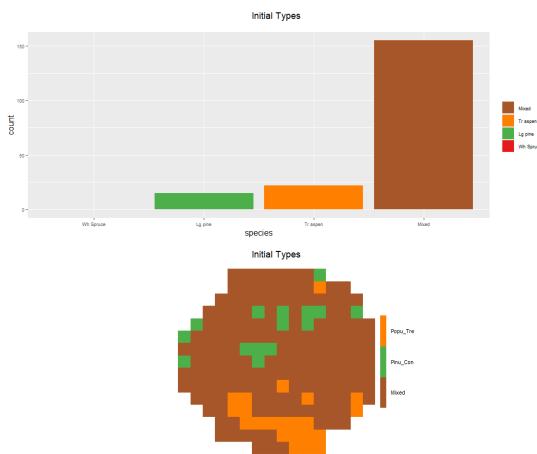


FIGURE 4.1: *Biomass_speciesData* automatically generates a plot of species dominance and number of presences in the study area when ‘.plotInitialTime=1’ is passed as an argument.

4.4 References

- [5] 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 – 6ACED147A990⁵. URL: <http://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990>.

⁵<https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990>

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

5

LandR Biomass_borealDataPrep Module

This documentation is work in progress. Please report any discrepancies or omissions at https://github.com/PredictiveEcology/Biomass_borealDataPrep/issues.

5.0.0.1 Authors:

Yong Luo Yong.Luo@gov.bc.ca¹ [aut], Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca² [aut, cre], Ceres Barros ceres.barros@ubc.ca³ [aut], Alex M. Chubaty achubaty@for-cast.ca⁴ [aut]

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

5.1.2 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. Nev-

¹<mailto:Yong.Luo@gov.bc.ca>

²<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

³<mailto:ceres.barros@ubc.ca>

⁴<mailto:achubaty@for-cast.ca>

ertheless, 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 data⁵.

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

5.1.3 Links to other modules

Biomass_borealDataPrep is intended to be used with *Biomass_core*⁶, but can be linked with other data modules that prepare inputs. See here⁷ for all available modules in the LandR ecosystem and select *Biomass_borealDataPrep* from the drop-down menu to see potential linkages.

- *Biomass_core*⁸: core forest dynamics simulation module. Used downstream from *Biomass_borealDataPrep*;
- *Biomass_speciesData*⁹: 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*¹⁰: calibrates four-species level traits using permanent sample plot data (i.e., repeated tree biomass measurements) across Western Canada. Used downstream from *Biomass_borealDataPrep*.

⁵Raw data layers downloaded by the module are saved in ‘dataPath(sim)’, which can be controlled via ‘options(reproducible.destinationPath = ...)’.

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

⁷https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

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

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

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

5.2 Module manual

5.2.1 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 discussed in detail in [Data acquisition and treatment](#).

After 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 ecolocation is termed ecoregionGroup 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.

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`.

5.2.2 Data acquisition and treatment

The only two objects that the user must supply are shapefiles that define the study area used to derive parameters (`studyAreaLarge`) and the study area where the simulation will happen (`studyArea`). The two objects can be identical if the user chooses to parametrise and run the simulations in the same area. If not identical, `studyArea` must be fully within `studyAreaLarge`. If `studyAreaLarge` and `studyArea` are in Canada, the module can automatically estimate and prepare all input parameters and objects for `Biomass_core`, as the default raw data are FAIR data [*sensu* 24] 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).

5.2.2.1 Defining simulation pixels and ecolocations

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

els). By default it uses land-cover classes from the Land Cover of Canada 2010 v1 map¹¹, 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 `forestedLCCclasses`).

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 `LCCclassesToReplaceNN`) that will be reclassified into a “stable” forested class (defined via the parameter `forestedLCCclasses`). 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 Canada¹² (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.

5.2.2.2 Species cover

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

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

¹¹<http://www.cec.org/north-american-environmental-atlas/land-cover-2010-modis-250m/>

¹²<https://open.canada.ca/data/en/dataset/3ef8e8a9-8d05-4fea-a8bf-7f5023d2b6e1>

imagery from 2001, obtained from the Canadian National Forest Inventory [5] – hereafter ‘kNN species data’.

5.2.2.3 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 [5] 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](#)).

5.2.2.4 Replacing initial biomass and age within known fire perimeters

Biomass_borealDataPrep can use fire perimeters to correct stand ages. To do so, it downloads the latest fire perimeter data from the Canadian Wildfire Data Base¹³ 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.

Taking two independent datasets for stand age (fire perimeters) and stand biomass (derived from MODIS satellite imagery) can cause discrepancies (e.g. stand age = 5 and aboveground biomass = 10000 m²/ha). This may be due to errors coming from a) a stand replacing disturbance that reset age to zero a few years before, but the biomass layer was not zeroed, or b) the disturbance was not stand-replacing (leaving biomass), but age was still zeroed. This means that either, aboveground biomass is wrong or age is.

Options to address this include 1) get better data for these two variables that do not contradict one another (not currently available to us) or 2) estimate one or the other. There is no obvious way to decide which one is incorrect, unless there is an independent data source.

In the current *Biomass_borealDataPrep* module version, we chose to correct both. If `P(sim)$fireURL` is provided and `P(sim)$overrideBiomassInFires` is TRUE, fire perimeters are used as the source of information for age, and *Biomass_core* is used to generate corresponding biomass values based on the estimated growth parameters and known species presence/absence (from the species cover layers).

This assumes that 1) recorded fires were stand-replacing, and so time since fire is the new stand age and 2) that the first year of the simulation is later than the first fire year in the fire perimeter data. The biomass spin-up with *Biomass_core* is only run in pixels where stand ages were corrected, for as long as the new stand age (i.e., the time since last fire). All species start with age = 0 and biomass = 0, and grow until time since last fire is achieved. The resulting species biomass is used as the initial biomass values for each species cohort in the actual simulation.

If the user does not want to perform this imputation, this step can be bypassed by setting the parameter `P(sim)$overrideBiomassInFires` to FALSE or `P(sim)$fireURL` to NULL or NA.

Also pixels that suffered *any kind* of data imputation (e.g., the age corrections

¹³<https://cwfis.cfs.nrcan.gc.ca/datamart>

detailed in the [previous section](#)) can be excluded from the simulation by setting `P(sim)$rmImputedPix == TRUE`.

5.2.2.5 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 repository¹⁴). 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 [13], which are by default filtered to the Boreal Shield West (BSW), Boreal Plains (BP) and Montane Cordillera Canadian Ecozones (via `P(sim)$speciesTableAreas`). Most trait values do not vary across these eco-zones 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 [6], 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`).

¹⁴https://github.com/dcyr/LANDIS-II_IA_generalUseFiles

5.2.2.6 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 table¹⁵.

5.2.3 Parameter estimation/calibration

5.2.3.1 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)$deciduousCoverDiscount` based on Northwest Territories data. However, the user can chose to re-estimate it by setting `P(sim)$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.

5.2.3.2 Maximum biomass and maximum aboveground net primary productivity

Biomass_borealDataPrep statistically estimates maximum biomass (`maxB`) and 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

¹⁵https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt

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 r²) and by the entire model (conditional r²) – 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 convergence 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 disabled (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 ~ log(age) + 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, $\max B$ 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 $\max B$ (Fig. 5.1) and steep increases in species biomasses during the first years of the simulation (Fig. 5.2).

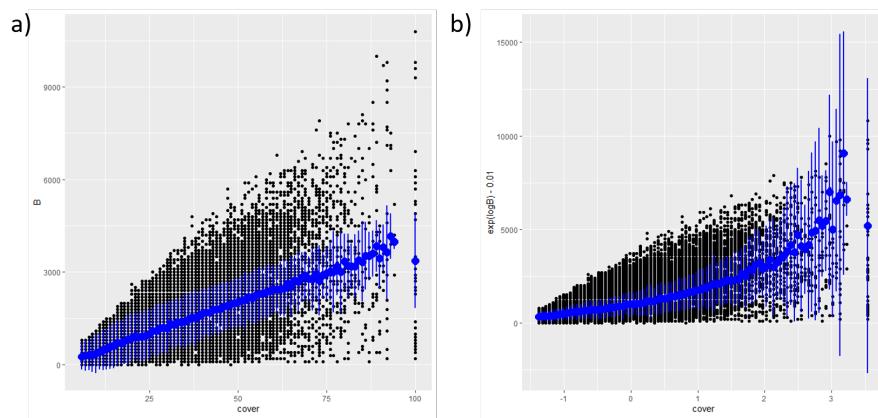


FIGURE 5.1: Modelling biomass as a linear vs. exponential relationship. a) ‘modelBiomass’ as ‘ $B \sim \logAge * \text{speciesCode} + \text{cover} * \text{speciesCode} + (\logAge + \text{cover} | \text{ecoregionGroup})$ ’. b) ‘modelBiomass’ as ‘ $\log B \sim \logAge * \text{speciesCode} + \text{cover} * \text{speciesCode} + (\logAge + \text{cover} | \text{ecoregionGroup})$ ’. Blue dots are marginal mean B values (back-transformed in b) cross ages with confidence intervals as the bars.

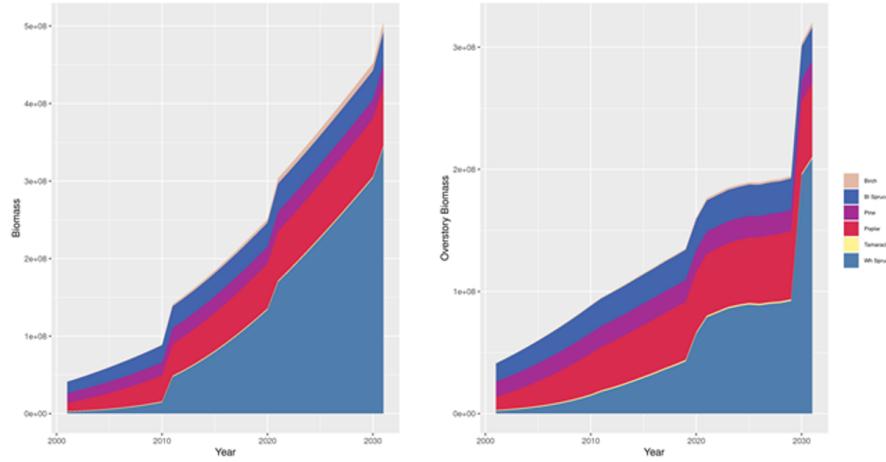


FIGURE 5.2: Thirty years of simulation with ‘maxB’ values estimated from a ‘logB ...‘biomassModel’ (see Fig. reffig:fig-biomassModelLogBtest). The steep increase in such little time is abnormal.

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 `maxB * 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 [21]. 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.

5.2.3.3 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 be-

tween 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 proportion of pixels with % cover > 0 – 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 quickly even for very large sample sizes (e.g., > 20 million points).

SEP is then predicted by species and ecolocation combination and the resulting values are integrated over the length of the succession time step (successionTimestep parameter) as:

$$\text{integratedSEP} = 1 - (1 - \text{estimatedSEP})^{e^{\text{successionTimestep}}} \quad (5.1)$$

This is important, since seed establishment only occurs once at every $\text{P(sim)}\$successionTimestep$, 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 presence probabilities and species' probabilities of resprouting (resproutprob, in the species table) (bounded between 0 and 1):

$$SEP = \text{integratedSEP} * (1 - \text{resproutprob}) \quad (5.2)$$

if $SEP > 1$, then:

$$SEP = 1 \quad (5.3)$$

if $SEP < 0$, then:

$$SEP = 0 \quad (5.4)$$

5.2.3.4 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 repository¹⁶), 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 [12], which are likely driven by higher moisture limitation in the west [10, 14].

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`.

5.2.3.5 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`:

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

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.
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.

5.2.4 Aggregating species

Biomass_borealDataPrep will use the input table sppEquiv and the parameter P(sim)\$sppEquivCol to select the naming convention to use for the simulation (see full list of **input objects** and **parameters** for details). The user

can use this table and parameter to define groupings that “merge” similar 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 name of the species group in sppEquivCol column of the sppEquiv table must be identical for each grouped species.

TABLE 5.1: 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. (continued below)

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

Modelled as

Abies balsamea
Abies lasiocarpa
Picea spp.
Picea spp.
Picea spp.
Picea spp.
Picea spp.
Pinus contorta var. contorta
Pinus contorta

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.

5.2.5 List of input objects

Below are the full lists of input objects (Table 5.3) 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 where the simulation will take place. This input object **must be supplied by the user**.
- *studyAreaLarge* – shapefile. A *SpatialPolygonsDataFrame* with a single polygon determining the where the statistical models for parameter estimation will be fitted. It **must** contain *studyArea* fully, if they are not identical. This object **must be supplied by the user**.

Tables

- `speciesTable` – a table of invariant species traits that must have the following columns (even if not all are necessary to the simulation): “species”, “Area”, “longevity”, “sexualmature”, “shadetolerance”, “fire-tolerance”, “seeddistance_eff”, “seeddistance_max”, “resproutprob”, “resproutage_min”, “resproutage_max”, “postfireregen”, “leaflongevity”, “wooddecayrate”, “mortalityshape”, “growthcurve”, “leafLignin”, “hardsoft”. The order can vary but the column names must be identical. See Scheller and Miranda [20] and *Biomass_core* manual for further detail about these columns.

TABLE 5.3: List of *Biomass_borealDataPrep* input objects and their description.

objectName	objectClass	desc	sourceURL
cloudFolderID	character	The google drive location where cloudCache will store large statistical objects	NA
columnsForPixelGroups	character	The names of the columns in cohortData that define unique pixelGroups. Default is c('ecoregionGroup', 'speciesCode', 'age', 'B')	NA
ecoregionLayer	SpatialPolygonsDataFrame	A SpatialPolygonsDataFrame that characterizes the unique ecological regions (ecoregionGroup) used to parameterize the biomass, cover, and species establishment probability models. It will be overlaid with landcover to generate classes for every ecoregion/LCC combination. It must have same extent and crs as studyAreaLarge. It is superseded by sim\$ecoregionRst if that object is supplied by the user	https://ss.ags.gc.ca/cansis/nedb/ecostrat/district/ecodistrict_shp.zip

objectName	objectClass	desc	sourceURL
ecoregionRaster	RasterLayer	A raster that characterizes the unique ecological regions used to parameterize the biomass, cover, and species establishment probability models. If this object is provided, it will supercede <code>sim\$ecoregionLayer</code> . It will be overlaid with <code>landcover</code> to generate classes for every ecoregion/LCC combination. It must have same extent and crs as <code>rasterToMatchLarge</code> if supplied by user - use reproducible::postProcess. If it uses an attribute table, it must contain the field 'ecoregion' to represent raster values	NA
firePerimeters	RasterLayer	Fire perimeters raster, with fire year information used to 'update' stand age using time since last fire as the imputed value. Only used if <code>P(sim)\$overrideAgeInFires = TRUE</code> . Biomass will also be updated in these pixels if <code>P(sim)\$overrideBiomassInFires = TRUE</code> and the last fire was later than 1985. Defaults to using fire perimeters in the Canadian National Fire Database, downloaded as a zipped shapefile with fire polygons, an attribute (i.e., a column) named 'YEAR', which is used to rasterize to the study area.	https://cfis.cfs.nrcan.gc.ca/downloads/nfdb/fire_poly/current_version/NFDB_poly.zip

objectName	objectClass	desc	sourceURL
rstLCC	RasterLayer	A land classification map in study area. It must be ‘corrected’, in the sense that: 1) Every class must not conflict with any other map in this module (e.g., speciesLayers should not have data in LCC classes that are non-treed); 2) It can have treed and non-treed classes. The non-treed will be removed within this module if <code>P(sim)\$omitNonTreedPixels</code> is TRUE; 3) It can have transient pixels, such as ‘young fire’. These will be converted to the nearest non-transient class, probabilistically if there is more than 1 nearest neighbour class, based on <code>P(sim)\$LCCClassesToReplaceNN</code> . The default layer used, if not supplied, is Canada national land classification in 2010. The metadata (res, proj, ext, origin) need to match <code>rasterToMatchLarge</code> .	NA
rasterToMatch	RasterLayer	A raster of the <code>studyArea</code> in the same resolution and projection as <code>rawBiomassMap</code> . This is the scale used for all outputs for use in the simulation. If not supplied will be forced to match the default <code>rawBiomassMap</code> .	NA
rasterToMatchLarge	RasterLayer	A raster of the <code>studyAreaLarge</code> in the same resolution and projection as <code>rawBiomassMap</code> . This is the scale used for all inputs for use in the simulation. If not supplied will be forced to match the default <code>rawBiomassMap</code> .	NA

objectName	objectClass	desc	sourceURL
rawBiomass Map	RasterLayer	total biomass raster layer in study area . Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map from 2001 (in tonnes/ha), unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata.	http://ftp.maps.cada.ca/pub/nrcan_rnca/Forests/canada/forests-attribute/s_attributes-forsts-canada/2001-atributes-2001/NF_L_MODIS250m_2001_kNN_Structure_Biomass_TotalLiveAboveGround_v1.tif

objectName	objectClass	desc	sourceURL
speciesLayers	RasterStack	cover percentage raster layers by species in Canada species map. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from 2001 using a cover threshold of 10 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata	http://ftp.maps.cdn.ca/pub/nrcan_rnca/Forests/canada/forests-atributes_attributes-2001/
speciesTable	data.table	a table of invariant species traits with the following trait columns: 'species', 'Area', 'longevity', 'sexualmature', 'shadetolerance', 'firer tolerance', 'seeddistance_eff', 'seeddistance_max', 'resproutprob', 'resproutage_min', 'resproutage_max', 'postfireregen', 'leaflongevity', 'wooddecayrate', 'mortalityshape', 'growthcurve', 'leafLignin', 'hardsoft'. Names can differ, but not the column order. Default is from Dominic Cyr and Yan Boulanger's project.	https://raw.githubusercontent.com/dcyr/LANDIS-II_IAGeneralUseFiles/master/speciesTraits.csv

objectName	objectClass	desc	sourceURL
sppColorVec	character	named character vector of hex colour codes corresponding to each species	NA
sppEquiv	data.table	table of species equivalencies. See ?LandR::sppEquivalencies_CA.	NA
sppNameVector	character	an optional vector of species names to be pulled from sppEquiv. Species names must match P(sim)\$sppEquivCol column in sppEquiv. If not provided, then species will be taken from the entire P(sim)\$sppEquivCol column in sppEquiv. See LandR::sppEquivalencies_CA.	NA

objectName	objectClass	desc	sourceURL
standAgeMap	RasterLayer	stand age map in study area. Must have a 'imputedPixID' attribute (a vector of pixel IDs) indicating which pixels suffered age imputation. If no pixel ages were imputed, please set this attribute to integer(0). Defaults to the Canadian Forest Service, National Forest Inventory, kNN-derived biomass map from 2001, unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata	http://ftp.maps.cdn.ca/pub/nrcan_rnca/Forests/canada/forests-attributes_attributes-canada/2001-attributes-2001/NF_L_MODIS250m_2001_kNN_Structure_Stand_Age_v1.tif
studyArea	SpatialPolygonsDataFrame	Polygon to use as the study area. Must be supplied by the user.	NA

objectName	objectClass	desc	sourceURL
studyAreaLarge	SpatialPolygonsDataFrame	multipolygon (potentially larger than studyArea) used for parameter estimation , Must be supplied by the user. If large r than studyArea, it must fully contain it.	NA

5.2.6 List of parameters

Table 5.4 lists all parameters used in *Biomass_borealDataPrep* and their detailed information. All have default values specified in the module's metadata.

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).

TABLE 5.4: List of *Biomass_borealDataPrep* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
biomassModel	call	lme4::lm.NA	NA		Model and formula for estimating biomass (B) from ecoregionGroup (currently ecoregionLayer LandCoverClass), speciesCode, logAge (gives a downward curving relationship), and cover. Defaults to a LMEM, which can be slow if dealing with very large datasets (e.g. 36 000 points take 20min). For faster fitting try P(sim)\$subsetDataBiomassModel == TRUE, or quote(RcppArmadillo:::fastLm(formula = B ~ logAge speciesCode ecoregionGroup + cover speciesCode ecoregionGroup)). A custom model call can also be provided, as long as the 'data' argument is NOT included.
coverModel	call	glm, cbi....	NA	NA	Model and formula used for estimating cover from ecoregionGroup and speciesCode and potentially others. Defaults to a GLMEM if there are > 1 grouping levels. A custom model call can also be provided, as long as the 'data' argument is NOT included
fixModelBiomass	logical	FALSE	NA	NA	should biomassModel be fixed in the case of non-convergence? Only scaling of variables and attempting to fit with a new optimizer (bobyqa, see ?lme4) are implemented at this time.

paramName	paramClass	default	min	max	paramDesc
subsetDataAttempts	integer	3	1	10	How many times should biomassModel be attempted to fit with a new data subset in case of non-convergence? Each time, the data is resampled (if subsetDataBiomassModel = TRUE) and the model re-fit with the original data, scaled variables and/or a different optimizer if fixModelBiomass = TRUE. Model refitting with original data, rescaled variables and/or a new optimizer occurs up to three times for each data subset, regardless of this parameter's value.
subsetDataBiomassModel	logical	NA	NA	NA	the number of samples to use when subsampling the biomass data model (biomassModel); Can be TRUE/FALSE/NULL or numeric; if TRUE, uses 50, the default. If FALSE/NULL no subsetting is done.
coverPctToBiomassModel	logical	NA	NA	NA	Model to estimate the relationship between % cover and % biomass, referred to as P(sim)\$fitDeciduousCoverDiscount It is a number between 0 and 1 that translates % cover, as provided in several databases, to % biomass. It is assumed that all hardwoods are equivalent and all softwoods are equivalent and that % cover of hardwoods will be an overestimate of the % biomass of hardwoods. E.g., 30% cover of hardwoods might translate to 20% biomass of hardwoods. The reason this discount exists is because hardwoods in Canada have a much wider canopy than softwoods.
deciduousCoverDiscount	double	0.8418911	NA	NA	This was estimated with data from NWT on March 18, 2020 and may or may not be universal. Will not be used if P(sim)\$fitDeciduousCoverDiscount == TRUE

paramName	paramClass	default	min	max	paramDesc
fitDeciduousCoverDiscount	logical	FALSE	NA	NA	If TRUE, this will re-estimate P(sim)\$fitDeciduousCoverDiscount This may be unstable and is not recommended currently. If FALSE, will use the current default
dataYear	numeric	2001	NA	NA	Used to override the default 'sourceURL' of KNN datasets (species cover, stand biomass and stand age), which point to 2001 data, to fetch KNN data for another year. Currently, the only other possible year is 2011.
ecoregionLayerField	character	NULL	NA	NA	the name of the field used to distinguish ecoregions, if supplying a polygon. Defaults to NULL and tries to use 'ECODISTRIC' where available (for legacy reasons), or the row numbers of sim\$ecoregionLayer. If this field is not numeric, it will be coerced to numeric.
exportModels	character	none	NA	NA	Controls whether models used to estimate maximum B/ANPP (biomassModel) and species establishment (coverModel) probabilities are exported for posterior analyses or not. This may be important when models fail to converge or hit singularity (but can still be used to make predictions) and the user wants to investigate them further. Can be set to 'none' (no models are exported), 'all' (both are exported), 'biomassModel' or 'coverModel'. BEWARE: because this is intended for posterior model inspection, the models will be exported with data, which may mean very large simList(s)!
forestedLCCClasses	numeric	1, 2, 3, ...,	0	NA	The classes in the rstLCC layer that are 'treed' and will therefore be run in Biomass_core. Defaults to forested classes in LCC2010 map.

paramName	paramClass	default	min	max	paramDesc
imputeBadAgeModel	lme4::lm.NA	NA			Model and formula used for imputing ages that are either missing or do not match well with biomass or cover. Specifically, if biomass or cover is 0, but age is not, or if age is missing (NA), then age will be imputed. Note that this is independent from replacing ages inside fire perimeters (see <code>P(sim)\$overrideAgeInFires</code>)
LCCClassesToReplaceNN	NA	NA			This will replace these classes on the landscape with the closest forest class <code>P(sim)\$forestedLCCClasses</code> . If the user is using the LCC 2005 land-cover data product for <code>rstLCC</code> , then they may wish to include 36 (cities – if running a historic range of variation project), and 34:35 (burns) Since this is about estimating parameters for growth, it doesn't make any sense to have unique estimates for transient classes in most cases. If no classes are to be replaced, pass ' <code>LCCClassesToReplaceNN</code> ' = <code>numeric(0)</code> when supplying parameters.
minCoverThresholdERIC5	0	100			Pixels with total cover that is equal to or below this number will be omitted from the dataset

paramName	paramClass	default	min	max	paramDesc
minRelativeBFun	function	LandR::mNA	NA		A quoted function that makes the table of min. relative B determining a stand shade level for each ecoregionGroup. Using the internal object pixelCohortData is advisable to access/use the list of ecoregionGroups per pixel. The function must output a data.frame with 6 columns, named ecoregionGroup and 'X1' to 'X5', with one line per ecoregionGroup code, and the min. relative biomass for each stand shade level X1-5. The default function uses values from LANDIS-II available at: https://github.com/dcyr/LANDIS-II_IA_generalUseFiles/ blob/master/LandisInputs/BSW/biomass-succession-main- inputs_BSW_Baseline.txt%7E .
omitNonTreedPixels	logical	TRUE	FALSE	TRUE	Should this module use only treed pixels, as identified by P(sim)\$forestedLCCClasses?
overrideAgeInFires	logical	TRUE	NA	NA	should stand age values inside fire perimeters be replaced with number of years since last fire?
overrideBiomaskInputs	logical	TRUE	NA	NA	should B values be re-estimated using Biomass_core for pixels within the fire perimeters for which age was replaced with time since last fire? Ignored if P(sim)\$overrideAgeInFires = FALSE. See firePerimeters input object and P(sim)\$overrideAgeInFires for further detail.
pixelGroupAgeGranularity	numeric	NA	NA		When assigning pixelGroup membership, this defines the resolution of ages that will be considered 'the same pixelGroup', e.g., if it is 10, then 6 and 14 will be the same

paramName	paramClass	default	min	max	paramDesc
pixelGroupBiomassClass	integer	100	NA	NA	When assigning pixelGroup membership, this defines the resolution of biomass that will be considered ‘the same pixelGroup’, e.g., if it is 100, then 5160 and 5240 will be the same
rmImputedPix	logical	FALSE	NA	NA	Should sim\$imputedPixID be removed from the simulation?
speciesUpdateFunction	list	LandR::s.NA	NA	NA	Unnamed list of (one or more) quoted functions that updates species table to customize values. By default, LandR::speciesTableUpdate is used to change longevity and shade tolerance values, using values appropriate to Boreal Shield West (BSW), Boreal Plains (BP) and Montane Cordillera (MC) ecoprovinces (see ?LandR::speciesTableUpdate for details). Set to NULL if default trait values from speciesTable are to be kept instead. The user can supply other or additional functions to change trait values (see LandR::updateSpeciesTable)
sppEquivCol	character	Boreal	NA	NA	The column in sim\$speciesEquivalency data.table to use as a naming convention.
speciesTableAreas	character	BSW, BP, MC	NA	NA	One or more of the Ecoprovince short forms that are in the speciesTable file, e.g., BSW, MC etc. Default is good for Alberta and other places in the western Canadian boreal forests.
subsetDataAgeModel	numeric	50	NA	NA	the number of samples to use when subsampling the age data model and when fitting coverPctToBiomassPctModel; Can be TRUE/FALSE/NULL or numeric; if TRUE, uses 50, the default. If FALSE/NULL no subsetting is done.
successionTimeStep	numeric	10	NA	NA	defines the simulation time step, default is 10 years

paramName	paramClass	default	min	max	paramDesc
useCloudCache	logical	TRUE	NA	NA	Some of the statistical models take long (at least 30 minutes, likely longer). If this is TRUE, then it will try to get previous cached runs from googledrive.
vegLeadingProportion	numeric	0.8	0	1	a number that defines whether a species is leading for a given pixel
.plotInitialTime	numeric	0	NA	NA	This is here for backwards compatibility. Please use .plots
.plots	character	NA	NA	NA	This describes the type of 'plotting' to do. See ?Plots for possible types. To omit, set to NA
.plotInterval	numeric	NA	NA	NA	This describes the simulation time interval between plot events
.saveInitialTime	numeric	NA	NA	NA	This describes the simulation time at which the first save event should occur
.saveInterval	numeric	NA	NA	NA	This describes the simulation time interval between save events
.seed	list	NA	NA	NA	Named list of seeds to use for each event (names). E.g., list('init' = 123) will set.seed(123) at the start of the init event and unset it at the end. Defaults to NULL, meaning that no seeds will be set
.sslVerify	integer	64	NA	NA	Passed to <code>httr::config(ssl_verifypeer = P(sim)\$sslVerify)</code> when downloading KNN (NFI) datasets. Set to 0L if necessary to bypass checking the SSL certificate (this may be necessary when NFI's website SSL certificate is not correctly configured).
.studyAreaName	character	NA	NA	NA	Human-readable name for the study area used. If NA, a hash of studyArea will be used.

paramName	paramClass	default	min	max	paramDesc
.useCache	character	inputOb.NA	NA		Internal. Can be names of events or the whole module name; these will be cached by SpaDES

5.2.7 List of outputs

The module produces the following outputs (Table 5.5), 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*.
- 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, max-ANPP, SEP).
- minRelativeB – minimum relative biomass thresholds that determine a shade level in each pixel. X0-5 represent site shade classes from no-shade (0) to maximum shade (5).
- sufficientLight – probability of germination for species shade tolerance (in species) and shade level (defined byminRelativeB)

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.

TABLE 5.5: List of *Biomass_borealDataPrep* output objects and their description.

objectName	objectClass	desc
biomassMap	RasterLayer	total biomass raster layer in study area, filtered for pixels covered by cohortData. Units in g/m2
cohortData	data.table	initial community table, containing corrected biomass (g/m2), age and species cover data, as well as ecolocation and pixelGroup information. This table defines the initial community composition and structure used by Biomass_core
ecoregion	data.table	ecoregionGroup look up table
ecoregionMap	RasterLayer	ecoregionGroup map that has mapcodes match ecoregion table and speciesEcoregion table
imputedPixelID	integer	A vector of pixel IDs - matching rasterMatch IDs - that suffered data imputation. Data imputation may be in age (to match last fire event post 1950s, or 0 cover), biomass (to match fire-related imputed ages, correct for missing values or for 0 age/cover), land cover (to convert non-forested classes into to nearest forested class)
pixelGroupMap	RasterLayer	initial community map that has mapcodes (pixelGroup IDs) match cohortData
pixelFate	data.table	A small table that keeps track of the pixel removals and cause. This may help diagnose issues related to understanding the creation of cohortData
minRelativeBiomass	data.frame	minimum relative biomass thresholds that determine a shade level in each pixel. X0-5 represent site shade classes from no-shade (0) to maximum shade (5).
modelCover	data.frame	If P(sim)\$exportModels is 'all', or 'cover', fitted cover model, as defined by P(sim)\$coverModel.
modelBiomass	data.frame	If P(sim)\$exportModels is 'all', or 'biomass', fitted biomass model, as defined by P(sim)\$biomassModel

objectName	objectClass	desc
rawBiomass	RasterLayer	total biomass raster layer in study area. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived total aboveground biomass map (in tonnes/ha) from 2001, unless 'dataYear' != 2001. See https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata
species	data.table	a table that of invariant species traits. Will have the same traits as the input speciesTable with values adjusted where necessary
speciesEcoregion	table	table of spatially-varying species traits (maxB, maxANPP, establishprob), defined by species and ecoregionGroup)
studyArea	SpatialPolygons	PolyFrom use as the study area corrected for any spatial properties' mismatches with respect to studyAreaLarge.
sufficientLight	data.frame	Probability of germination for species shade tolerance (in species) and shade level (defined by minRelativeB) combinations. Table values follow LANDIS-II test traits available at: https://raw.githubusercontent.com/LANDIS-II-Foundation/Extensions-Succession/master/biomass-succession-archive/trunk/tests/v6.0-2.0/biomass-succession_test.txt

5.2.8 Simulation flow and module events

Biomass_borealDataPrep initialises itself and prepares all inputs, provided it has internet access to retrieve the raw datasets, for parametrisation and use by *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);
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)

5.3 Usage example

This module can be run stand-alone, but it won't do much more than prepare inputs for `Biomass_core`. Hence, we provide a usage example of this module and a few others in this repository¹⁷ and in Barros et al. [3].

5.4 References

- [3] Ceres Barros et al. "Empowering ecological modellers with a PERFECT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation". In: *Methods in Ecology and Evolution* 14 (2023), pp. 173–188. DOI: [10.1111/2041-210X.14034](https://doi.org/10.1111/2041-210X.14034)¹⁸.
- [5] A Beaudoin et al. *Species composition, forest properties and land cover types across Canada's forests at 250m resolution for 2001 and 2011*. 2017. DOI: [10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990](https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990)¹⁹. URL: <http://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990>.
- [6] 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.
- [10] E. H. (Ted) Hogg, J. P. Brandt, and M. Michaelian. "Impacts of a regional drought on the productivity, dieback, and biomass of western Canadian aspen forests". In: *Canadian Journal of Forest Research* 38.6 (June 2008), pp. 1373–1384. ISSN: 0045-5067, 1208-6037. DOI: [10.1139/X08-001](https://doi.org/10.1139/X08-001)²⁰. URL: <http://www.nrcresearchpress.com/doi/10.1139/X08-001>.
- [12] Christian Messier, Sylvain Parent, and Yves Bergeron. "Effects of over-story and understory vegetation on the understory light environment in mixed boreal forests". In: *Journal of Vegetation Science* 9.4 (Aug. 1998),

¹⁷https://github.com/CeresBarros/LandRBiomass_publication

¹⁸<https://doi.org/10.1111/2041-210X.14034>

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

²⁰<https://doi.org/10.1139/X08-001>

- pp. 511–520. ISSN: 11009233. DOI: 10 . 2307 / 3237266²¹. URL: <http://doi.wiley.com/10.2307/3237266>.
- [13] NRCAN. *National Ecological Framework for Canada - Terrestrial Ecozones*. 2013.
- [14] 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>.
- [20] Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.
- [21] 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>.
- [24] 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 . 18²⁴. URL: <http://www.nature.com/articles/sdata201618>.

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>

²⁴<https://doi.org/10.1038/sdata.2016.18>



6

LandR Biomass_core Module

Biomass core v. 1.3.10¹

Get help Report issues²

6.0.0.1 Authors:

Yong Luo yluo1@lakeheadu.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], Ian Eddy ian.eddy@nrcan-rncan.gc.ca⁷ [ctb], Jean Marchal jean.d.marchal@gmail.com⁸ [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.

6.1 Module Overview

6.1.1 Quick links

- General functioning
- List of input objects
- List of parameters

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

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

³<mailto:yluo1@lakeheadu.ca>

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

⁵<mailto:ceres.barros@ubc.ca>

⁶<mailto:achubaty@for-cast.ca>

⁷<mailto:ian.eddy@nrcan-rncan.gc.ca>

⁸<mailto:jean.d.marchal@gmail.com>

- List of outputs
- Simulation flow and module events

6.1.2 Summary

LandR *Biomass_core* (hereafter *Biomass_core*) is the core forest succession simulation module of the LandR ecosystem of SpaDES modules [see 8]. It simulates tree cohort ageing, growth, mortality and competition for light resources, as well as seed dispersal (Fig. 6.1), 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, 19], 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 [19] for a detailed reading of the mechanisms implemented in the model.

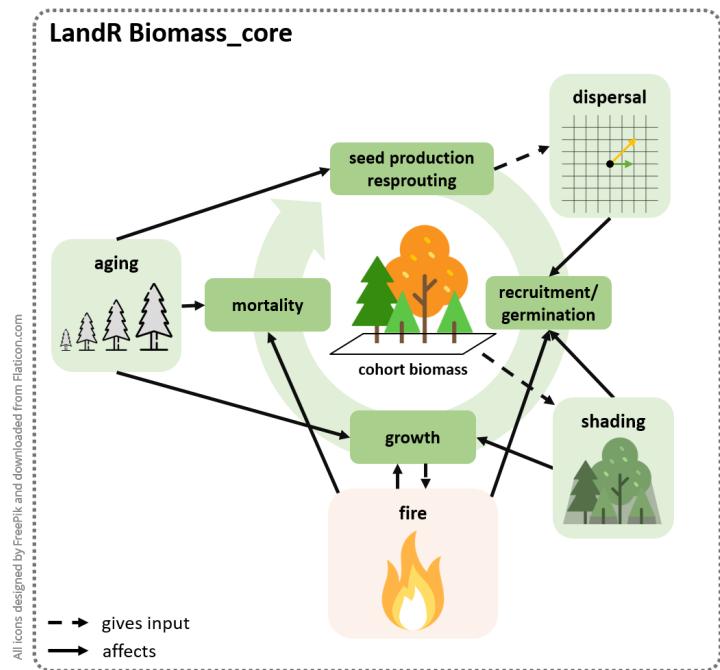


FIGURE 6.1: *Biomass_core* simulates tree cohort growth, mortality, recruitment and dispersal dynamics, as a function of cohort ageing and competition for light (shading) and space, as well as disturbances like fire (simulated using other modules).

6.1.3 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*. For those not yet in the LandR Manual⁹ see the individual module's documentation (.Rmd file) available in its repository.

See here¹⁰ for all available modules and select *Biomass_core* from the drop-down menu to see linkages.

Data and calibration modules:

- *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;
- *Biomass_borealDataPrep*¹²: 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*¹³: calibrates four-species level traits using permanent sample plot data (i.e., repeated tree biomass measurements) across Western Canada.

Disturbance-related modules:

- *Biomass_regeneration*¹⁴: simulates cohort biomass responses to stand-replacing fires (as in LBSE), including cohort mortality and regeneration through resprouting and/or serotiny;
- *Biomass_regenerationPM*¹⁵: like *Biomass_regeneration*, but allowing partial mortality. Based on the LANDIS-II Dynamic Fuels & Fire System extension [22];
- *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 cli-

⁹<https://landr-manual.predictiveecology.org/>

¹⁰https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem

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

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

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

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

¹⁵https://github.com/PredictiveEcology/Biomass_regenerationPM

mate, 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*¹⁶: wildfire ignition and cover-sensitive wildfire spread model based on a fire return interval input. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*;
- *scfm*¹⁷: 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*¹⁸: 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.

6.2 Module manual

6.2.1 General functioning

Biomass_core is a forest landscape model based on the LANDIS-II Biomass Succession Extension v.3.2.1 model [LBSE, 19]. 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. 6.1). 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 (maxi-

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

¹⁷<https://github.com/PredictiveEcology/scfm>

¹⁸https://github.com/PredictiveEcology/Biomass_validationKNN

mum 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 (`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 [19]. 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’ [19]. 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 [19], Scheller and Domingo [18] and Scheller and Domingo [17] for further details with respect to the above mentioned mechanisms implemented in `Biomass_core`. In a later sec-

tion of this manual, we highlight existing differences between *Biomass_core* and LBSE, together with comparisons between the two modules.

6.2.2 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 LBSE¹⁹, *Biomass_core* initialised 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.

6.2.2.1 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

¹⁹in 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) [19].

(`speciesLayers` raster layers) data (see Table 6.7) 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 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).

6.2.2.2 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 6.7), 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 [7](#)], 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 6.1 shows an example of a table of invariant species traits. Note that `Biomass_core` (alone) requires all the columns Table 6.1 in to be present, with the exception of `firetolerance`, `postfireregen`, `resproutprob`, `re-`

`sproutage_min` and `resproutage_max`, which are used by the post-fire regeneration modules (*Biomass_regeneration* and *Biomass_regenerationPM*).

Please see Scheller and Domingo [18, p.18] and Scheller and Miranda [19, p.16] for further detail.

TABLE 6.1: 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. (continued below)

speciesCode	longevity	sexualmature	shadetolerance	firetolerance
<code>Abie_sp</code>	200	20	2.3	1
<code>Pice_eng</code>	460	30	2.1	2
<code>Pice_gla</code>	400	30	1.6	2
<code>Pinu_sp</code>	150	15	1	2
<code>Popu_sp</code>	140	20	1	1
<code>Pseu_men</code>	525	25	2	3

TABLE 6.2: Table continues below

postfireregen	resproutprob	resproutage_min	resproutage_max
none	0	0	0
none	0	0	0
none	0	0	0
serotiny	0	0	0
resprout	0.5	10	70
none	0	0	0

seeddistance_eff	seeddistance_max	mortalityshape	growthcurve
25	100	15	0
30	250	15	1
100	303	15	1
30	100	15	0
200	5000	25	0
100	500	15	1

6.2.2.3 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 `mANPPproportion`; also for Western Canadian forests). See Table 6.4 for an example.

TABLE 6.4: 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	<code>Abie_sp</code>	1	8567	285	1
1_03	<code>Pice_eng</code>	0.983	10156	305	1
1_03	<code>Popu_sp</code>	0.737	8794	293	1
1_03	<code>Pseu_men</code>	1	17534	132	1
1_09	<code>Abie_sp</code>	0.112	1499	50	1
1_09	<code>Pice_gla</code>	0.302	3143	102	1
1_09	<code>Pinu_sp</code>	0.714	2569	86	1
1_09	<code>Popu_sp</code>	0.607	3292	110	1
1_09	<code>Pseu_men</code>	0.997	6020	45	1
1_03	<code>Abie_sp</code>	0.989	8943	225	2
1_03	<code>Pice_eng</code>	0.985	9000	315	2
1_03	<code>Popu_sp</code>	0.6	8600	273	2
1_03	<code>Pseu_men</code>	1	13534	142	2
1_09	<code>Abie_sp</code>	0.293	2099	45	2
1_09	<code>Pice_gla</code>	0.745	3643	90	2
1_09	<code>Pinu_sp</code>	0.5	2569	80	2

ecoregionGroup	speciesCode	establishprob	maxB	maxANPP	year
1_09	Popu_sp	0.67	3262	111	2
1_09	Pseu_men	1	6300	43	2

6.2.2.4 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 [19].

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

Site shade varies from X0 (no shade) to X5 (maximum shade). By default, `Biomass_core` uses the same minimum realtive biomass threshold values across all ecolocations, adjusted from a publicly available LANDIS-II table²⁰ to better reflect Western Canada boreal forest dynamics (see Table 6.5). `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.

TABLE 6.5: Example of a minimum relative biomass table (the `minRelativeB` table object in the module), with two ecolocations (ecoregionGroups) sharing the same values

ecoregionGroup	0.15	0.25	0.5	0.75	0.85
1_03	0.15	0.25	0.5	0.75	0.85
1_09	0.15	0.25	0.5	0.75	0.85

6.2.2.5 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 6.1) and the site shade [defined by the amount of biomass in the pixel – see `mini-`

²⁰https://github.com/dcyr/LANDIS-II_IA_generalUseFiles

mum relative biomass parameter and 19, p.14]. By default, both *Biomass_core* and *Biomass_borealDataPrep* use a publicly available LANDIS-II table (called *sufficientLight* in the module; Table 6.6).

TABLE 6.6: 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	X0	X1	X2	X3	X4	X5
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

6.2.2.6 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.

6.2.3 List of input objects

All of *Biomass_core*'s input objects have (theoretical) defaults that are produced automatically by the module²¹. 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.

²¹usually, 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*

Of the inputs listed in Table 6.7, 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)²². 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.
- `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`

²²<https://open.canada.ca/data/en/dataset/3ef8e8a9-8d05-4fea-a8bf-7f5023d2b6e1>

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`.
- `sppNameVector` – (OPTIONAL) a character vector of species to be simulated. If provided, `Biomass_core` uses this vector to (attempt to) obtain `speciesLayers` for the listed species. If not provided, the user (or another module) can pass a filtered `sppEquiv` table (i.e., containing only the species that are to be simulated). If neither is provided, then `Biomass_core` attempts to use any species for which it finds available species % cover data in the study area.

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 [19].
- *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).

TABLE 6.7: List of *Biomass_core* input objects and their description.

objectName	objectClass	desc	sourceURL
biomassMap	RasterLayer	total biomass raster layer in study area (in g/m^2), filtered for pixels covered by cohortData. Only used if <code>P(sim)\$initialBiomassSource == 'biomassMap'</code> , which is currently deactivated.	
cceArgs	list	a list of quoted objects used by the growthAndMortalityDriver calculateClimateEffect function	NA
cohortData	data.table	data.table with cohort-level information on age and biomass, by pixelGroup and ecolocation (i.e., ecoregionGroup). If supplied, it must have the following columns: pixelGroup (integer), ecoregionGroup (factor), speciesCode (factor), B (integer in g/m^2), age (integer in years)	NA

objectName	objectClass	desc	sourceURL
ecoregion	data.table	ecoregion look up table	https://raw.githubusercontent.com/LANDIS-II-Foundation/Experiments-Succession/master/biomass-succession-archive/trunk/texts/v6.0-2.0/ecoregions.txt
ecoregionMap	RasterLayer	ecoregion map that has mapcodes match ecoregion table and speciesEcoregion table. Defaults to a dummy map matching rasterToMatch with two regions	NA
lastReg	numeric	an internal counter keeping track of when the last regeneration event occurred	NA
minRelativeB	data.frame	table defining the relative biomass cut points to classify stand shadiness.	NA

objectName	objectClass	desc	sourceURL
pixelGroupMap	RasterLayer	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., via the Biomass_borealDataPrep module).	NA
rasterToMatch	RasterLayer	a raster of the studyArea in the same resolution and projection as biomassMap	NA
species	data.table	a table of invariant species traits with the following trait columns: 'species', 'Area', 'longevity', 'sexualmature', 'shadetolerance', 'firetolerance', 'seeddistance_eff', 'seeddistance_max', 'resproutprob', 'mortalityshape', 'growthcurve', 'resproutage_min', 'resproutage_max', 'postfireregen', 'wooddecayrate', 'leaflongevity', 'leafLignin', 'hardsoft'. The last seven traits are not used in Biomass_core, and may be omitted. However, this may result in downstream issues with other modules. Default is from Dominic Cyr and Yan Boulanger's project	https://raw.githubusercontent.com/dcyr/LANDIS-II_I_A_generalUseFiles/master/speciesTraits.csv
speciesEcoRegion	data.table	table of spatially-varying species traits (maxB, maxANPP, establishprob), defined by species and ecoregionGroup) Defaults to a dummy table based on dummy data os biomass, age, ecoregion and land cover class	NA

objectName	objectClass	desc	sourceURL
speciesLayers	RasterStack	percent cover raster layers of tree species in Canada. Defaults to the Canadian Forestry Service, National Forest Inventory, kNN-derived species cover maps from 2001 using a cover threshold of 10 - see https://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990 for metadata	http://ftp.maps.cada.ca/pub/nrcan_rnca/Forests/canada/forests-attribute/s_attributes/forests-canada/2001-atributes_2001/
sppColorVect	character	A named vector of colors to use for plotting. The names must be in <code>sim\$sppEquiv</code> [[<code>sim\$sppEquivCol</code>]], and should also contain a color for 'Mixed'.	NA
sppEquiv	data.table	table of species equivalencies. See <code>LandR::sppEquivalencies_CA</code> .	NA

objectName	objectClass	desc	sourceURL	
sppNameVec	vector	character	an optional vector of species names to be pulled from sppEquiv. Species names must match P(sim)\$sppEquivCol column in sppEquiv. If not provided, then species will be taken from the entire P(sim)\$sppEquivCol column in sppEquiv. See LandR::sppEquivalencies_CA.	NA
studyArea	SpatialPolygonsDataFrame	SpatialPolygonsDataFrame	Polygon to use as the study area. Must be provided by the user	NA
studyAreaReporting	SpatialPolygonsDataFrame	SpatialPolygonsDataFrame	multipolygon (typically smaller/unbuffered than studyArea) to use for plotting/reporting. Defaults to studyArea.	NA

objectName	objectClass	desc	sourceURL
sufficientLight	data.frame	table defining how the species with different shade tolerance respond to stands shade. Default is based on LANDIS-II Biomass Succession v6.2 parameters	https://raw.githubusercontent.com/LANDIS-II-Foundation/Experiments-Succession/master/biomass-succession-archive/trunk/test/v6.0-2.0/biomass-succession_test.txt
treedFirePixelTableSinceLastDisp	data.table	3 columns: pixelIndex, pixelGroup, and burnTime. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding pixelGroup and time it occurred	NA

6.2.4 List of parameters

In addition to the above inputs objects, *Biomass_core* uses several parameters²³ 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 package²⁴ (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 full set of parameters is in Table 6.8. 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;

²³in Spades lingo parameters are “small” objects, such as an integer or boolean, that can be controlled via the `parameters` argument in `simInit`.

²⁴<https://github.com/ianmseddy/LandR.CS>

- vegLeadingProportion – relative biomass threshold to consider a species “leading” (i.e., dominant);

TABLE 6.8: List of *Biomass_core* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
calcSummaryBGM	character	end	NA	NA	A character vector describing when to calculate the summary of biomass, growth and mortality. Currently any combination of 5 options is possible: 'start' - as before vegetation succession events, i.e. before dispersal, 'postDisp' - after dispersal, 'postRegen' - after post-disturbance regeneration (currently the same as 'start'), 'postGM' - after growth and mortality, 'postAging' - after aging, 'end' - at the end of vegetation succession events, before plotting and saving. The 'end' option is always active, being also the default option. If NULL, then will skip all summaryBGM related events
calibrate	logical	FALSE	NA	NA	Do calibration? Defaults to FALSE

paramName	paramClass	default	min	max	paramDesc
cohortDefinitionCols	character	pixelGro....	NA	NA	cohortData columns that determine what constitutes a cohort This parameter should only be modified if additional modules are adding columns to cohortData
cutpoint	numeric	1e+10	NA	NA	A numeric scalar indicating how large each chunk of an internal data.table is, when processing by chunks
initialB	numeric	10	1	NA	initial biomass values of new age-1 cohorts. If NA or NULL, initial biomass will be calculated as in LANDIS-II Biomass Suc. Extension (see Scheller and Miranda, 2015 or ?LandR:::initiateNewCohorts)

paramName	paramClass	default	min	max	paramDesc
gmcsGrowthLimits	numeric	66.66666....	NA	NA	if using LandR.CS for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model
gmcsMortLimits	numeric	66.66666....	NA	NA	if using LandR.CS for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model

paramName	paramClass	default	min	max	paramDesc
gmcsMinAge	numeric	21	0	NA	if using LandR.CS for climate-sensitive growth and mortality, the minimum age for which to predict climate-sensitive growth and mortality. Young stands (< 30) are poorly represented by the PSP data used to parameterize the model.
growthAndMortalityDrivers	character	LandR	NA	NA	package name where the following functions can be found: calculateClimateEffect, assignClimateEffect (see LandR.CS for climate sensitivity equivalent functions, or leave default if this is not desired)
growthInitialTime	numeric	0	NA	NA	Initial time for the growth event to occur

paramName	paramClass	default	min	max	paramDesc
initialBiomassSource	character	cohortData	NA	NA	Currently, there are three options: 'spinUp', 'cohortData', 'biomassMap'. If 'spinUp', it will derive biomass by running spinup derived from Landis-II. If 'cohortData', it will be taken from the cohortData object, i.e., it is already correct, by cohort. If 'biomassMap', it will be taken from sim\$biomassMap, divided across species using sim\$speciesLayers percent cover values 'spinUp' uses sim\$standAgeMap as the driver, so biomass is an output . That means it will be unlikely to match any input information about biomass, unless this is set to 'biomassMap', and a sim\$biomassMap is supplied. Only the 'cohortData' option is currently active.
keepClimateCols	logical	FALSE	NA	NA	include growth and mortality predictions in cohortData?
minCohortBiomass	numeric	0	NA	NA	cohorts with biomass below this threshold (in g/m^2) are removed. Not a LANDIS-II BSE parameter.

paramName	paramClass	default	min	max	paramDesc
mixedType	numeric	2	NA	NA	How to define mixed stands: 1 for any species admixture; 2 for deciduous > conifer. See ?LandR::vegTypeMapGenerator.
plotOverstory	logical	FALSE	NA	NA	swap max age plot with overstory biomass
seedingAlgorithm	character	wardDisp....	NA	NA	choose which seeding algorithm will be used among 'noSeeding' (no horizontal, nor vertical seeding - not in LANDIS-II BSE), 'noDispersal' (no horizontal seeding), 'universalDispersal' (seeds disperse to any pixel), and 'wardDispersal' (default; seeds disperse according to distance and dispersal traits). See Scheller & Miranda (2015) - Biomass Succession extension, v3.2.1 User Guide
spinupMortalityfraction	numeric	0.001	NA	NA	defines the mortality loss fraction in spin up-stage simulation. Only used if P(sim)\$initialBiomassSource == 'biomassMap', which is currently deactivated.

paramName	paramClass	default	min	max	paramDesc
sppEquivCol	character	Boreal	NA	NA	The column in <code>sim\$sppEquiv</code> data.table to use as a naming convention
successionTimestep	numeric	10	NA	NA	defines the simulation time step, default is 10 years. Note that growth and mortality always happen on a yearly basis. Cohorts younger than this age will not be included in competitive interactions
vegLeadingProportion	numeric	0.8	0	1	a number that defines whether a species is leading for a given pixel
.maxMemory	numeric	5	NA	NA	maximum amount of memory (in GB) to use for dispersal calculations.
.plotInitialTime	numeric	0	NA	NA	Vector of length = 1, describing the simulation time at which the first plot event should occur. To plotting off completely use <code>P(sim)\$plots</code> .
.plotInterval	numeric	NA	NA	NA	defines the plotting time step. If NA, the default, <code>.plotInterval</code> is set to <code>successionTimestep</code> .

paramName	paramClass	default	min	max	paramDesc
.plots	character	object	NA	NA	Passed to types in Plots (see ?Plots). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at end(sim). If NA, plotting is turned off completely (this includes plot saving).
.plotMaps	logical	TRUE	NA	NA	Controls whether maps should be plotted or not. Set to FALSE if P(sim)\$plots == NA
.saveInitialTime	numeric	NA	NA	NA	Vector of length = 1, describing the simulation time at which the first save event should occur. Set to NA if no saving is desired. If not NA, then saving will occur at P(sim)\$saveInitialTime with a frequency equal to P(sim)\$saveInterval
.saveInterval	numeric	NA	NA	NA	defines the saving time step. If NA, the default, .saveInterval is set to P(sim)\$successionTimestep.

paramName	paramClass	default	min	max	paramDesc
.sslVerify	integer	64	NA	NA	Passed to <code>http::config(ssl_verifypeer = P(sim)\$sslVerify)</code> when downloading KNN (NFI) datasets. Set to 0L if necessary to bypass checking the SSL certificate (this may be necessary when NFI's website SSL certificate is not correctly configured).
.studyAreaName	character	NA	NA	NA	Human-readable name for the study area used. If NA, a hash of <code>studyArea</code> will be used.
.useCache	character	.inputOb....	NA	NA	Internal. Can be names of events or the whole module name; these will be cached by <code>SpaDES</code>
.useParallel	ANY	2	NA	NA	Used only in seed dispersal. If numeric, it will be passed to <code>data.table::setDTthreads</code> and should be <= 2; If TRUE, it will be passed to <code>parallel::makeCluster</code> ; and if a cluster object, it will be passed to <code>parallel::parClusterApplyB</code> .

6.2.5 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 6.9) can be saved via the `outputs` argument in `simInit`²⁵.

TABLE 6.9: 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 the reporting study area.
ANPPMap	RasterLayer	ANPP map at each succession time step (in g /m ²)
biomassMap	RasterLayer	total biomass raster layer in study area (in g/m ²), filtered for pixels covered by <code>cohortData</code> . Only used if <code>P(sim)\$initialBiomassSource == 'biomassMap'</code> , which is currently deactivated.

²⁵see `?SpaDES.core::outputs`

objectName	objectClass	desc
cohortData	data.table	data.table with cohort-level information on age, biomass, aboveground primary productivity (year's biomass gain) and mortality (year's biomass loss), by pixelGroup and ecolocation (i.e., ecoregionGroup). Contains at least the following columns: pixelGroup (integer), ecoregionGroup (factor), speciesCode (factor), B (integer in g/m^2), age (integer in years), mortality (integer in g/m^2), aNPPAct (integer in g/m^2). May have other columns depending on additional simulated processes (i.e., climate sensitivity; see, e.g., P(sim)\$keepClimateCols).
ecoregion	data.table	ecoregion look up table
ecoregionMap	RasterLayer	map with mapcodes match ecoregion table and speciesEcoregion table. Defaults to a dummy map matching rasterToMatch with two regions.
inactivePixelIndex	logical	internal use. Keeps track of which pixels are inactive.
inactivePixelIndexReportinginteger		internal use. Keeps track of which pixels are inactive in the reporting study area.
lastFireYear	numeric	Year of the most recent fire.
lastReg	numeric	an internal counter keeping track of when the last regeneration event occurred.
minRelativeB	data.frame	define the relative biomass cut points to classify stand shade.

objectName	objectClass	desc
mortalityMap	RasterLayer	map of biomass lost (in g/m^2) at each succession time step.
pixelGroupMap	RasterLayer	updated community map at each succession time step.
regenerationOutput	data.table	If <code>P(sim)\$calibrate == TRUE</code> , an summary of seed dispersal and germination success (i.e., number of pixels where seeds successfully germinated) per species and year.
reproductionMap	RasterLayer	Regeneration map (biomass gains in g/m^2) at each succession time step
simulatedBiomassMap	RasterLayer	Biomass map at each succession time step (in g/m^2)
simulationOutput	data.table	contains simulation results by ecoregionGroup (main output)
simulationTreeOutput	data.table	Summary of several characteristics about the stands, derived from cohortData
species	data.table	a table that has species traits such as longevity, shade tolerance, etc. Currently obtained from LANDIS-II Biomass Succession v.6.0-2.0 inputs
speciesEcoregion	data.table	define the maxANPP, maxB and SEP change with both ecoregion and simulation time.
speciesLayers	RasterStack	species percent cover raster layers, based on input speciesLayers object. Not changed by this module.
spinupOutput	data.table	Spin-up output. Currently deactivated.

objectName	objectClass	desc
sppColorVect	character	A named vector of colors to use for plotting. The names must be in sim\$sppEquiv[[sim\$sppEquivCol]], and should also contain a color for 'Mixed'.
summaryBySpecies	data.table	The total species biomass (in g/m^2 as in cohortData), average age and aNPP (in g/m^2 as in cohortData), across the landscape (used for plotting and reporting).
summaryBySpecies1	data.table	Number of pixels of each leading vegetation type (used for plotting and reporting).
summaryLandscape	data.table	The averages of total biomass (in tonnes/ha , not g/m^2 like in cohortData), age and aNPP (also in tonnes/ha) across the landscape (used for plotting and reporting).
treedFirePixelTableSinceLastDisptable	data.table	3 columns: pixelIndex, pixelGroup, and burnTime. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding pixelGroup and time it occurred
vegTypeMap	RasterLayer	Map of leading species in each pixel, colored according to sim\$sppColorVect. Species mixtures calculated according to P(sim)\$vegLeadingProportion and P(sim)\$mixedType.

6.2.6 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 data and calibration modules or by *Biomass_core* itself (during `simInit` and the `init` event²⁶);
2. Disturbances (OPTIONAL) – simulated by a disturbance module (e.g., *LandMine*);
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 [17] 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 [21] for further detail.
6. Cohort age binning (every `successionTimestep`; `cohortAgeReclassification` event):

²⁶`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.

- 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 [19] 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 (`plotAvg`s and `plotSummaryBySpecies` events);
 9. Save outputs (`save` event).
- ... (repeat 2-9) ...

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

6.2.7.1 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, 19]. 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 6.10 and Fig. 6.2). The calculation of competition, growth and mortality for all cohorts at the same time also circumvented this issue.

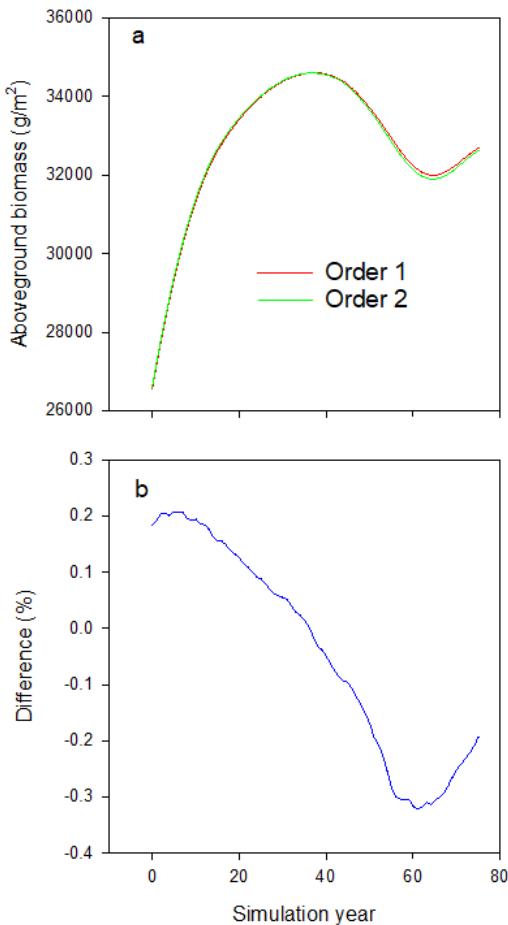


FIGURE 6.2: Differences in total landscape aboveground biomass when using two different input species orders for the same community. These simulations demonstrate how the sequential calculation of the competition index, combined with a lack of explicit species ordering affect the overall landscape aboveground biomass in time when using different input species orders (see Table

reftab:tableLBSEtestI). 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

$$\frac{\text{Biomass}_{\text{order2}} - \text{Biomass}_{\text{order1}}}{\text{Biomass}_{\text{order2}}} * 100$$

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)*" [19, 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 6.12 and Fig. 6.3). 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.

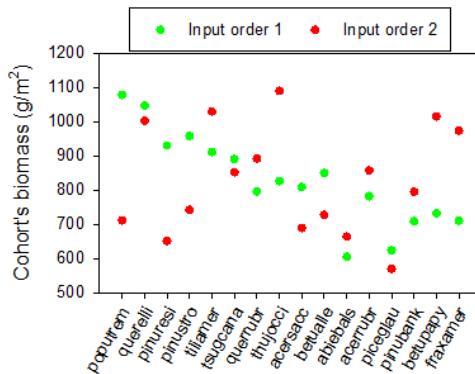


FIGURE 6.3: Differences in the biomass assigned to new cohorts, summed for each species across pixels, when using two different input species orders for the same community and when the succession time step is 1. These simulations demonstrate how the different summation of total cohort biomass for a succession 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).

6.2.7.2 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.

6.2.7.2.1 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*²⁷ and *Biomass_regenerationPM*²⁸). 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

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

²⁸https://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.

6.2.7.2.2 Hashing

Our first strategy to improve simulation efficiency in `Biomass_core` was to use a hashing mechanism [25]. 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. 6.4). 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.

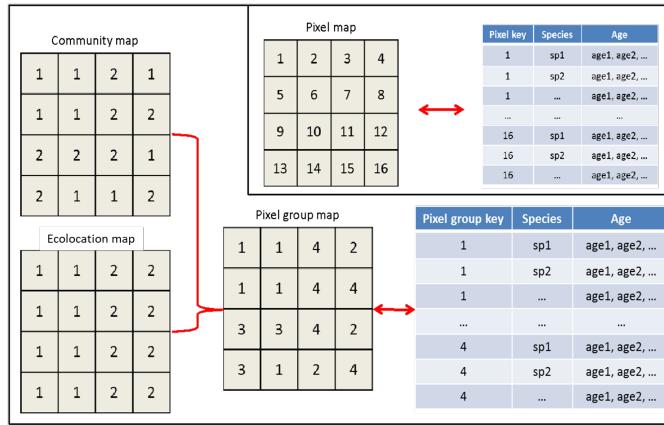


FIGURE 6.4: Hashing design for *Biomass_core*. In the re-coded *Biomass_core*, the pixel group map was hashed based on the unique combination of species composition ('community map') and ecolocation map, and associated with a lookup table. The insert in the top-right corner was the original design that linked the map to the lookup table by pixel key.

6.2.7.2.3 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 [11].

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.

6.2.7.2.4 Testing

Finally, we implemented code testing to facilitate bug detection by comparing the outputs of functions (etc.) to expected outputs [23]. 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.

6.2.7.3 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 6.14–6.16). The remaining input parameters were taken from a LANDIS-II training course (Tables 6.17–6.21), and contained species attributes information of 16 common tree species in boreal forests and 2 ecolocations. We ran simulations for 1000 years, with a succession time step of 10 and three 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. 6.5), 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. 6.5 of this appendix).

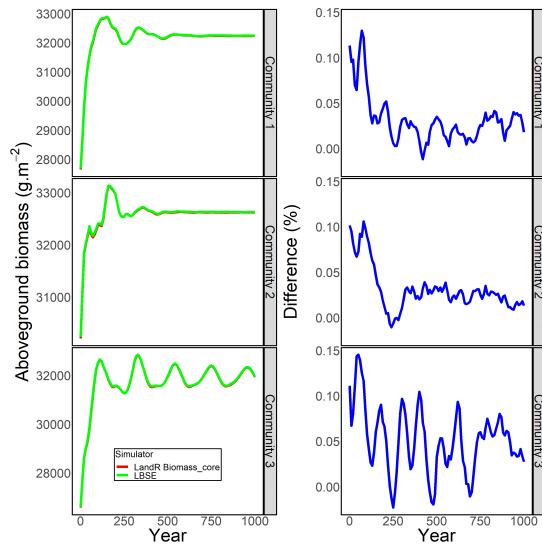


FIGURE 6.5: Visual comparison of simulation outputs for three randomly generated initial communities (left panels) and difference between those outputs (right panels). The % difference between LBSE and *Biomass_core* were calculated as $\frac{\text{Biomass}_{LBSE} - \text{Biomass}_{\text{Biomass}_\text{core}}}{\text{Biomass}_{LBSE}} * 100$

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. 6.6a). 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. 6.6a). *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. 6.6b).

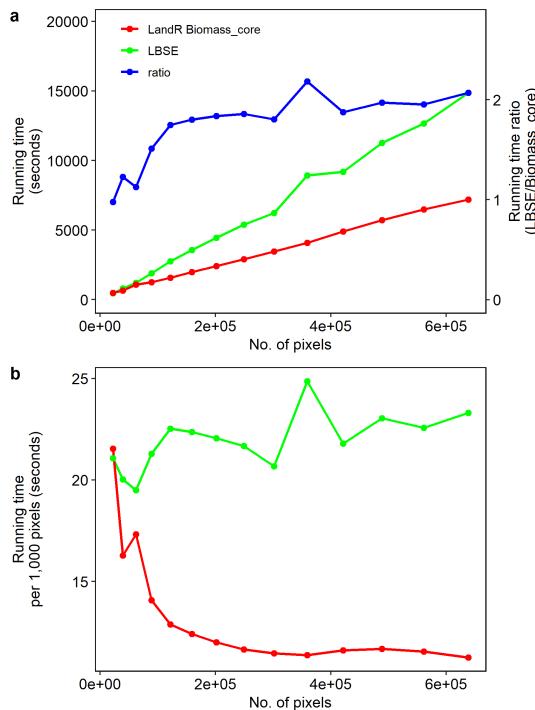


FIGURE 6.6: Simulation efficiencies of LBSE and *Biomass_core* with increasing map size, in terms of a) mean running time across repetitions (left y-axis) and the ratio LBSE to *Biomass_core* running times (right y-axis and blue line), and b) running time scalability as the mean running time per 1000 pixels.

6.3 Usage example

6.3.1 Set up R libraries

```
options(repos = c(CRAN = "https://cloud.r-project.org"))
tempDir <- tempdir()
```

```
pkgPath <- file.path(tempDir, "packages", version$platform,  
paste0(version$major,  
      ".", strsplit(version$minor, "[.]")[[1]][1]))  
dir.create(pkgPath, recursive = TRUE)  
.libPaths(pkgPath, include.site = FALSE)  
  
if (!require(Require, lib.loc = pkgPath)) {  
  remotes::install_github(  
    "PredictiveEcology/Require@5c44205bf407f613f53546be652a438ef1248147"  
,  
    upgrade = FALSE, force = TRUE)  
  library(Require, lib.loc = pkgPath)  
}  
  
setLinuxBinaryRepo()
```

6.3.2 Get the module and module dependencies

We can use the SpaDES.project::getModule function to download the module to the module folder specified above. Alternatively, see SpaDES-modules repository²⁹ to see how to download this and other SpaDES modules, or fork/clone from its GitHub repository³⁰ directly.

After downloading the module, it is important to make sure all module R package dependencies are installed in their correct version. SpaDES.project::packagesInModules makes a list of necessary packages for all modules in the paths\$modulePath, and Require installs them.

```
Require(  
  "PredictiveEcology/SpaDES.project@6d7de6ee12fc967c7c60de44f1aa3b04e6eeb5db"  
,  
  require = FALSE, upgrade = FALSE, standAlone = TRUE)  
  
paths <- list(inputPath = normPath(file.path(tempDir,  
  "inputs")),
```

²⁹<https://github.com/PredictiveEcology/SpaDES-modules>

³⁰https://github.com/PredictiveEcology/Biomass_core/

```

cachePath = normPath(file.path(tempDir, "cache")),
modulePath = normPath(file.path(tempDir,
    "modules")), outputPath = normPath(file.path(tempDir,
    "outputs"))

SpaDES.project:::getModule(modulePath = paths$modulePath,
c("PredictiveEcology/Biomass_core@master"),
    overwrite = TRUE

## make sure all necessary packages are installed:
outs <- SpaDES.project:::packagesInModules(modulePath =
paths$modulePath)
Require(c(uname(unlist(outs)), "SpaDES"), require = FALSE,
standAlone = TRUE

## load necessary packages
Require(c("SpaDES", "LandR", "reproducible", "pemisc"), upgrade
= FALSE,
    install = FALSE)

```

6.3.3 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 (`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 what to produce and choose to output cohortData tables every year – note that these are not pixel-based, so to “spatialise” results *a posteriori* the `pixelBroupMap` 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 <- sppEquivalencies_CA[get(speciesNameConvention) %in%
  speciesToUse]
# Assign a colour convention for graphics for each species
sppColorVect <- sppColors(sppEquiv, speciesNameConvention,
newVals = "Mixed",
  palette = "Set1")

## Usage example
modules <- as.list("Biomass_core")
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))
```

6.3.4 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.

Below, we pass some useful reproducible options that control caching ("reproducible.useCache") and where inputs should be downloaded to ("reproducible.destinationPath").

```
opts <- options(reproducible.useCache = TRUE,
reproducible.destinationPath = paths$inputPath,
spades.useRequire = FALSE)
graphics.off()
mySim <- simInitAndSpades(times = times, params = parameters,
modules = modules, objects = objects, paths = paths, outputs
= outputs,
debug = TRUE)
```

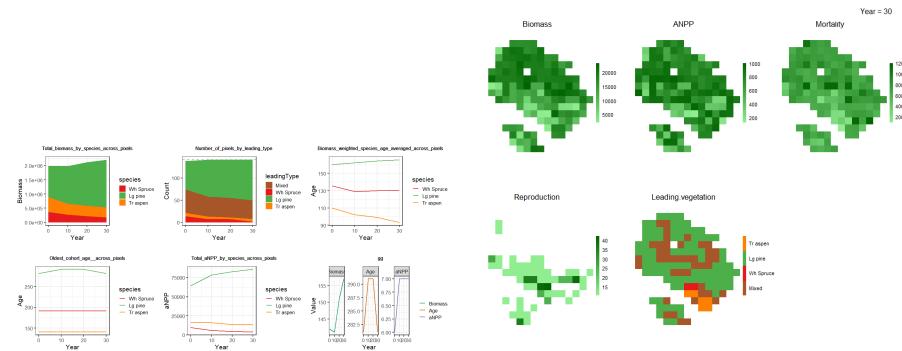


FIGURE 6.7: *Biomass_core* automatically generates simulation visuals of species dynamics across the landscape in terms of total biomass, number of presences and age and productivity (above), as well as yearly plots of total biomass, productivity, mortality, reproduction and leading species in each pixel (below).

6.4 Appendix

6.4.1 Tables

TABLE 6.10: 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. (continued below)

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

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

TABLE 6.12: 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. (continued below)

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

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

TABLE 6.14: 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 6.15: Randomly generated community combination no. 2 used in the recruitment comparison runs.

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
0	acerrubr	22	26	30	40	47	145	146
0	betualle	23	41	43	120	209	227	270
0	fraxamer	25	90	119	173	185	282	NA
0	pinuresi	48	53	70	121	157	NA	NA
0	pinustro	5	82	126	298	352	NA	NA
0	querrubr	2	30	34	74	77	162	245
1	acerrubr	2	39	43	84	116	127	143
1	pinubank	34	57	75	NA	NA	NA	NA
1	querelli	108	202	218	243	NA	NA	NA
1	querrubr	5	117	131	186	189	246	NA
1	tiliamer	10	19	46	80	133	148	231
1	tsugcana	31	48	190	246	330	NA	NA
2	pinubank	11	37	38	47	67	93	NA
2	querrubr	11	48	57	177	180	228	236
2	tiliamer	28	42	78	79	223	250	NA
2	tsugcana	140	202	372	381	451	NA	NA
3	acersacc	48	107	262	265	NA	NA	NA
3	betupapy	4	12	45	65	83	96	NA
3	poputrem	13	20	37	75	90	NA	NA
3	querelli	72	90	104	115	116	265	278
3	tiliamer	20	21	56	98	237	NA	NA
3	tsugcana	86	224	425	429	NA	NA	NA
4	fraxamer	77	133	181	NA	NA	NA	NA
4	pinustro	13	37	67	220	287	293	375
4	querrubr	27	48	89	97	NA	NA	NA
4	thujocci	91	244	305	390	NA	NA	NA
5	abiebals	86	95	119	121	127	158	NA
5	betualle	83	113	136	161	216	231	NA
5	betupapy	10	38	64	NA	NA	NA	NA
5	piceglau	16	63	70	102	NA	NA	NA
6	acerrubr	8	34	112	NA	NA	NA	NA
6	betupapy	1	31	57	61	74	80	91
6	fraxamer	63	100	108	140	196	294	NA
6	pinubank	15	19	44	47	51	80	NA
6	thujocci	78	146	163	213	214	228	NA

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
6	tsugcana	47	108	387	389	449	NA	NA

TABLE 6.16: 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

Community	Species	Age 1	Age 2	Age 3	Age 4	Age 5	Age 6	Age 7
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 6.17: Invariant species traits table used in comparison runs. (continued below)

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

Seeddistance_max	Mortalityshape	Growthcurve
160	10	0.25
200	10	0.25
200	10	0.25
400	10	0.25
5000	10	0.25
140	10	0.25
200	10	0.25
100	10	0.25
275	10	0.25
210	10	0.25
5000	10	0.25
3000	10	0.25
3000	10	0.25
60	10	0.25
120	10	0.25
100	10	0.25

TABLE 6.19: 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 6.20: 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 6.21: Species ecolocation table used in comparison runs. SEP stands for species establishment probability, maxB for maximum biomass and maxANPP for maximum aboveground net primary productivity. Values were held constant throughout the simulation.

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

Ecolocation	Species	SEP	maxANPP	maxB
1	querrubr	0.66	1017	30510
1	thujocci	0.76	1090	32700
1	tiliamer	0.54	1078	32340
1	tsugcana	0.22	1096	32880

6.5 References

- [7] 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.
- [8] 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>.
- [11] 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>, %20<https://github.com/PredictiveEcology/reproducible>.
- [17] Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – Conceptual Description*. Apr. 20, 2012.
- [18] Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – User Guide*. July 19, 2011.
- [19] Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.
- [21] Robert M. Scheller and David J. Mladenoff. “A forest growth and biomass module for a landscape simulation model, LANDIS: design, validation, and application”. In: *Ecological Modelling* 180.1 (Dec. 2004), pp. 211–229. DOI: [10.1016/j.ecolmodel.2004.01.022](https://doi.org/10.1016/j.ecolmodel.2004.01.022)³¹. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304380004003837>.
- [22] Brian R. Sturtevant et al. *LANDIS-II Dynamic Fire System Extension v3.0 – User Guide*. 2018.

³¹<https://doi.org/10.1016/j.ecolmodel.2004.01.022>

- [23] Hadley Wickham. “testthat: Get Started with Testing”. In: *The R Journal* 3.1(2011), p. 5. DOI: 10.32614/RJ-2011-002³². URL: <https://journal.r-project.org/archive/2011/RJ-2011-002/index.html>.
- [25] 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.2011.04.032³³. URL: <https://www.sciencedirect.com/science/article/pii/S0304380011002651>.

³²<https://doi.org/10.32614/RJ-2011-002>

³³<https://doi.org/10.1016/j.ecolmodel.2011.04.032>

7

LandR Biomass_regeneration Module

This documentation is work in progress. Please report any discrepancies or omissions at https://github.com/PredictiveEcology/Biomass_regeneration/issues.

7.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca¹ [aut, cre], Yong Luo yluo1@lakeheadu.ca² [aut], Ceres Barros cbarros@mail.ubc.ca³ [aut], Alex M. Chubaty achubaty@for-cast.ca⁴ [ctb]

7.1 Module Overview

7.1.1 Module summary

Biomass_regeneration is a SpaDES module that simulates post-disturbance regeneration mechanisms for Biomass_core. As such, this module is mostly based on the post-disturbance regeneration mechanisms present in LANDIS-II Biomass Succession v3.2.1 extension (see LANDIS-II Biomass Succession v3.2 User Guide⁵ and Scheller and Mladenoff (2004)⁶. At the moment, the Biomass_regeneration module only simulates post-fire disturbance effects on forest species, by simulating post-fire mortality and activating serotiny or resprouting mechanisms for each species, depending on their traits (i.e. ability to resprout and/or germinate from seeds, serotiny,

¹<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

²<mailto:yluo1@lakeheadu.ca>

³<mailto:cbarros@mail.ubc.ca>

⁴<mailto:achubaty@for-cast.ca>

⁵[https://github.com/LANDIS-II-Foundation/Extension-Biomass-Succession/
blob/master/docs/LANDIS-II%20Biomass%20Succession%20v3.2%20User%20Guide.docx](https://github.com/LANDIS-II-Foundation/Extension-Biomass-Succession/blob/master/docs/LANDIS-II%20Biomass%20Succession%20v3.2%20User%20Guide.docx)

⁶[https://pdfs.semanticscholar.org/4d38/d0be6b292eccd444af399775d37a757d1967.
pdf](https://pdfs.semanticscholar.org/4d38/d0be6b292eccd444af399775d37a757d1967.pdf)

following fire). Post-fire mortality behaves in a stand-replacing fashion, i.e. should a pixel be within a fire perimeter (determined by a fire raster) all cohorts see their biomasses set to 0.

As for post-fire regeneration, the module first evaluates whether any species present prior to fire are serotinous. If so, these species will germinate depending on light conditions and their shade tolerance, and depending on their (seed) establishment probability (i.e. germination success) in that pixel. The module then evaluates if any species present before fire are capable of resprouting. If so the model growth these species depending, again, on light conditions and their shade tolerance, and on their resprouting probability (i.e. resprouting success). For any given species in any given pixel, only serotiny or resprouting can occur. Hence, species that are capable of both will only resprout if serotiny was not activated.

In LANDIS-II, resprouting could never occur in a given pixel if serotiny was activated for one or more species. According to the manual:

If serotiny (only possible immediately following a fire) is triggered for one or more species, then neither resprouting nor seeding will occur. Serotiny is given precedence over resprouting as it typically has a higher threshold for success than resprouting. This slightly favors serotinous species when mixed with species able to resprout following a fire.

(LANDIS-II Biomass Succession v3.2 User Guide⁷)

This is no longer the case in Biomass_regeneration, where both serotinity and resprouting can occur in the same pixel, although not for the same species. We feel that this is more realistic ecologically, as resprouters will typically regenerate faster after a fire, often shading serotinous species and creating interesting successional feedbacks (e.g. light-loving serotinous species having to “wait” for canopy gaps to germinate).

⁷[https://github.com/LANDIS-II-Foundation/Extension-Biomass-Succession/
blob/master/docs/LANDIS-II%20Biomass%20Succession%20v3.2%20User%20Guide.docx](https://github.com/LANDIS-II-Foundation/Extension-Biomass-Succession/blob/master/docs/LANDIS-II%20Biomass%20Succession%20v3.2%20User%20Guide.docx)

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

objectName	objectClass	desc
cohortData	data.table	age cohort Biomass table hooked to pixel group map by pixelGroupIndex at succession time step
lastFireYear	numeric	Year of the most recent fire year
pixelGroupMap	RasterLayer	Pixel group map that changes map by each succession time step
serotinyResproutSuccessPixels	numeric	Pixels that were successfully regenerated via serotiny or resprouting. This is a subset of treedBurnLoc
postFireRegenSummary	data.table	summary table of species post-fire regeneration
severityMap	RasterLayer	map of tree severity, as in the area and post-fire mortality biomass loss
severityData	data.table	A data table of pixels severity, as in the area and post-fire mortality biomass loss. May also have severity class used to calculate mortality
treedFirePixelTableSinceLastDisp	data.table	3 columns: pixelIndex, pixelGroup, and burnTime. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding pixelGroup and time it occurred

TABLE 7.2: List of *Biomass_regeneration* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
calibrate	logical	FALSE	NA	NA	Do calibration? Defaults to FALSE
cohortDefinitionCohort	character	seedero	NA	NA	cohort type used in cohorts to determine unique cohorts
fireInitialTime	numeric	1	NA	NA	The event time that the first fire disturbance event occurs
fireTimeStep	numeric	1	NA	NA	the number of time units between successive fire events in a fire module
initialBiomass	numeric	10	1	NA	Initial biomass values of new age-1 cohorts. If NA or NULL, initial biomass will be calculated as in LANDIS-II Biomass Suc. Extension (see Scheller and Miranda, 2015 or ?LandR_.initiateNewCohorts)
successionTimeStep	numeric	10	1	NA	succession time step
plotEvent	character	screen	NA	NA	Used by Plot function, which can be optionally used here
plotInitialTime	numeric	0	NA	NA	This describes the simulation time at which the first plot event should occur
plotTimeStep	numeric	10	1	NA	This describes the simulation time interval between plot events
sequentialTimeStep	numeric	NA	NA	NA	This describes the simulation time at which the first event should occur
saveInterval	numeric	NA	NA	NA	This describes the simulation time interval between save events
useCache	character	inputObj	NA	NA	Should this entire module be run with caching activated? This is generally intended for data-type modules, where stochasticity and time are not relevant

7.1.2 General flow of *Biomass_regeneration* processes - fire disturbances only

1. Removal of biomass in disturbed, i.e. burnt, pixels
2. Activation of serotiny for serotinous species present before the fire
3. Activation of resprouting for resprouter species present before the fire and for which serotiny was not activated
4. Establishment/growth of species for which serotiny or resprouting were activated

7.1.3 Module inputs and parameters

Table 7.1 shows the full list of module inputs.

Summary of user-visible parameters (Table 7.2):

7.1.4 Module outputs

Description of the module outputs (Table 7.3).

TABLE 7.3: List of *Biomass_regeneration* outputs and their description.

objectName	objectClass	desc
cohortData	data.table	age cohort Biomass table hooked to pixel group map by pixelGroupIndex at succession time step
lastFireYear	numeric	Year of the most recent fire year
pixelGroupMap	RasterLayer	Pixel group map that changes map by each succession time step
serotinyResproutSuccessPixels	numeric	Pixels that were successfully regenerated via serotiny or resprouting. This is a subset of treedBurnLoc
postFireRegenSummary	data.table	summary table of species post-fire regeneration
severityMap	RasterLayer	map of tree severity, as in the area and post-fire mortality biomass loss
severityData	data.table	A data table of pixels severity, as in the area and post-fire mortality biomass loss. May also have severity class used to calculate mortality
treedFirePixelTableSinceLastDisp	data.table	3 columns: pixelIndex, pixelGroup, and burnTime. Each row represents a forested pixel that was burned up to and including this year, since last dispersal event, with its corresponding pixelGroup and time it occurred

7.1.5 Links to other modules

Primarily used with the LandR Biomass suite of modules, namely Biomass_core⁸.

7.2 Getting help

- https://gitter.im/PredictiveEcology/LandR_Biomass ## References

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

Wildfire submodel

Lorem ipsum ... (TODO)



8

LandMine Module

This documentation is work in progress. Please report any discrepancies or omissions at <https://github.com/PredictiveEcology/LandMine/issues>.

8.0.0.1 Authors:

Eliot J B McIntire [`eliot.mcintire@nrcan-rncan.gc.ca`](mailto:eliot.mcintire@nrcan-rncan.gc.ca)¹ [aut, cre], Alex M. Chubaty [`achubaty@for-cast.ca`](mailto:achubaty@for-cast.ca)² [ctb]

8.1 Module Overview

8.1.1 Module summary

Landmine is a model created for simulating the natural range of variation for landscapes in the boreal forest [1, 2]. It has been widely used by the public and the private sector for various purposes. This SpaDES module is a rewrite of the fire component in native R.

8.1.2 Model Differences

The current version has not yet been fully tested and compared with the original version, but there are currently several known differences:

1. Fire sizes are taken from a Truncated Pareto distribution, resulting in numerous very small fires, and few large fires;
2. Parameters have been fitted to the landscapes that are under study in the LandWeb project.

¹[`mailto:eliot.mcintire@nrcan-rncan.gc.ca`](mailto:eliot.mcintire@nrcan-rncan.gc.ca)

²[`mailto:achubaty@for-cast.ca`](mailto:achubaty@for-cast.ca)

TABLE 8.1: LandMine species codes.

Species	Group	Code
Jack pine	Pine (PINU)	Pinu_ban
Lodgepole pine	Pine (PINU)	Pinu_con
Unspecified pine species	Pine (PINU)	Pinu_sp
Paper birch	Deciduous (DECI)	Betu_pap
Balsam poplar	Deciduous (DECI)	Popu_bal
Trembling aspen	Deciduous (DECI)	Popu_tre
Larch/Tamarack	Deciduous (DECI)	Lari_lar
Black spruce	Black spruce (PICE_MAR)	Pice_mar
White spruce	White spruce (PICE_GLA)	Pice_gla
Fir species	Fir (ABIE)	Abie_sp

8.1.3 Known Species

Landmine requires the following codes as inputs (the genus and species codes below), which converts and groups species as follows. Each of the species groups has its own Rate of Spread (ROS) for fire spreading:

TABLE 8.2: List of *LandMine* input objects and their description.

TABLE 8.3: List of *LandMine* parameters and their description.

8.1.4 Module inputs and parameters

Table 8.2 shows the full list of module inputs.

Summary of user-visible parameters (Table 8.3)

8.1.5 Module outputs

Description of the module outputs (Table 8.4).

TABLE 8.4: List of *LandMine* outputs and their description.

objectName	objectClass	desc
burnTime	numeric	The actual event time of the burn event. This is simply a reassignment from <code>PlanBURNInitTime</code> .
firesize	list	A list of data tables, one per burn event, each with two columns, <code>size</code> and <code>maxsize</code> . These indicate the actual sizes and expected sizes burned, respectively. These can be put into a single data table with <code>rbListDataFromFireSizes</code> , <code>ideal -> years</code> .
firestartInterval	RasterLayer	<code>A raster</code> showing the fire return interval. This is created from <code>rbCurrentBurn</code> .
firestartIntervalByPolyID	numeric	The fire return interval by polygon ID.
firestartIntervalByYear	numeric	A vector of the fire return intervals, ordered by the numerical representation of polygon ID
firestartIntervalIndex	data.table	The index of the time of fire return between successive fire events in a fire return histogram.
firestartIndex	list	A list of data tables, one per fire return histogram.
kbest	numeric	A numeric scalar value that is the optimal value of 'K' in the Truncated Pareto distribution (<code>truncapareto</code>)
numFireYear	numeric	The average number of fires per year, by fire return interval based on <code>rbCurrentBurn</code> .
rbCurrentBurn	RasterLayer	<code>A raster</code> showing the state of the land cover, where each pixel is either 1 or 0 indicating burned or not burned.
rbCurrentBurnCumulative	RasterLayer	<code>A raster</code> showing the cumulative number of pixels burned per year.
spEqLum	data.table	Same as input, but with new column, <code>LandMint</code> .

8.2 Usage

To run this Landmine module alone (*i.e.*, for fitting), the following should work (iff raster inputs for `rstStudyRegion` and `rstFlammable` are available), assuming all R packages are available.

NB: Paths will have be changed for a different user.

8.2.1 Package dependencies

To determine which packages are used by `LandMine`, use:

```
SpaDES.core:::packages(modules = "LandMine", paths = "...")[[1]]  
  
## [1] "SpaDES.core"  
## [2] "assertthat"  
## [3] "data.table"  
## [4] "ggplot2"  
## [5] "grDevices"  
## [6] "gridExtra"  
## [7] "magrittr"  
## [8] "PredictiveEcology/LandR@development (>= 1.1.0.9003)"  
## [9] "PredictiveEcology/pemisc@development"  
## [10] "PredictiveEcology/SpaDES.tools@development"  
## [11] "raster"  
## [12] "RColorBrewer"  
## [13] "VGAM"
```

8.2.2 Module usage

```
studyArea <- SpaDES.tools:::randomStudyArea(seed = 1234, size =  
1e+10)  
rasterToMatch <- raster(studyArea, res = 250)
```

```
times <- list(start = 0, end = 13)
```

```
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) parameters
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) <-
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) list(#
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) LandMine
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) =
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) list(flushCachedRandomFRI
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) =
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) TRUE)
parameters <- list(# LandMine = list(flushCachedRandomFRI =
TRUE) ) )

modules <- list("LandMine")

objects <- list(studyArea = studyArea, rasterToMatch =
rasterToMatch)

paths <- list(cachePath = cacheDir, modulePath = moduleDir,
inputPath = inputDir,
outputPath = outputDir)
```

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

dev()
mySimOut <- spades(mySim, .plotInitialTime = times$start, debug
= TRUE)
```

8.3 Testing the burn algorithm

```
Require(c("data.table", "DEoptim", "parallel", "SDMTools"))

s <- simInit(times = times, params = parameters, modules =
modules,
             objects = objects, paths = paths)

ros <- raster(extent(0, 250000, 0, 250000), res = 250, vals = 0)
ros <- ros == 0
fireSize <- 1e+05

source(file.path(moduleDir, "LandMine", "R", "burn.R"))

burnFun <- function(ros, centreCell, fireSize, spawnNewActive,
                     sizeCutoffs, burn1, spreadProb) {
  burned <- burn1(landscape = ros, startCells = centreCell,
                  fireSizes = fireSize, spreadProb = spreadProb,
                  spreadProbRel = ros,
                  spawnNewActive = spawnNewActive, sizeCutoffs =
sizeCutoffs)
  burnedMap <- raster(ros)
  burnedMap[] <- NA

  burnedMap[burned$pixels] <- burned$initialPixels
  LM <- SDMTools::PatchStat(burnedMap, cellsize =
res(burnedMap)[1])
  list(burnedMap = burnedMap, LM = LM)
}

makeParallel <- function(wantParallel, numClus, cl, funs,
addlPkgs) {
  if (wantParallel) {
    library(parallel)
    if (is.null(cl)) {
```

```
        cl <- parallel::makeCluster(numClus)
    }
    funs <- c(funs)
    parallel::clusterExport(cl, funs)
    # env <- environment()
    parallel::clusterExport(cl, "addlPkgs", envir =
parent.frame())
    parallel::clusterEvalQ(cl, {
        lapply(addlPkgs, library, character.only = TRUE)
    })
    return(cl)
}
}

wrap <- function(cl = NULL, reps, par, burn1, burnFun, objs,
addlPkgs) {
    funNames <- c(deparse(substitute(burn1)),
deparse(substitute(burnFun)))
    cl <- makeParallel(TRUE, numClus = if (length(cl))
        length(cl) else detectCores() - 1, cl, c(funNames,
objs), addlPkgs)
    objs <- append(mget(objs, envir = parent.frame()),
list(burn1 = burn1))
    parallel::clusterExport(cl, "objs", envir = parent.frame())
    burnMapList <- parallel::clusterApplyLB(cl, reps,
function(r) {
        do.call("burnFun", objs)
    })
    return(list(cl = cl, out = burnMapList))
}

fitSN <- function(sna, ros, centreCell, fireSizes = 10^(2:5),
desiredPerimeterArea = 0.004, burnFun, burn1) {
    sizeCutoffs <- 10^sna[5:6]
    spreadProb <- sna[7]
    sna <- c(10^(sna[1]), 10^(sna[2]), 10^(sna[3]), 10^(sna[4]))
    bfs1 <- lapply(fireSizes, function(fireSize) {
        burnFun(ros, centreCell, fireSize, sna, sizeCutoffs,
```

```

        burn1 = burn1, spreadProb = spreadProb)
    })
res <- lapply(seq(bfs1), function(bfCount) {
    abs(log(bfs1[[bfCount]]$LM[1, "perim.area.ratio"])) -
        log(desiredPerimeterArea)) + 100 *
    (sum(bfs1[[bfCount]]$burnedMap[],
         na.rm = TRUE) < fireSizes[bfCount])
    # it needs to get to above 90,000 HA for it to
    # count
})
a <- sum(unlist(res)) # * log10(fireSizes)) # weigh larger
ones more
attr(a, "bfs1") <- bfs1
a
}

fitSN2 <- function(par, ros, centreCell, fireSizes = 10^(2:5),
                     desiredPerimeterArea = 0.003, spreadProb = 0.9) {
    sizeCutoffs <- 10^c(par[4], par[5])
    bfs1 <- lapply(fireSizes, function(fireSize) {
        sna <- min(-0.15, par[1] + par[2] * log10(fireSize))
        sna <- 10^c(sna * par[3], sna * 2 * par[3], sna * 3 *
            par[3], sna * 4 * par[3])
        # sna <- -1
        burnFun(ros, centreCell, fireSize, sna, sizeCutoffs,
                burn1, spreadProb)
    })
    res <- lapply(seq(bfs1), function(bfCount) {
        abs(log(bfs1[[bfCount]]$LM[1, "perim.area.ratio"])) -
            log(desiredPerimeterArea)) + 100 *
        (sum(bfs1[[bfCount]]$burnedMap[],
             na.rm = TRUE) < fireSizes[bfCount])
        # it needs to get to above 90,000 HA for it to
        # count
    })
    a <- sum(unlist(res))
    attr(a, "bfs1") <- bfs1
}

```

```
a  
}
```

8.4 Optimizing parameters

The following code chunk tries to find values of `spawnNewActive` that creates “reasonable” fire shapes at all sizes.

```
wantParallel <- TRUE
maxRetriesPerID <- 4 ## 4 retries (5 attempts total)
spreadProb <- 0.9
spawnNewActive <- c(0.46, 0.2, 0.26, 0.11)
sizeCutoffs <- c(8000, 20000)
# spawnNewActive <- c(0.1, 0.04, 0.025, 0.01) sizeCutoffs
# <- c(8e3, 2e4)

NineCorners <- cellFromRowCol(ros, row = nrow(ros)/4 * rep(1:3,
  3), col = ncol(ros)/4 * rep(1:3, each = 3))

centreCell <- cellFromRowCol(ros, row = nrow(ros)/2, col =
  ncol(ros)/2)

## Set variables
objs <- c("ros", "centreCell", "fireSize", "spawnNewActive",
  "sizeCutoffs", "spreadProb")
funs <- c("burnFun", "burn1")
addlPkgs <- c("data.table", "raster", "SDMTools",
  "SpaDES.tools")
```

```
### SET UP CLUSTER FOR PARALLEL
numCores <- min(detectCores()/2, 120L)
if (Sys.info()["sysname"] == "Windows") {
  cl <- parallel::makeCluster(numCores)
} else if (grepl("W-VIC-A1053|pinus[.]for-cast[.]ca",
  Sys.info()["nodename"])) {
  require("future")

  ## TODO: customize this to make a cluster on several
  ## machines via ssh
```

```
clNames <- c(rep("localhost", 0), rep("picea.for-cast.ca",
  25), rep("pseudotsuga.for-cast.ca", 80))

## check the cluster: 1. at least 10 populations per
## parameter; 2. ensure it's a multiple of
## the number of params (7); 3. R can't create more
## than ~125 socket connections.
stopifnot(length(clNames) >= 70, length(clNames)%%7 == 0,
  length(clNames) <= 120)
cl <- makeClusterPSOCK(clNames, homogeneous = FALSE, verbose
= TRUE)

  clusterSetRNGStream(cl, sample(1e+08, 1))
} else {
  cl <- parallel::makeCluster(numCores, type = "FORK")
  clusterSetRNGStream(cl, sample(1e+08, 1))
}

if (!inherits(cl[[1]], "forknode")) {
  parallel::clusterExport(cl, funs)
  parallel::clusterExport(cl, objs)
  # env <- environment()
  parallel::clusterExport(cl, "addlPkgs", envir =
parent.frame())
  parallel::clusterEvalQ(cl, {
    lapply(addlPkgs, library, character.only = TRUE)
  })
}
```

```
opt_sn <- DEoptim(fitSN, lower = c(-2, -3, -3, -3, 1, 3.5, 0.75),
  upper = c(-0.1, -0.5, -0.5, -1, 3.5, 5, 1), control =
DEoptim.control(VTR = 0.001,
  NP = as.integer(length(clNames)), itermax = 200, cluster
= cl,
  strategy = 6), ros = ros, centreCell = centreCell,
  fireSizes = c(10,
  100, 1000, 10000, 1e+05), desiredPerimeterArea = 0.003,
```

```

burnFun = burnFun, burn1 = burn1)

opt_sn$optim$bestmem ## best param values
# dput(unname(opt_sn$optim$bestmem))

## assign with suffix to facilitate multiple DEoptim runs
timestamp <- format(Sys.time(), "%Y%M%d%H%M%S")
assign(paste0("opt_sn_", timestamp), opt_sn)
saveRDS(opt_sn, file.path(moduleDir, "LandMine", "data",
paste0(timestamp,
"_DEoptim_250m.rds")))

```

```

fs_sn <- c(10, 100, 1000, 10000, 1e+05)
fit_sn <- fitsN(sna = c(-1, -1, -1, -2, 2, 4, 0.9), ros = ros,
centreCell = centreCell, fireSizes = fs_sn,
desiredPerimeterArea = 0.003,
burnFun = burnFun, burn1 = burn1)
bfs1_sn <- purrr::transpose(attr(fit_sn, "bfs1"))
LM_sn <- do.call(rbind, bfs1_sn$LM)
plot(fs_sn, LM_sn[, "perim.area.ratio"]) ## NOTE: visual
inspection - not too round; not too sinuous

```

A second (alternative) version tries the optimization using fewer parameters, to test whether a simpler version gets better/different results. Although this version was not used for the final module, we preserve it here for posterity.

```

fs_optim2 <- c(0.2, 1:8) * 10000
opt_sn2 <- DEoptim(fitSN2, lower = c(1, -1, 1, 3, 4), upper =
c(3,
-0.3, 3, 4, 5), control = DEoptim.control(VTR = 0.001,
itermax = 40,
cluster = cl, strategy = 6), ros = ros, centreCell =
centreCell,
fireSizes = fs_optim2, desiredPerimeterArea = 0.003)

```

```
fs_sn2 <- round(runif(10, 10, 4000))
fit_sn2 <- fitSN2(par = c(2, -0.63333, 1, 3.2, 4.4), ros = ros,
  centreCell = centreCell, fireSizes = fs_sn2,
  desiredPerimeterArea = 0.003,
  spreadProb = 0.9)
bfs1_sn2 <- purrr::transpose(attr(fit_sn2, "bfs1"))
LM_sn2 <- do.call(rbind, bfs1_sn2$LM)
plot(fs_sn2, LM_sn2[, "perim.area.ratio"]) ## NOTE: visual
  inspection - not too round; not too sinuous
```

8.5 Manual inspection of optimization results

8.5.1 Original (2018) version

The original version was run using 100m pixels, despite the simulations being run using 250m pixels. This was corrected and rerun below.

```
## 10,000 hectares burns gave this
spawnNewActive[2:3] <- c(0.0235999945606232, 0.0263073265505955)

# 100,000 hectare burns gave this spawnNewActive <-
# 10^c(-1.264780, -1.970946, -1.288213, -2.202580)
spawnNewActive <- 10^c(-0.586503645288758, -1.08108837273903,
-2.14391896536108, -1.00221184641123)
sizeCutoffs <- 10^c(3.37711253212765, 4.52040993282571)

sns <- c(-1.733262, -0.933318, -2.562183, -2.493687, 3.064458,
4.812305)
spawnNewActive <- 10^sns[1:4]
sizeCutoffs <- 10^sns[5:6]
# spawnNewActive <- 10^c(-1.646419, -1.815395, -2.809013,
# -2.613337) sizeCutoffs <- 10^c(3.888317, 4.641961)

## 100,000 pixel fires -- the next worked, but I think we
## can get better sns <- structure(
## c(-1.652459,-0.962121,-0.964879,-2.304902, 3.522345,
## 4.173242), .Names = c('par1',
sns <- structure(c(-1.641197, -1.152821, -0.697335, -1.751917,
3.720378, 4.034059), .Names = c("par1", "par2", "par3",
"par4",
"par5", "par6"))
spawnNewActive <- 10^sns[1:4]
sizeCutoffs <- 10^(sns[5:6])
fireSize <- 30000

## 100
sns <- c(-0.77716149196811, -0.769325340166688,
-1.2772046867758,
```

```
-1.99332102853805, 3.14260408212431, 4.46155184064992)

## 1000
sns <- c(-0.775107, -1.03176, -0.599669, -1.958105, 3.048958,
        4.275831)

## seemed good for 100,000, pretty good for 1e3
sns <- c(-1.54885, -0.97052, -1.38305, -1.93759, 3.20379,
        4.13237)

## good for 100 000, 10 000 ha -- too sinuous for 1000 and
## 100 ha
sns <- c(-1.537203, -1.462981, -0.524957, -1.002567, 3.642046,
        4.501754)

## good for 100 000, 10 000 ha (except some fires @ 1e5
## don't make it to full size) -- too sinuous for smaller
sns <- c(-1.484338, -1.22044, -2.948275, -2.15594, 3.945281,
        4.904893)

sns <- c(-1.495247, -0.800494, -1.58235, -2.270646, 3.530671,
        4.663245)

## final optimization after 75 iterations, Good: 1e5, 1e4
sns <- c(-1.47809, -0.86224, -1.34532, -1.93568, 3.27149,
        4.20741)

## based on equal weights 10^(1:5)
sns <- c(-0.923528, -1.804549, -1.760455, -1.793594, 1.683355,
        4.466668)

## With spreadProb = 0.9 # Pretty GOOD!
sns <- c(-0.73152, -0.501823, -0.605968, -1.809726, 2.202732,
        4.69606, 0.9) ## used in module

## With spreadProb = 0.9 # Optimal
sns <- c(-0.978947, -0.540946, -0.790736, -1.583039, 2.532013,
        4.267547, 0.94673)
```

```
spawnNewActive <- 10^sns[1:4]
sizeCutoffs <- 10^(sns[5:6])
if (length(sns) == 7) spreadProb <- sns[7]

# from linear model version
par <- c(1.548899, -0.396904, 2.191424, 3.903082, 4.854002)
sizeCutoffs <- 10^c(par[4], par[5])
sna <- min(-0.15, par[1] + par[2] * log10(fireSize))
sna <- 10^c(sna * par[3], sna * 2 * par[3], sna * 3 * par[3],
             sna * 4 * par[3])
spawnNewActive <- sna

#####
clearPlot()
dev()

for (i in 1:5) {
  fireSize <- 10^i
  dim <- round(sqrt(fireSize) * 5 * 250)
  ros <- raster(extent(0, dim, 0, dim), res = 250, vals = 1)
  centreCell <- cellFromRowCol(ros, rownr = nrow(ros)/2, colnr
= ncol(ros)/2)
  reps <- paste0("rep", 1:4 + (log10(fireSize) - 1) * 4)
  burnedMapList <- wrap(cl = cl, reps = reps, par = TRUE,
burn1 = burn1,
  burnFun = burnFun, objs = objs, libs = addlPkgs)
  names(burnedMapList$out) <- reps
  burnedMapList <- purrr::transpose(burnedMapList$out)
  do.call(rbind, burnedMapList$LM)

  Plot(burnedMapList$burnedMap, cols = c("red"), new = FALSE,
    na.color = "white", legend = FALSE, visualSqueeze = 0.7,
    title = paste0("Fire size: ", fireSize))
}
```

```
reps <- paste0("rep", 1:1)
perims <- list()
perm <- list()
mod <- list()
dev()
clearPlot()
fireSizes <- 10^(4)

for (fs in fireSizes) {
  for (i in 1:1) {
    ros <- raster(extent(0, 2e+05, 0, 2e+05), res = 250,
           vals = 1)
    NineCorners <- cellFromRowCol(ros, rownr = nrow(ros)/4 *
      rep(1:3, 3), colnr = ncol(ros)/4 * rep(1:3, each = 3))
    centreCell <- NineCorners
    ran <- runif(4, -3, -1)
    spawnNewActive <- 10^ran
    # spawnNewActive <- 10^c(-0.1, -0.75, -1.2,
    # ran*2.5)
    fireSize = rep(fs, length(centreCell))
    sizeCutoffs <- 10^c(1, 3)
    burnedMapList <- wrap(cl = cl, reps = reps, par = TRUE,
                           burn1 = burn1, burnFun = burnFun, objs = objs, libs
                           = addlPkgs)
    names(burnedMapList$out) <- reps
    burnedMapList <- purrr::transpose(burnedMapList$out)
    do.call(rbind, burnedMapList$LM)
    Plot(burnedMapList$burnedMap, new = TRUE, zero.color =
      "white")
    perims[[i]] <- data.frame(perim =
      burnedMapList$LM$rep1$perim.area.ratio,
                                 spawnNewActive = mean(spawnNewActive), others =
                                 t(spawnNewActive))
  }
  fsChar <- as.character(fs)
  perm[[fsChar]] <- rbindlist(perims)
  perm[[fsChar]]$perim <- log10(perm[[fsChar]]$perim)
  perm[[fsChar]]$spawnNewActive <-
  perm[[fsChar]]$spawnNewActive
```

```

Plot(perm[[fsChar]]$perim, perm[[fsChar]]$spawnNewActive,
      new = TRUE, addTo = paste0("fs", fsChar))
mod[[fsChar]] <- lm(spawnNewActive ~ perim, data =
perm[[fsChar]])
}

(predict(mod[["10"]], data.frame(perim = log10(0.003))))
(predict(mod[["100"]], data.frame(perim = log10(0.003))))
log10(predict(mod[["1000"]], data.frame(perim = log10(0.003))))

```

8.5.2 Current (2022) version

The original version was run using 100m pixels, despite the simulations being run using 250m pixels. This version uses 250m pixels.

```

## final optimization after 170 iterations (fitSN)
sns <- unname(opt_sn$optim$bestmem)

spawnNewActive <- 10^sns[1:4]
sizeCutoffs <- 10^(sns[5:6])
spreadProb <- sns[7]

## from linear model version
par <- c(1.548899, -0.396904, 2.191424, 3.903082, 4.854002)
sizeCutoffs <- 10^c(par[4], par[5])
sna <- min(-0.15, par[1] + par[2] * log10(fireSize))
sna <- 10^c(sna * par[3], sna * 2 * par[3], sna * 3 * par[3],
           sna * 4 * par[3])
spawnNewActive <- sna

clearPlot()

for (i in 1:5) {
  fireSize <- 10^i
  dim <- round(sqrt(fireSize) * 5 * 250)
  ros <- raster(extent(0, dim, 0, dim), res = 250, vals = 1)
  centreCell <- cellFromRowCol(ros, row = nrow(ros)/2, col =
  ncol(ros)/2)

```

```
reps <- paste0("rep", 1:4 + (log10(fireSize) - 1) * 4)
burnedMapList <- wrap(cl = cl, reps = reps, par = TRUE,
burn1 = burn1,
    burnFun = burnFun, objs = objs, addlPkgs = addlPkgs)
names(burnedMapList$out) <- reps
burnedMapList <- purrr::transpose(burnedMapList$out)
do.call(rbind, burnedMapList$LM)

Plot(burnedMapList$burnedMap, cols = c("red"), new = FALSE,
    na.color = "white", legend = FALSE, visualSqueeze = 0.7,
    title = paste0("Fire size: ", fireSize))
}
```

8.5.3 Cleaning up

```
parallel::stopCluster(cl)
unlink(cacheDir, recursive = TRUE)
```

8.5.4 Code and data availability

Code available from <https://github.com/PredictiveEcology/LandMine>.

8.5.5 Links to other modules

Originally developed as part of the LandWeb³ project. ## References

- [1] David W Andison. “Managing for landscape patterns in the sub-boreal forests of British Columbia”. PhD Thesis. Vancouver, BC: University of British Columbia, 1996. URL: <https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0075275>.
- [2] David W. Andison. “Temporal patterns of age-class distributions on foothills landscapes in Alberta”. In: *Ecography* 21.5 (1998), pp. 543–550. ISSN: 09067590. DOI: [10.1111/j.1600-0587.1998.tb00446.x](https://doi.org/10.1111/j.1600-0587.1998.tb00446.x)⁴.

³<https://github.com/PredictiveEcology/LandWeb>

⁴<https://doi.org/10.1111/j.1600-0587.1998.tb00446.x>



Model outputs

- burnSummaries module ... (TODO)
- LandMine module ... (TODO)
- LandWeb_outputs module ... (TODO)
- LandWeb_summary module ... (TODO)
- timeSinceFire module ... (TODO)

Previously-run model outputs are available to download for authorized users, and are described below.

8.6 Currently Selected Spatial Area

- Currently selected reporting polygon (.shp)
-

8.7 Large Patches Data for study region

- Large Patches Data (.csv)
 - Large patches histograms (.png)
-

8.8 Leading Vegetation Cover Data for study region

- Leading Vegetation Cover Data (.csv)
- Leading Vegetation Cover histograms (.png)
- Leading Vegetation Cover boxplots (.png)

8.9 Simulation Rasters (cropped to study region)

- Flammability maps (.grd)
 - Time Since Fire maps (.tif)
 - Vegetation type maps (.grd, .tif)
-

8.10 Additional R Data Files (advanced users)

- Simulation data files (.RData, .rds)

The downloaded outputs are bundled into a zip file (*e.g.*, `LandWeb_v2.0.0_2019-09-23.zip`) with the following directory and file structure:

```
LandWeb_v2.0.0_2019-09-23/
|_ boxplots/
    |_ leading_*.csv
|_ histograms/
    |_ largePatches/
        |_ 100/
        |_ 500/
        |_ 1000/
        |_ 5000/
    |_ leading/
    |_ largePatches_*.csv
|_ polygons/
|_ rasters/
    |_ rstTimeSinceFire_*.tif
    |_ vegTypeMap_*.tif*
|_ INFO.md
|_ README.md
|_ simulationOutput_*.rds
```

9

burnSummaries Module

9.0.0.1 Authors:

Alex M Chubaty achubaty@for-cast.ca¹ [aut, cre] (<https://orcid.org/0000-0001-7146-8135>)

9.1 Module Overview

9.1.1 Module summary

Provide a brief summary of what the module does / how to use the module.

Module documentation should be written so that others can use your module. This is a template for module documentation, and should be changed to reflect your module.

9.1.2 Module inputs and parameters

Describe input data required by the module and how to obtain it (e.g., directly from online sources or supplied by other modules) If sourceURL is specified, downloadData("burnSummaries", "...") may be sufficient. Table 9.1 shows the full list of module inputs.

Provide a summary of user-visible parameters (Table 9.2)

¹<mailto:achubaty@for-cast.ca>

TABLE 9.1: List of *burnSummaries* input objects and their description.

objectName	objectClass	desc	sourceURL
speciesLayers	RasterStack	initial percent cover raster layers used for simulation.	NA
speciesLayers	RasterStack	initial percent cover raster layers used for simulation.	NA

TABLE 9.2: List of *burnSummaries* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
reps	integer	1, 2, 3, ...	1	NA	number of replicates/runs per study area.
simOutputPath	character	/tmp/rtm...	NA	NA	Directory specifying the location of the simulation outputs.
upload	logical	FALSE	NA	NA	If TRUE, uses the 'googledrive' package to upload figures.
uploadTo	character	NA	NA	NA	If 'upload = TRUE', a Google Drive folder id corresponding to '.studyAreaName'.
.plots	character	screen	NA	NA	Used by Plots function, which can be optionally used here.
.plotInitialTime	numeric	0	NA	NA	Describes the simulation time at which the first plot event should occur.
.plotInterval	numeric	NA	NA	NA	Describes the simulation time interval between plot events.
.saveInitialTime	numeric	NA	NA	NA	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 - e.g., a hash of the studyarea obtained using 'reproducible::studyAreaName()'
.seed	list	NA	NA	NA	Named list of seeds to use for each event (names).
.useCache	logical	FALSE	NA	NA	Should caching of events or module be used?

TABLE 9.3: List of *burnSummaries* outputs and their description.

objectName	objectClass	desc
fireSizes	data.table	summary fire sizes table

9.1.3 Events

Describe what happens for each event type.

9.1.4 Plotting

Write what is plotted.

9.1.5 Saving

Write what is saved.

9.1.6 Module outputs

Description of the module outputs (Table 9.3).

9.1.7 Links to other modules

Describe any anticipated linkages to other modules, such as modules that supply input data or do post-hoc analysis.

9.1.8 Getting help

- provide a way for people to obtain help (e.g., module repository issues page)

References

10

LandWeb_output Module

10.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca¹ [aut, cre], Yong Luo yluo1@lakeheadu.ca² [aut], Alex M. Chubaty achubaty@for-cast.ca³ [ctb]

10.1 Module Overview

10.1.1 Module summary

This module saves LandR Biomass outputs for use with LandWeb post-processing.

10.1.2 Module inputs and parameters

Table 10.1 shows the full list of module inputs.

Provide a summary of user-visible parameters (Table 10.2)

Description of the module outputs (Table 10.3).

TABLE 10.2: List of *LandWeb_output* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
speciesType	character	NA	NA	NA	The column in simSpecs defining mixed stands; 1 for species admixture; 2 for deciduous + conifer. See <i>vegTypeMapGenerator</i> .
appEquivCol	character	LandWeb	NA	NA	The column in simSpecs/eqvalency data table to use as a naming convention
summaryInterval	numeric	50	NA	NA	This describes summary interval for this module
mapPlotGeneration	character	0.5	NA	NA	This describes the map plot generation interval for a given plot
plotInitialTime	numeric	0	NA	NA	This describes the simulation time at which the first plot event should occur
plotInterval	numeric	1	NA	NA	This describes the simulation time interval between plot events
plot	character	vector	NA	NA	Plot settings for this module. Plots will be made within this module, if set. Note that plots (or their data saving) will ONLY occur at random. If NA, plotting is turned off completely (this includes plot settings).
useCache	logical	FALSE	NA	NA	Should this entire module be run with caching activated? This is generally intended for data-type modules, where stochasticity and time are not relevant

TABLE 10.3: List of *LandWeb_output* outputs and their description.

objectName	objectClass	desc
vegTypeMap	Raster	NA

10.1.4 Links to other modules

Originally developed as part of the LandWeb⁴ project.

10.1.5 Getting help

https://github.com/fRI-Research/LandWeb_output/issues ## References

⁴<https://github.com/PredictiveEcology/LandWeb>

11

timeSinceFire Module

This documentation is work in progress. Please report any discrepancies or omissions at <https://github.com/fRI-Research/timeSinceFire/issues>.

11.0.0.1 Authors:

Steve G Cumming [¹](mailto:stevec@sbf.ulaval.ca) [aut, cre], Alex M. Chubaty [²](mailto:achubaty@for-cast.ca) [ctb]

11.1 Module Overview

11.1.1 Module summary

Yet Another Age Map Maintainer. This one is peculiar to the LandWeb application.

ageMap is incremented without bound on all flammable cells; cells identified as having been burned in the current year are set to 0.

Any statistics on age structure are to be calculated here, but none are yet implemented...because with the current fire model they would be pretty boring.

11.1.2 Module inputs and parameters

Table 11.1 shows the full list of module inputs.

A summary of user-visible parameters is provided in Table 11.2.

¹<mailto:stevec@sbf.ulaval.ca>

²<mailto:achubaty@for-cast.ca>

TABLE 11.1: List of *timeSinceFire* input objects and their description.

objectName	objectClass	desc	sourceURL
fireReturnInterval	RasterLayer	A Raster where the pixels represent the fire return interval, in years.	NA
rstCurrentBurn	RasterLayer	Binary raster of fires, 1 meaning 'burned', 0 or NA is non-burned	NA
rstFlammable	RasterLayer	A binary Raster, where 1 means 'can burn'.	NA
rstTimeSinceFire	RasterLayer	A Raster where the pixels represent the number of years since last burn.	NA

TABLE 11.2: List of *timeSinceFire* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
fireReturnInterval	numeric	NA	NA	NA	interval of time units between successive fire events.
returninterval	numeric	1	NA	NA	interval between main events
starttime	numeric	0	NA	NA	time of first burn event
plotinterval	numeric	NA	NA	NA	simulation time at which the first plot event should occur
plottimeval	numeric	NA	NA	NA	simulation time at which the first plot event should occur
plots	character	screen	NA	NA	Passed to <code>grid::grid</code> in <code>Plot</code> (see <code>Plot</code>). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at <code>trdsize</code> . If NA, plotting is turned off completely (this includes plot saving).
ageeventtime	numeric	NA	NA	NA	simulation time at which the first age event should occur
seedinterval	numeric	NA	NA	NA	simulation time at which the first seed event should occur
useCache	logical	FALSE	NA	NA	simulation time at which the first save event should occur

11.1.3 Events

11.1.3.1 Init

The `Init` event creates the RasterLayer `rstTimeSinceFire`. To do this, it rasterizes the template vegetation map `LCC05` using the `FireReturnInterval` field of the SpatialPolygonDataFrame `shpStudyRegion`. This procedure retains the NAs which mask the actual study region within the template bounding rectangle.

Then, the RasterLayer `rstFlammable` is used to mask out areas of open water, rock, etc which can't burn and thus for which `timeSinceFire` is not applicable. These become NAs in `rstTimeSinceFire`.

The results is that all flammable cells within each polygon in the shapefile are set to the fire return interval specified for that polygon / ecoregion. Under the basic van Wagner model being implemented, this is the expected landscape mean age. The ecoregion age structure will equilibrate to the exponential distribution within a few multiples of the return interval.

No colour ramp or legend is created for this layer.

In the short term, this initial uniform age distribution will result in very high proportions of cells with TSFs greater than the return interval. If this becomes a problem, one could initialize to the regional median age. This can be done by multiplying the `FireReturnInterval` by $\log(2)$ and then rounding; or some other lower quantile could be chosen: see the wikipedia page³ for the general quantile function.

Alternatively, a random exponential age structure could be generated for

³https://en.wikipedia.org/wiki/Exponential_distribution

TABLE 11.3: List of *timeSinceFire* outputs and their description.

objectName	objectClass	desc
burnLoc	integer	Cell indices where burns occurred in the latest year. It is derived from 'rstCurrentBurn'.
rstTimeSinceFire	RasterLayer	A Raster where the pixels represent the number of years since last burn.

each ecoregion from the current `rstTimeSinceFire`, roughly as follows. See the wiki page for details and possible alternative methods.

```
U_ <- runif(ncell(rstTimeSinceFire))
T_ <- (-log(U_)) * rstTimeSinceFire[]
rstTimeSinceFire[] <- round(T_)
```

11.1.3.2 Plotting

A bare call to `Plot(sim$rstTimeSinceFire)`. If you really want to see this, you'll have to live with the automated colour scheme and legend, or hack `Init` to your satisfaction.

11.1.3.3 Saving

Nothing is saved at present.

11.1.3.4 Age

This is the main event. `rstFlammable` is incremented by one. Then burned cells, as specified in the input vector `burnLoc` are set to age 0.

11.1.4 Module outputs

Description of the module outputs (Table 11.3).

11.1.5 Code and data availability

Code available from <https://github.com/fRI-Research/timeSinceFire>.

11.1.6 Links to other modules

Originally developed as part of the LandWeb⁴ project. ## References

⁴<https://github.com/PredictiveEcology/LandWeb>



12

LandWeb_summary Module

This documentation is work in progress. Please report any discrepancies or omissions at https://github.com/PredictiveEcology/LandWeb_summary/issues.

12.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca¹ [aut, cre], Alex M. Chubaty achubaty@for-cast.ca² [aut]

12.1 Module Overview

12.1.1 Module summary

LandWeb simulation post-processing and summary creation.

12.1.2 Module inputs and parameters

Table 12.1 shows the full list of module inputs.

Provide a summary of user-visible parameters (Table 12.2)

¹<mailto:eliot.mcintire@nrcan-rncan.gc.ca>

²<mailto:achubaty@for-cast.ca>

TABLE 12.1: List of *LandWeb_summary* input objects and their description.

objectName	objectClass	desc	sourceURL
ml	imap	map list object from <code>LandWeb_preamble</code>	NA
speciesLayers	RasterStack	initial percent cover raster layers used for simulation.	NA
sppColorVect	character	A named vector of colors to use for plotting. The names must be in <code>'sim\$sppEquiv[[P(sim)\$sppEquivCol]]'</code> , and should also contain a color for 'Mixed'	NA
sppEquiv	data.table	table of species equivalencies. See <code>'LandR:sppEquivalences_CA'</code> .	NA

TABLE 12.2: List of *LandWeb_summary* parameters and their description.

paramName	paramClass	default	min	max	paramDesc
ageClasses	character	Young, 1,...	NA	NA	descriptions/labels for age classes (serial stages)
ageClassSort	integer	1, 2, ..., 8,...	NA	NA	order of age classes (serial stages) in relation between age classes
ageClassMaxAge	integer	400	NA	NA	maximum possible age
reps	integer	1, 2, 3,...	1	NA	number of replicated/runs per study area.
simOutputPath	character	/tmp/Rm...	NA	NA	Directory specifying the location of the simulation outputs.
sppEquivCol	character	EN,gener...	NA	NA	The column in 'simSppEquiv' data.table to use as a naming convention
summaryInterval	integer	100	NA	NA	simulation time interval at which to take 'snapshots' used for summary analyses
summaryPeriod	integer	0, 1000	NA	NA	lower and upper bounds for the range of simulation time used for summary analyses
timeSeriesTimes	integer	600, 6000,...	NA	NA	times series times to build time series showing leading cover change over time
upload	logical	FALSE	NA	NA	If TRUE, uses the 'googledrive' package to upload figures.
uploadTo	character	NA	NA	NA	if 'upload = TRUE', a Google Drive folder id corresponding to 'studyAreaName'.
vegLeadingProportion	numeric	0.8	0	1	a number that defines whether a species is leading for a given pixel
version	integer	3	2	3	LandWeb model version (2 for runs using vegetation parameter forcings, else 3).
.dlnut	character	NA	NA	NA	Qualified environment to be evaluated on each node in a parallel cluster when running 'map' analyses. Useful to set options or create non-serializable objects on each node.
multiFiles	logical	FALSE	NA	NA	Used to evaluate each node in a parallel cluster when running 'map' analyses. Useful to set options or create non-serializable objects on each node.
plots	character	screen	NA	NA	Used by 'plot' function, which can be optionally used here.
plotInitialTime	numeric	0	NA	NA	Describes the simulation time at which the first plot event should occur.
plotInterval	numeric	NA	NA	NA	Describes the simulation time interval between plot events.
saveInitialTime	numeric	NA	NA	NA	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	Non-serializable name of the study area obtained using 'reproducible::studyAreaName()'
useCache	character	studyAreaName... NA	NA	NA	Name of event to be cached.
useParallel	logical	TRUE	NA	NA	Logical. If TRUE, and there is more than one calculation to do at any stage, it will create and use a parallel cluster via 'makeOptimalCluster()'

TABLE 12.3: List of *LandWeb_summary* outputs and their description.

objectName	objectClass	desc
ml	map	map list object

12.1.3 Module outputs

12.1.3.1 Large Patches Data for study region

- Large Patches Data (.csv)
- Large patches histograms (.png)

12.1.3.2 Leading Vegetation Cover Data for study region

- Leading Vegetation Cover Data (.csv)
- Leading Vegetation Cover histograms (.png)
- Leading Vegetation Cover boxplots (.png)

Description of the module outputs (Table 12.3).

12.1.4 Links to other modules

Initially developed for use with the LandWeb³ model.

12.1.5 Getting help

- https://github.com/PredictiveEcology/LandWeb_summary/issues ## References

³<https://github.com/PredictiveEcology/LandWeb>

Web app

As of spring 2023, fRI Research no longer hosts the LandWeb App. However, with appropriate access to the output data, the app may be run in a local shiny instance.

Previously available from <https://landweb.ca>.



13

Modifying LandWeb

13.1 Example 1: adding new reporting polygons

'Reporting polygons' refer to the polygons by which LandWeb results are summarized within the study area. These are created in the `LandWeb_preamble` module, and added to `sim$ml` (an object of class `map`, from the `map` package). Reporting polygons are not used during simulation, but rather during post-processing in the `LandWeb_summaries` module.

Adding a new reporting polygons is straightforward - one simply needs to edit the `LandWeb_preamble` code to 1) create the object and 2) add it to the `map` (*i.e.*, `sim$ml`) object. For reporting polygons retrieved from public or Google Drive URLs, both steps may be combined in a single `mapAdd()` call.

Downstream use requires each set of reporting polygons to have a `Name` (and `shinyLabel`) field (column). These fields are typically derived from a single field in the raw source, and are thus identical in value. Additionally, the helper function `joinreportingPolygons()` should be used following a geospatial intersection operation (*e.g.*, using `postProcess()`) to ensure that post-intersection labels like `Names.1` and `shinyLabel.1` are corrected.

13.1.1 National-scale polygons

When adding reporting polygons derived from *national-scale* data sources, the changes need to be made in `LandWeb_preamble.R` as well as each of the study-area-specific files in `LandWeb_preamble/R/` so that LandWeb runs using any study area can make use the new reporting polygons.

```
## LandWeb_preamble.R

## create the national-scale object and add to map object in a
single call
```

```

ml <- mapAdd(map = ml, layerName = "National Ecoregions",
             useSACrs = TRUE, poly = TRUE, overwrite = TRUE,
             url = ,
             "https://sis.agr.gc.ca/cansis/nsdb/ecostrat/region/ecoregion_shp.zip")
,
columnNameForLabels = "REGION_NAM", isStudyArea =
FALSE, filename2 = NULL)

## ensure there is a 'Name' field in the new object
ml[["National Ecoregions"]][["Name"]] <- ml[["National
Ecoregions"]][["REGION_NAM"]]

```

Next, we need to modify each of the regional files to make use of these new national polygons, but intersected and cropped to the study area:

```

## LandWeb_preamble/R/provSK.R

## 1. create the reporting polygons by intersecting the National
Ecoregions with SK admin boundary
SK.natler <- postProcess(ml[["National Ecoregions"]],
                           studyArea = SK, useSACrs = TRUE,
                           filename2 = file.path(dataDir,
                           "SK_NATLER.shp")) |>
joinReportingPolygons(SK)

## 2. add to the map object
ml <- mapAdd(SK.natler, ml, layerName = "SK NATLER", useSACrs =
TRUE, poly = TRUE,
             analysisGroupReportingPolygon = "SK NATLER",
             columnNameForLabels = "REGION_NAM", filename2 =
NULL)

```

13.1.2 Regional or study-area-specific polygons

If adding regional or study-area-specific reporting polygons (*e.g.*, for an FMA's active/passive landbase), then the new reporting polygons

should be added to the corresponding study area code file found in LandWeb_preamble/R.

```
## LandWeb_preamble/R/WestFraser.R

## 1. create the reporting polygons by downloading and cleaning
## up shapefile
wf_br.lbstatus <- Cache(
  prepInputs,
  url = "https://drive.google.com/file/d/1A7N_EIb02wMBI_YTmU2Z-
  bQwqC9sY_EC/",
  destinationPath = dataDir,
  targetFile = "BRL_Landbase.shp", alsoExtract = "similar",
  fun = "sf::st_read", studyArea = wf_br, useSAcrs = TRUE
)
wf_br.lbstatus <- wf_br.lbstatus[st_is_valid(wf_br.lbstatus), ]
## remove invalid geometries
wf_br.lbstatus <- wf_br.lbstatus[!st_is_empty(wf_br.lbstatus),
] ## remove empty polygons
wf_br.lbstatus <- Cache({
  mutate(wf_br.lbstatus, Name = LBC_LBStat, geometry = geometry,
  .keep = "used") |>
    group_by(Name) |>
    summarise(geometry = sf::st_union(geometry)) |>
    ungroup() |>
    mutate(shinyLabel = Name, .before = geometry) |>
    joinReportingPolygons(wf_br)
})

## 2. add to the map object
ml <- mapAdd(wf_br.lbstatus, ml, layerName = "West Fraser Blue
Ridge LBstatus", useSAcrs = TRUE, poly = TRUE,
  analysisGroupReportingPolygon = "West Fraser Blue
  Ridge LBstatus",
  columnNameForLabels = "Name", filename2 = NULL)
```

13.2 Example 2: adding a new study area

Here we will create a custom study area for an area in northwestern Alberta. We want to use the latest FMU map from Spring 2022 (<https://www.alberta.ca/forest-management-agreements>).

1. **Externally from LandWeb**, create a shapefile (or similar) for the new study area, dissolving any internal polygon boundaries.

```
nwab <- sf::st_read("AB_FMAs_2022-03-25/Forest  
Management Unit.shp") |>  
  subset(FMU_CODE %in% c("F10", "F11", "F14", "F15",  
  "F20", "F23", "F26", "F51", "M01",  
  "P08", "P14", "P19", "P20",  
  "P21", "P22", "P52", "P55",  
  "S10", "S11", "S14")) |>  
  sf::st_union()
```

2. **Externally from LandWeb**, upload the shapefile to Google Drive and record the file's Google Drive ID.

```
shpfile <- "inputs/NW_AB.shp"  
sf::st_write(nwab, shpfile)  
  
result <- googledrive::drive_put(shpfile, googledrive_<br/>  
  ::as_id("1LsYuuYICkcpElAkEABFM5zJXf5tTyMLG"))  
fid <- result$id
```

3. Determine the name to use for running LandWeb with your new study area, and modify `LandWeb_preamble.R` to use this new study area by name. Add a new case to the `if-else` block in `LandWeb_preamble.R`:

```

} else if (grep("customABNW", P(sim)$studyAreaName))
{
  ml <- customABNW(ml, P(sim)$studyAreaName, dataDir,
  sim$canProvs, P(sim)$bufferDist, asStudyArea = TRUE)

```

4. Add this new study area to the `map` object by wrapping the necessary components in a new function defined in a new file at `LandWeb_preamble/R`. It's simplest to use an existing function/file in `LandWeb_preamble/R` as a template, making modifications as appropriate for your new study area.
 - a. copy `LandWeb_preamble/R/provAB.R` to `LandWeb_preamble/R/NWAB.R`;
 - b. edit `LandWeb_preamble/R/NWAB.R`, to crop AB to the new study area (NWAB) and subsequently using NWAB in lieu of AB throughout;
 - c. remove any unnecessary elements (*e.g.*, planning units and planning regions).
5. You should now be able to run the LandWeb model setting '`.studyAreaName = "customABNW"`' and it will use the new study area.

13.3 Example 3: updating the LTHFC map

Using an updated version of a data source requires only basic code modifications provided the following conditions are met:

- spatial data attributes remain the same (*e.g.*, field names are the same);
- a new URL is provided to the new data source;
- the new file name is different from the previous version (*e.g.*, it has a version number or date).

Here we show how to modify the `LandWeb_preamble` module to use a hypothetical new version of the longs-term historic fire cycle (LTHFC) map.

1. looking at `LandWeb_preamble.R` we see that the current version of the file is `landweb_ltfc_v8.shp`.

2. create a new spatial layer and save as `landweb_ltfc_v9.shp`.
 3. zip the new shapefile (with auxiliary files) upload this zip file to Google Drive as `landweb_ltfc_v9.zip`, making note of the Google Drive share URL.
 4. modify the **two** `mapAdd()` calls in `LandWeb_preamble.R` that reference the previous version (v8) of the LTHFC map to use the new (v9) Google Drive URL.
-

13.4 Contributing changes

- via pull request¹ against development branch on GitHub

¹<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-a-pull-request>

References



Bibliography

- [1] David W Andison. “Managing for landscape patterns in the sub-boreal forests of British Columbia”. PhD Thesis. Vancouver, BC: University of British Columbia, 1996. URL: <https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0075275>.
- [2] David W. Andison. “Temporal patterns of age-class distributions on foothills landscapes in Alberta”. In: *Ecography* 21.5 (1998), pp. 543–550. ISSN: 09067590. DOI: [10.1111/j.1600-0587.1998.tb00446.x](https://doi.org/10.1111/j.1600-0587.1998.tb00446.x)².
- [3] Ceres Barros et al. “Empowering ecological modellers with a PERFECT workflow: Seamlessly linking data, parameterisation, prediction, validation and visualisation”. In: *Methods in Ecology and Evolution* 14 (2023), pp. 173–188. DOI: [10.1111/2041-210X.14034](https://doi.org/10.1111/2041-210X.14034)³.
- [4] A Beaudoin et al. “Mapping attributes of Canada’s forests at moderate resolution through kNN and MODIS imagery”. In: *Canadian Journal of Forest Research* 44 (2014), pp. 521–532. DOI: [10.1139/cjfr-2013-0401](https://doi.org/10.1139/cjfr-2013-0401)⁴.
- [5] A Beaudoin et al. *Species composition, forest properties and land cover types across Canada’s forests at 250m resolution for 2001 and 2011*. 2017. DOI: [10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990](https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990)⁵. URL: <http://open.canada.ca/data/en/dataset/ec9e2659-1c29-4ddb-87a2-6aced147a990>.
- [6] 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.
- [7] 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.

²<https://doi.org/10.1111/j.1600-0587.1998.tb00446.x>

³<https://doi.org/10.1111/2041-210X.14034>

⁴<https://doi.org/10.1139/cjfr-2013-0401>

⁵<https://doi.org/10.23687/EC9E2659-1C29-4DDB-87A2-6ACED147A990>

- [8] 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>.
- [9] John Cosco. COMMON ATTRIBUTE SCHEMA (CAS) FOR FOREST INVENTORIES ACROSS CANADA. Feb. 2011, p. 117.
- [10] 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/X08-001⁶. URL: <http://www.nrcresearchpress.com/doi/10.1139/X08-001>.
- [11] 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>, %20<https://github.com/PredictiveEcology/reproducible>.
- [12] 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: 10.2307/3237266⁷. URL: <http://doi.wiley.com/10.2307/3237266>.
- [13] NRCan. *National Ecological Framework for Canada - Terrestrial Ecozones*. 2013.
- [14] 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/nclimate1293⁸. URL: <http://www.nature.com/articles/nclimate1293>.
- [15] Paul D Pickell and Nicholas C Coops. *Development of historical forest attribute layers using Landsat time series and kNN imputation for the western Canadian boreal forest*. Tech. rep. University of British Columbia, Dec. 2016, pp. 1–27.
- [16] Robert M Scheller et al. “Design, development, and application of LANDIS-II, a spatial landscape simulation model with flexible temporal and spatial resolution”. In: *Ecological Modelling* 201 (2007), pp. 409–419. ISSN: 03043800. DOI: 10.1016/j.ecolmodel.2006.10.009⁹.

⁶<https://doi.org/10.1139/X08-001>

⁷<https://doi.org/10.2307/3237266>

⁸<https://doi.org/10.1038/nclimate1293>

⁹<https://doi.org/10.1016/j.ecolmodel.2006.10.009>

- [17] Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – Conceptual Description*. Apr. 20, 2012.
- [18] Robert M. Scheller and James B. Domingo. *LANDIS-II Model v6.0 – User Guide*. July 19, 2011.
- [19] Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.
- [20] Robert M. Scheller and Brian R. Miranda. *LANDIS-II Biomass Succession v3.2 Extension – User Guide*. 2015.
- [21] 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.2004.01.022¹⁰. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304380004003837>.
- [22] Brian R. Sturtevant et al. *LANDIS-II Dynamic Fire System Extension v3.0 – User Guide*. 2018.
- [23] Hadley Wickham. “testthat: Get Started with Testing”. In: *The R Journal* 3.1 (2011), p. 5. DOI: 10.32614/RJ-2011-002¹¹. URL: <https://journal.r-project.org/archive/2011/RJ-2011-002/index.html>.
- [24] 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.2016.18¹². URL: <http://www.nature.com/articles/sdata201618>.
- [25] 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.2011.04.032¹³. URL: <https://www.sciencedirect.com/science/article/pii/S0304380011002651>.

¹⁰ <https://doi.org/10.1016/j.ecolmodel.2004.01.022>

¹¹ <https://doi.org/10.32614/RJ-2011-002>

¹² <https://doi.org/10.1038/sdata.2016.18>

¹³ <https://doi.org/10.1016/j.ecolmodel.2011.04.032>