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

LandWeb Manual
v. 3.0.0

Contents

List of Figures

List of Tables

Overview

This manual is a live document which is automatically updated as changes are made to 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 [5] 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 [12, 8] 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.
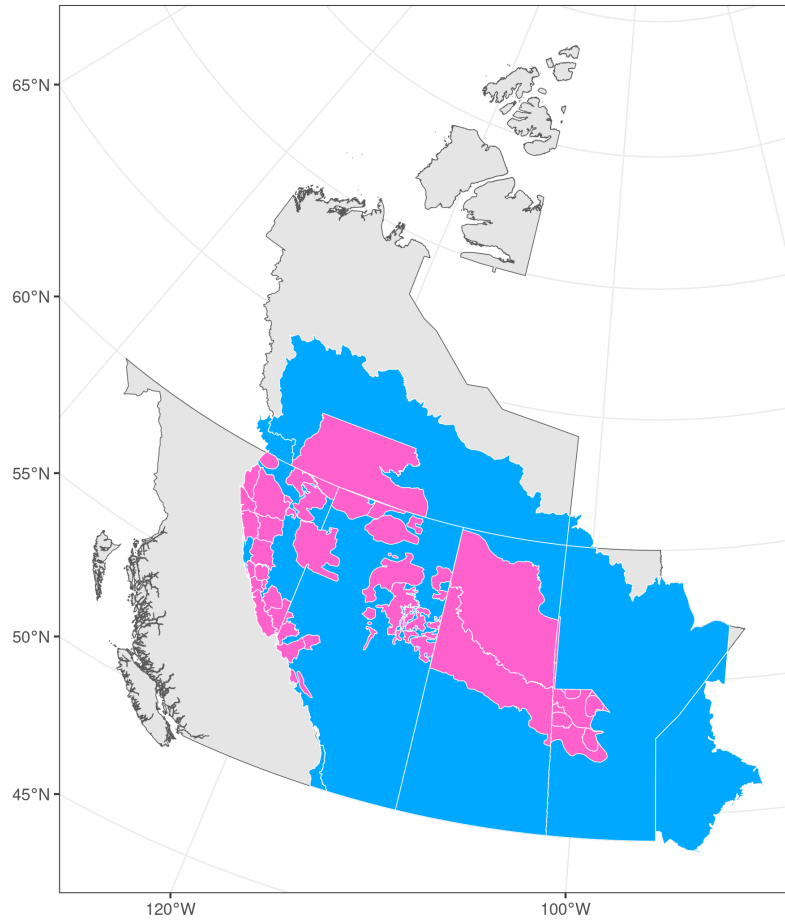
FIGURE 1: LandWeb study area (blue) with mountain and boreal caribou ranges highlighted (pink).

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 3]: 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 [7] 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.ca[1]).

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 [12, 8]. Our re-implemented model largely follows the original LANDIS-II source code (v 3.6.2; Scheller and Miranda [11]), 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 Andison'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 available from the links below.

- LandR Manual v3.0.0[2] (*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 `local-host:8080`.

Login to the Rstudio session as user `rstudio` and password `<mySecretPass-word>` (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

`Rtools` provides the necessary compilers etc. to build and install R packages from source on Windows.

Download[1] and install `Rtools` version 4.2 *as administrator*. During installation, be sure to check the option to add `Rtools` to your `PATH`.

1.1.2.2 macOS

1.1.2.2.1 *Xcode command line tools*

To build software, you will need the Xcode command line tools[2], which include various compilers and `git` version control software.

---

[1]https://cran.r-project.org/bin/windows/Rtools/rtools42/rtools.html
[2]https://developer.apple.com/downloads/

```
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 \
    curl \
    libarchive-dev \
    libcairo2-dev \
    libcurl4-openssl-dev \
    libgit2-dev \
    libglpk-dev \
    libgmp3-dev \
    libicu-dev \
    libjq-dev \
    libmagick++-dev \
    libnode-dev \
    libpng-dev \
    libprotobuf-dev \
    libprotoc-dev \
    libssh2-1-dev \
    libssl-dev \
```

```
    libxml2-dev \
    libxt-dev \
    make \
    pandoc pandoc-citeproc \
    protobuf-compiler \
    sysstat \
    texlive-bibtex-extra \
    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 to add some addi-
tional 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.

    :::{.rmdimportant} 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).

:::{.rmdwarning} 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 instructions[3];

   b.   Be sure to uncheck all scopes.

   c.   Copy this token and save it in a text file in your home directory called .Renviron.

```
## ~/.Renviron
GITHUB_PAT=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

   If you are not sure where your home directory is located, run the following in an R session:

```
Sys.getenv("HOME")
```

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.

---

[3]https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

1.1.5   R and Rstudio

1. Download and install R version 4.2.3.

    :::{.rmdimportant} **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

:::{.rmdcaution} **This is out of date, as R 4.3 is the most recent version of R.**

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/rig). :::

```
## install R
sudo apt-get -y install r-base r-base-dev
r-cran-littler
sudo apt-get build-dep -y r-cran-rjava r-cran-tkrplot
sudo R CMD javareconf
```

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

   :::{.rmdimportant} Windows users should install Rstudio *as administrator*. :::

3. (optional) On Linux, configure `ccache` to speed up R package re-installation and updates[4].

```
## 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)'; \
  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
```

---

[4]http://dirk.eddelbuettel.com/blog/2017/11/27/#011_faster_package_installation_one

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 --single-branch -b development \
  --recurse-submodules="." \
  --recurse-submodules=":(exclude)app" \
  --recurse-submodules=":(exclude)deploy" \
  -j8  https://github.com/PredictiveEcology/LandWeb
```

NOTE: the app and deploy submodule repos are *private*.

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.

TABLE 1.1: LandWeb project directory structure

directory	description
R/	additional R helper scripts
app/	web-app code (git submodule; private repo)
cache/	all per-run and per-study area cache files stored here
deploy/	web-app deployment code (git submodule; private repo)
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

1.4   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
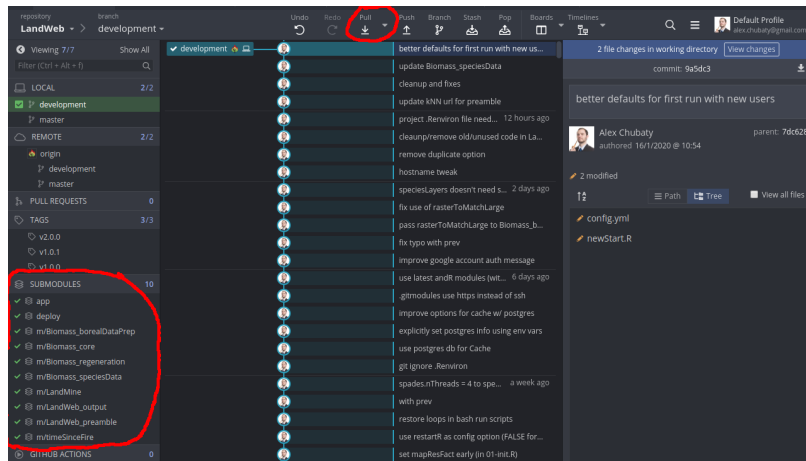
FIGURE 1.1: Screenshot showing showing code commits in Git Kraken. The submodules pane is highlighted on the bottom left.

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 --recursive
```

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.ca[5]).

1.6 Getting help

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

---

[5]mailto:achubaty@for-cast.ca

Study areas

Lorem ipsum ... (TODO)

2

LandR LandWeb_preamble *Module*

2.0.0.1 Authors:

Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca[1] [aut, cre], Alex M. Chubaty achubaty@for-cast.ca[2] [aut], Ceres Barros cbarros@mail.ubc.ca[3] [aut]

2.1 Module Overview

2.1.1 Module summary

Set up study areas and parameters for LandWeb simulations.

2.2 Parameters

Provide a summary of user-visible parameters.

---

[1] mailto:eliot.mcintire@nrcan-rncan.gc.ca
[2] mailto:achubaty@for-cast.ca
[3] mailto:cbarros@mail.ubc.ca

paramName	paramClass	default	min	max	paramDesc
bufferDist	numeric	25000	20000	1e+05	Study area buffer distance (m) used to make studyArea.
bufferDistLarge	numeric	50000	20000	1e+05	Study area buffer distance (m) used to make studyAreaLarge.
forceResprout	logical	FALSE	NA	NA	'TRUE' forces all species to resprout, setting 'resproutage_min' to zero, 'resproutage_max' to 400, and 'resproutProb' to 1.0.
friMultiple	numeric	1	0.5	2	Multiplication factor for adjusting fire return intervals.
dispersalType	character	default	NA	NA	One of 'aspen', 'high', 'none', or 'default'.
minFRI	numeric	40	0	200	The value of fire return interval below which, pixels will be

2.3 Data dependencies

2.3.1 Input data

Description of the module inputs.

objectName	objectClass	desc	sourceURL
canProvs	SpatialPolygonsDataFrame	Canadian provincial boundaries shapefile	NA

2.3.2 Output data

Description of the module outputs.

objectName	objectClass	desc
CC TSF	RasterLayer	Time since fire (aka age) map derived from Current Conditions data.
fireReturnInterval	RasterLayer	fire return interval raster
LandTypeCC	RasterLayer	Land Cover Classification map derived from Current Conditions data.
ml	map	'map' object containing study areas, reporting polygons, etc. for post-processing.
LCC	RasterLayer	A key output from this module: it is the result of LandR::overlayLCCs on LCC2005 and LandTypeCC
nonTreePixels	integer	NA
rasterToMatch	RasterLayer	NA
rasterToMatchLarge	RasterLayer	NA
rasterToMatchReporting	RasterLayer	NA
ROSTable	data.table	A data.table with 3 columns, 'age', 'leading', and 'ros'. The values under the 'age' column can be 'mature', 'immature', 'young' and compound versions of these, e.g., 'immature_young' which can be used when 2 or more age classes share same 'ros'. 'leading' should be vegetation type. 'ros' gives the rate of spread values for each age and type.
rstFlammable	RasterLayer	NA
speciesParams	list	list of updated species trait values to be used to updated 'speciesTable' to create 'species'.
speciesTable	data.table	a table of invariant species traits with the following trait colums:

2.4 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.

3

LandR Biomass_core *Module*

Biomass core v. 1.3.10 [1]

Get help Report issues [2]

3.0.0.1 Authors:

Yong Luo yluo1@lakeheadu.ca[3] [aut], Eliot J B McIntire eliot.mcintire@nrcan-rncan.gc.ca[4] [aut, cre], Ceres Barros ceres.barros@ubc.ca[5] [aut], Alex M. Chubaty achubaty@for-cast.ca[6] [aut], Ian Eddy ian.eddy@nrcan-rncan.gc.ca[7] [ctb], Jean Marchal jean.d.marchal@gmail.com[8] [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.

3.1 Module Overview

3.1.1 Quick links

- General functioning

- List of input objects

- List of parameters

---

[1] https://github.com/PredictiveEcology/Biomass_core
[2] https://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

3.1.2   Summary

LandR *Biomass_core* (hereafter *Biomass_core*) is the core forest succession simulation module of the LandR ecosystem of SpaDES modules [see 5]. It simulates tree cohort ageing, growth, mortality and competition for light resources, as well as seed dispersal (Fig. 3.1), 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, 11], 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 [11] for a detailed reading of the mechanisms implemented in the model.
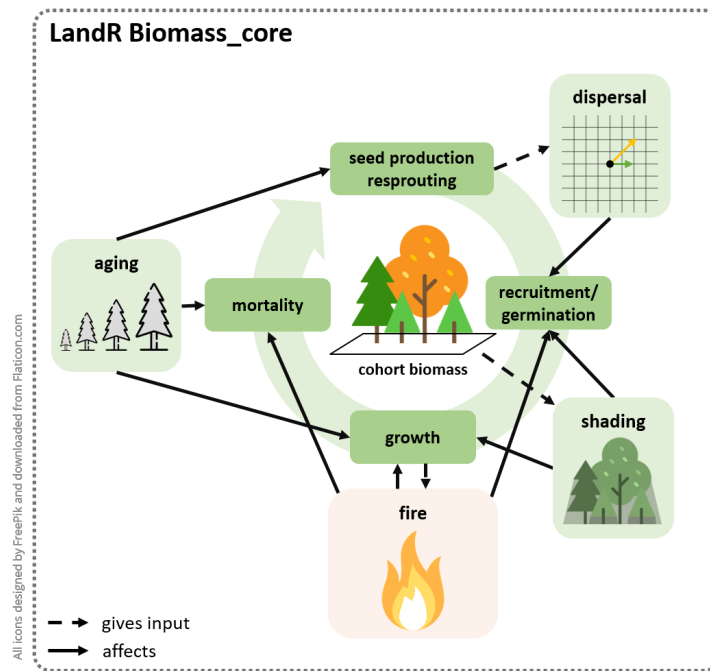
FIGURE 3.1: *Biomass_core* simulates tree cohort growth, mortality, recruitment and dispersal dynamics, as a function of cohort ageing and competition for light (shading) and space, as well as disturbances like fire (simulated using other modules).

3.1.3   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 Manual[9] see the individual module's documentation (`.Rmd` file) available in its repository.

See here[10] for all available modules and select *Biomass_core* from the dropdown 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 [13];

- *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/
[10] https://rpubs.com/PredictiveEcology/LandR_Module_Ecosystem
[11] https://github.com/PredictiveEcology/Biomass_speciesData
[12] https://github.com/PredictiveEcology/Biomass_borealDataPrep
[13] https://github.com/PredictiveEcology/Biomass_speciesParameters
[14] https://github.com/PredictiveEcology/Biomass_regeneration
[15] https://github.com/PredictiveEcology/Biomass_regenerationPM

mate, land-cover, fire occurrence and fire perimeter data. Requires using *Biomass_regeneration* or *Biomass_regenerationPM*. See modules prefixed *"fire-Sense_"* 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.

3.2 Module manual

3.2.1 General functioning

Biomass_core is a forest landscape model based on the LANDIS-II Biomass Succession Extension v.3.2.1 model [LBSE, 11]. 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. 3.1). Note that, by default, cohorts are unique combinations of species and age, but this can be changed via the cohortDefinitionCols parameter (see List of parameters).

Specifically, cohort growth is driven by both invariant (growth shape parameter, growthcurve) and spatio-temporally varying species traits (maximum biomass, maxB, and maximum annual net primary productivity, max-

---

[16]https://github.com/PredictiveEcology/LandMine
[17]https://github.com/PredictiveEcology/scfm
[18]https://github.com/PredictiveEcology/Biomass_validationKNN

ANPP), 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 [11]. 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' [11]. 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 [11], Scheller and Domingo [10] and Scheller and Domingo [9] 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.

3.2.2 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 LBSE[19], *Biomass_core* initialises the simulation using data-derived initial cohort biomass and age. This information is ideally supplied by data and calibration modules like *Biomass_borealDataPrep* (Links to other modules), but *Biomass_core* can also initialise itself using theoretical data.

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

3.2.2.1 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

---

[19]in 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) [11].

(`speciesLayers` raster layers) data (see Table 3.7) and formatted into the `co-hortData` 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).

3.2.2.2 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 3.7), 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 4], 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 3.1 shows an example of a table of invariant species traits. Note that *Biomass_core* (alone) requires all the columns Table 3.1 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 [10, p.18] and Scheller and Miranda [11, p.16] for further detail.

TABLE 3.1: 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
Abie_sp	200	20	2.3	1
Pice_eng	460	30	2.1	2
Pice_gla	400	30	1.6	2
Pinu_sp	150	15	1	2
Popu_sp	140	20	1	1
Pseu_men	525	25	2	3

TABLE 3.2: 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

3.2.2.3 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 3.4 for an example.

TABLE 3.4: 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	Abie_sp	1	8567	285	1
1_03	Pice_eng	0.983	10156	305	1
1_03	Popu_sp	0.737	8794	293	1
1_03	Pseu_men	1	17534	132	1
1_09	Abie_sp	0.112	1499	50	1
1_09	Pice_gla	0.302	3143	102	1
1_09	Pinu_sp	0.714	2569	86	1
1_09	Popu_sp	0.607	3292	110	1
1_09	Pseu_men	0.997	6020	45	1
1_03	Abie_sp	0.989	8943	225	2
1_03	Pice_eng	0.985	9000	315	2
1_03	Popu_sp	0.6	8600	273	2
1_03	Pseu_men	1	13534	142	2
1_09	Abie_sp	0.293	2099	45	2
1_09	Pice_gla	0.745	3643	90	2
1_09	Pinu_sp	0.5	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

3.2.2.4   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 [11].

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 table[20] to better reflect Western Canada boreal forest dynamics (see Table 3.5). *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 3.5: Example of a minimum relative biomass table (the `minRelativeB` table object in the module), with two ecolocations (`ecoregionGroups`) sharing the same values

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

3.2.2.5   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 3.1) and the site shade [defined by the amount of biomass in the pixel – see mini-

---

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

mum relative biomass parameter and 11, p.14]. By default, both *Biomass_core* and *Biomass_borealDataPrep* use a publicly available LANDIS-II table (called `sufficientLight` in the module; Table 3.6).

TABLE 3.6: 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

3.2.2.6 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.

3.2.3 List of input objects

All of *Biomass_core*'s input objects have (theoretical) defaults that are produced automatically by the module[21]. 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.

---

[21]usually, 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 3.7, 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 `species-Layers` 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 if 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 [11].

• 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 3.7: List of *Biomass_core* input objects and their description.

objectName	objectClass	desc	sourceURL
biomassMap	RasterLaye r	total biomass raster layer in study area (in g/m^2), filtered for pixels cover ed by `cohortData`. Only used if `P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.	
cceArgs	list	a list of quoted objects used by the `gr owthAndMortalityDriver calculateClimat eEffect` function	NA
cohortData	data.table	`data.table` with cohort-level informati on on age and biomass, by pixelGroup and ecolocation (i.e., `ecoregionGroup`). I f supplied, it must have the following c olumns: `pixelGroup` (integer), `ecoregi onGroup` (factor), `speciesCode` (factor), `B` (integer in g/m^2), age (inte ger in years)	NA

objectName	objectClass	desc	sourceURL
ecoregion	data.table	ecoregion look up table	https://ra w.githubus ercontent. com/LANDIS -II-Founda tion/Exten sions-Succ ession/mas ter/biomas s-successi on-archive /trunk/tes ts/v6.0-2. 0/ecoregio ns.txt
ecoregionM ap	RasterLaye r	ecoregion map that has mapcodes match ec oregion table and `speciesEcoregion` tab le. Defaults to a dummy map matching `ra sterToMatch` with two regions	NA
lastReg	numeric	an internal counter keeping track of whe n the last regeneration event occurred	NA
minRelativ eB	data.frame	table defining the relative biomass cut points to classify stand shadeness.	NA

objectName	objectClass	desc	sourceURL
pixelGroup Map	RasterLaye r	a raster layer with `pixelGroup` IDs per pixel. Pixels are grouped based on iden tical `ecoregionGroup`, `speciesCode`, `age` and `B` composition, even if the us er supplies other initial groupings (e.g ., via the `Biomass_borealDataPrep` modu le.	NA
rasterToMa tch	RasterLaye r	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 colums: 'species', 'Area', 'longevity', 'sexualmature', 'sh adetolerance', 'firetolerance', 'seeddis tance_eff', 'seeddistance_max', 'resprou tprob', 'mortalityshape', 'growthcurve', 'resproutage_min', 'resproutage_max', ' postfireregen', 'wooddecayrate', 'leaflo ngevity' 'leafLignin', 'hardsoft'. The l ast seven traits are not used in Biomass _core , and may be ommited. However, thi s may result in downstream issues with o ther modules. Default is from Dominic Cy r and Yan Boulanger's project	https://ra w.githubus ercontent. com/dcyr/L ANDIS-II_I A_generalU seFiles/ma ster/speci esTraits.c sv
speciesEco region	data.table	table of spatially-varying species trait s (`maxB`, `maxANPP`, `establishprob`), defined by species and `ecoregionGroup`) Defaults to a dummy table based on dumm y data os biomass, age, ecoregion and la nd cover class	NA

objectName	objectClass	desc	sourceURL
speciesLay ers	RasterStac k	percent cover raster layers of tree spec ies in Canada. Defaults to the Canadian Forestry Service, National Forest Invent ory, kNN-derived species cover maps from 2001 using a cover threshold of 10 - se e https://open.canada.ca/data/en/dataset /ec9e2659-1c29-4ddb-87a2-6aced147a990 fo r metadata	http://ftp .maps.cana da.ca/pub/ nrcan_rnca n/Forests_ Foret/cana da-forests -attribute s_attribut s-forests- canada/200 1-attribut es_attribu ts-2001/
sppColorVe ct	character	A named vector of colors to use for plot ting. The names must be in `sim$sppEquiv [[sim$sppEquivCol]]`, and should also co ntain a color for 'Mixed'.	NA
sppEquiv	data.table	table of species equivalencies. See `Lan dR::sppEquivalencies_CA`.	NA

objectName	objectClass	desc	sourceURL
sppNameVec tor	character	an optional vector of species names to b e pulled from `sppEquiv`. Species names must match `P(sim)$sppEquivCol` column i n `sppEquiv`. If not provided, then spec ies will be taken from the entire `P(sim)$sppEquivCol` column in `sppEquiv`. See `LandR::sppEquivalencies_CA`.	NA
studyArea	SpatialPol ygonsDataF rame	Polygon to use as the study area. Must b e provided by the user	NA
studyAreaR eporting	SpatialPol ygonsDataF rame	multipolygon (typically smaller/unbuffer ed than studyArea) to use for plotting/r eporting. Defaults to `studyArea`.	NA

objectName	objectClass	desc	sourceURL
sufficient Light	data.frame	table defining how the species with diff erent shade tolerance respond to stand s hade. Default is based on LANDIS-II Biom ass Succession v6.2 parameters	https://ra w.githubus ercontent. com/LANDIS -II-Founda tion/Exten sions-Succ ession/mas ter/biomas s-successi on-archive /trunk/tes ts/v6.0-2. 0/biomass- succession _test.txt
treedFireP ixelTableS inceLastDi sp	data.table	3 columns: `pixelIndex`, `pixelGroup`, a nd `burnTime`. Each row represents a for ested pixel that was burned up to and in cluding this year, since last dispersal event, with its corresponding `pixelGrou p` and time it occurred	NA

3.2.4 List of parameters

In addition to the above inputs objects, *Biomass_core* uses several parameters[23] 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 package[24] (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 3.8. 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;

---

[23]in `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 3.8: 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 succesion 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`
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

paramName	paramClass	default	min	max	paramDesc
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)
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

paramName	paramClass	default	min	max	paramDesc
gmcsMortLimits	numeric	66.66666....	NA	NA	if using `LandR.CS` for climate-sensitive growth and mortality, a percentile is used to estimate the effect of climate on growth/mortality (currentClimate/referenceClimate). Upper and lower limits are suggested to circumvent problems caused by very small denominators as well as predictions outside the data range used to generate the model
gmcsMinAge	numeric	21	0	NA	if using `LandR.CS` for climate-sensitive growth and mortality, the minimum age for which to predict climate-sensitive growth and mortality. Young stands (< 30) are poorly represented by the PSP data used to parameterize the model.
growthAndMortalityDrivers	character	LandR	NA	NA	package name where the following functions can be found: `calculateClimateEffect`, `assignClimateEffect` (see `LandR.CS` for climate sensitivity equivalent functions, or leave default if this is not desired)
growthInitialTime	numeric	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)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.
sppEquivCol	character	Boreal	NA	NA	The column in `sim$sppEquiv` data.table to use as a naming convention

paramName	paramClass	default	min	max	paramDesc
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 `P(sim)$.plots`.
.plotInterval	numeric	NA	NA	NA	defines the plotting time step. If `NA`, the default, `.plotInterval` is set to `successionTimestep`.
.plots	character	object	NA	NA	Passed to `types` in `Plots` (see `?Plots`). There are a few plots that are made within this module, if set. Note that plots (or their data) saving will ONLY occur at `end(sim)`. If `NA`, plotting is turned off completely (this includes plot saving).

paramName	paramClass	default	min	max	paramDesc
.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`.
.sslVerify	integer	64	NA	NA	Passed to `httr::config(ssl_verifypeer = 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	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`
.useParallel	ANY	2	NA	NA	Used only in seed dispersal. If numeric, it will be passed to `data.table::setDTthreads` and should be <= 2; If `TRUE`, it will be passed to `parallel::makeCluster`; and if a cluster object, it will be passed to `parallel::parClusterApplyB`.

3.2.5 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 3.9) can be saved via the `outputs` argument in `simInit`[25].

TABLE 3.9: 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 `cohortData`. Only used if `P(sim)$initialBiomassSource == 'biomassMap'`, which is currently deactivated.

---

[25]see `?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., cliamte 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.
inactivePixelIndexReporting	integer	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 `P(sim)$calibrate == TRUE`, an summary of seed dispersal and germination success (i.e., number of pixels where seeds successfully germinated) per species and year.
reproductionMap	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).
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
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`.

3.2.6   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 `si-mInit` and the `init` event[26]);

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 [9] 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 [12] for further detail.

6. Cohort age binning (every `successionTimestep`; `cohortAgeRe-classification` 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 [11] for further detail.

   7.   Summary tables of regeneration (`summaryRegen` event), biomass, age, growth and mortality (`summaryBGM` event);

   8.   Plots of maps (`plotMaps` event) and averages (`plotAvgs` and `plot-SummaryBySpecies` events);

   9.   Save outputs (`save` event).

… (repeat 2-9) …

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

3.2.7.1   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, 11]. 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 3.10 and Fig. 3.2). The calculation of competition, growth and mortality for all cohorts at the same time also circumvented this issue.
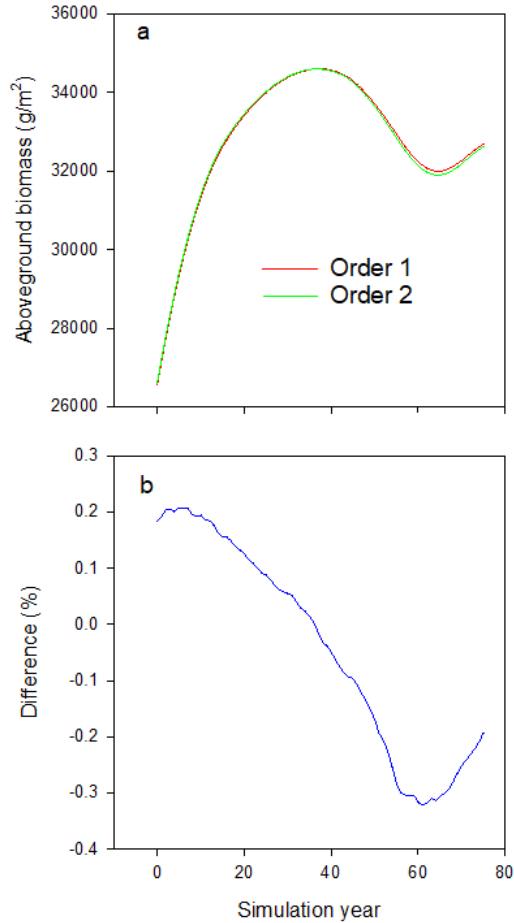
FIGURE 3.2: Differences in total landscape aboveground biomass when using two different input species orders for the same community. These simulations demonstrate how the sequential calculation of the competition index, combined with a lack of explicit species ordering affect the overall landscape aboveground biomass in time when using different input species orders (see Table

reftab:tableLBSEtest1). In order to prevent differences introduced by cohort recruitment, species' ages at sexual maturity were changed to the species' longevity values, and the simulation ran for 75 years to prevent any cohorts from reaching sexual maturity. The bottom panel shows the difference between the two simulations in percentage, calculated as
$$\frac{Biomass_{order2} - Biomass_{order1}}{Biomass_{order2}} * 100$$

Third, in LBSE the calculation of total pixel biomass for the purpose of calculating the initial biomass of a new cohort included the (previously calculated) biomass of other new cohorts when succession time step = 1, but not when time step was > 1. This does not reflect the documentation in the User's Guide, which stated that "*Bsum [total pixel biomass] is the current total biomass for the site (not including other new cohorts)*" [11, 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 3.12 and Fig. 3.3). 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.
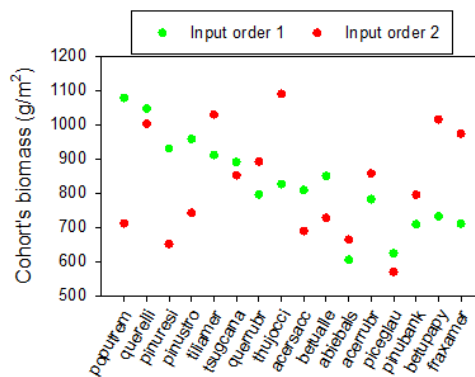
FIGURE 3.3: Differences in the biomass assigned to new cohorts, summed for each species across pixels, when using two different input species orders for the same community and when the succession time step is 1. These simulations demonstrate how the different summation of total cohort biomass for a succession 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 bellow a certain threshold of biomass. In some simulation set-ups, we noticed that *Biomass_core* (and LBSE) were able to generate many very small cohorts in the understory that, due to cohort competition, were not able to gain biomass and grow. However, because competition 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).

3.2.7.2   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.

3.2.7.2.1   *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

---

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

3.2.7.2.2   *Hashing*

Our first strategy to improve simulation efficiency in *Biomass_core* was to use a hashing mechanism [15]. 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. 3.4). 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.
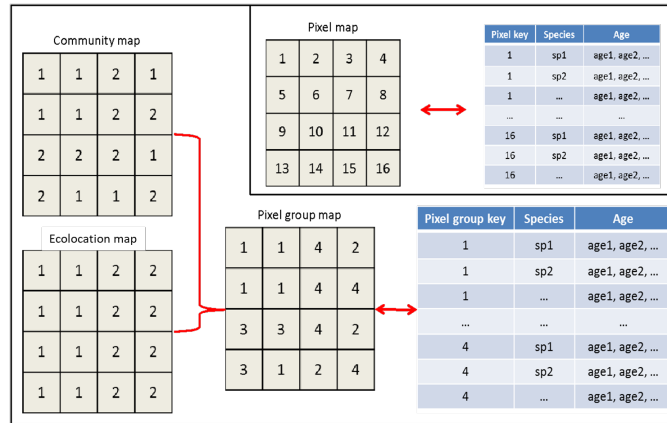
FIGURE 3.4: Hashing design for *Biomass_core*. In the re-coded *Biomass_core*, the pixel group map was hashed based on the unique combination of species composition ('community map') and ecolocation map, and associated with a lookup table. The insert in the top-right corner was the original design that linked the map to the lookup table by pixel key.

3.2.7.2.3   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 [6].

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.

3.2.7.2.4 *Testing*

Finally, we implemented code testing to facilitate bug detection by comparing the outputs of functions (etc.) to expected outputs [14]. 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.

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

In the recoding of *Biomass_core*, we used integration tests to ensured 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 3.14-3.16). The remaining input parameters were taken from a LANDIS-II training course (Tables 3.17-3.21), 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. 3.5), 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. 3.5 of this appendix).
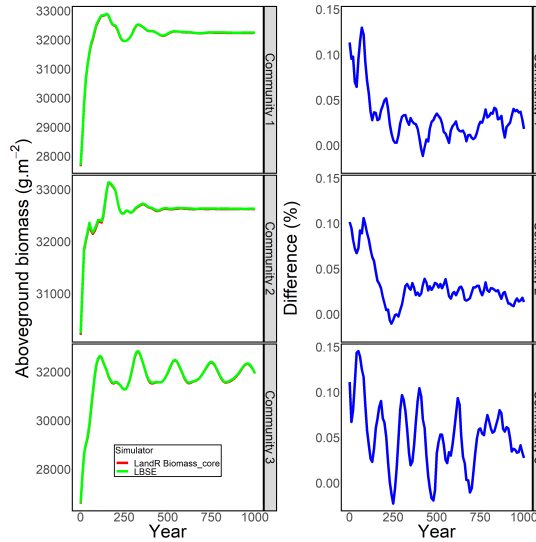
FIGURE 3.5: Visual comparison of simulation outputs for three randomly generated initial communities (left panels) and difference between those outputs (right panels). The % difference between LBSE and *Biomass_core* were calculated as $\frac{Biomass_{LBSE} - Biomass_{Biomass_{core}}}{Biomass_{LBSE}} * 100$

To examine how running time changed with map size, we ran simulations using maps with increasing number of pixels, from 22,201 to 638,401 pixels. All maps were initialised with a single ecolocation and 7 different communities. Simulations were run for 120 years using a succession time step of 10 and replicated three times. To eliminate the effect of hardware on running time, we used machines that were all purchased at the same time, with equal specifications and running Windows 7. Each simulation ran on 2 CPU threads with a total RAM of 4000 Mb.

For both LBSE and *Biomass_core*, the simulation time increased linearly with number of pixels, but the increase rate was smaller for *Biomass_core* (Fig. 3.6a). 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. 3.6a). *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. 3.6b).
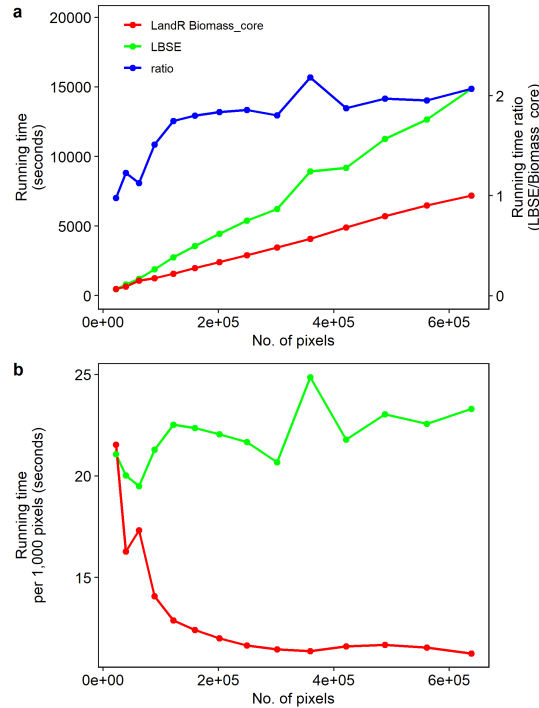
FIGURE 3.6: Simulation efficiencies of LBSE and *Biomass_core* with increasing map size, in terms of a) mean running time across repetitions (left y-axis) and the ratio LBSE to *Biomass_core* running times (right y-axis and blue line), and b) running time scalability as the mean running time per 1000 pixels.

3.3   Usage example

3.3.1   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()
```

3.3.2   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 repository[29] to see how to download this and other SpaDES modules, or fork/clone from its GitHub repository[30] 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")),
    cachePath = normPath(file.path(tempDir, "cache")),
    modulePath = normPath(file.path(tempDir,
        "modules")), outputPath = normPath(file.path(tempDir,
```

---

[29]https://github.com/PredictiveEcology/SpaDES-modules
[30]https://github.com/PredictiveEcology/Biomass_core/

```
        "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(unname(unlist(outs)), "SpaDES"), require = FALSE,
standAlone = TRUE)

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

3.3.3   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 `pixel-BroupMap` 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
```

```r
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))
```

3.3.4 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").

```r
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)
```
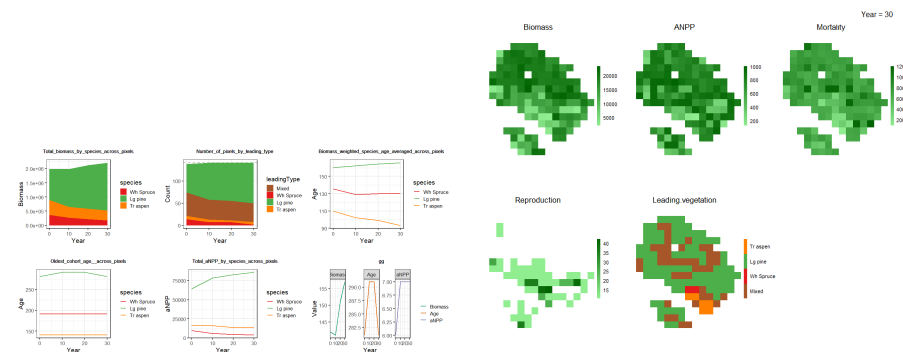

FIGURE 3.7: *Biomass_core* automatically generates simulation visuals of species dynamics across the landscape in terms of total biomass, number of presences and age and productivity (above), as well as yearly plots of total biomass, productivity, mortality, reproduction and leading species in each pixel (below).

3.4   Appendix

3.4.1   Tables

TABLE 3.10: 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 3.12: 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
pinustro	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	pinustro
tiliamer	1	poputrem
tsugcana	1	pinuresi
abiebals	1	acersacc
betupapy	1	betualle
fraxamer	1	piceglau

TABLE 3.14: 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 3.15: 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 3.16: 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 3.17: 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

Seeddistance_max	Mortalityshape	Growthcurve

TABLE 3.19: 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 3.20: 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 3.21: 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

Ecolocation	Species	SEP	maxANPP	maxB
1	pinuresi	0.56	1017	30510
1	pinustro	0.72	1090	38150
1	poputrem	1	1078	32340
1	querelli	0.96	1096	32880
1	querrubr	0.66	1017	30510
1	thujocci	0.76	1090	32700
1	tiliamer	0.54	1078	32340
1	tsugcana	0.22	1096	32880

3.5   References

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

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

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

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

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

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

[12]   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[31]. URL: https://linkinghub.elsevier.com/retrieve/pii/S0304380004003837.

---

[31]https://doi.org/10.1016/j.ecolmodel.2004.01.022

[13]   Brian R. Sturtevant et al. *LANDIS-II Dynamic Fire System Extension v3.0 – User Guide*. 2018.

[14]   Hadley Wickham. "testthat: Get Started with Testing". In: *The R Journal* 3.1 (2011), p. 5. DOI: 10.32614/RJ-2011-002[32]. URL: https://journal.r-project.org/archive/2011/RJ-2011-002/index.html.

[15]   Jian Yang et al. "An innovative computer design for modeling forest landscape change in very large spatial extents with fine resolutions". In: *Ecological Modelling* 222.15 (Aug. 10, 2011), pp. 2623–2630. DOI: 10.1016/j.ecolmodel.2011.04.032[33]. 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

Wildfire submodel

Lorem ipsum ... (TODO)

Model outputs

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

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.

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[34].

[3]   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[35].

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

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

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

[7]   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.

[8]   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.2006.10.009[36].

---

[34] https://doi.org/10.1111/j.1600-0587.1998.tb00446.x
[35] https://doi.org/10.1139/cjfr-2013-0401
[36] https://doi.org/10.1016/j.ecolmodel.2006.10.009

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

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

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

[12]   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[37]. URL: https://link inghub.elsevier.com/retrieve/pii/S0304380004003837.

[13]   Brian R. Sturtevant et al. *LANDIS-II Dynamic Fire System Extension v3.0 – User Guide*. 2018.

[14]   Hadley Wickham. "testthat: Get Started with Testing". In: *The R Journal* 3.1 (2011), p. 5. DOI: 10.32614/RJ-2011-002[38]. URL: https://journal.r-pr oject.org/archive/2011/RJ-2011-002/index.html.

[15]   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: 10.1 016/j.ecolmodel.2011.04.032[39]. URL: https://www.sciencedirect.com /science/article/pii/S0304380011002651.

---

[37]https://doi.org/10.1016/j.ecolmodel.2004.01.022
[38]https://doi.org/10.32614/RJ-2011-002
[39]https://doi.org/10.1016/j.ecolmodel.2011.04.032