Biomass_core

Jean Marchal (jean.d.marchal@gmail.com), Ceres Barros (cbarros@mail.ubc.ca), Eliot McIntire, Yong Lu

25 January 2021

This documentation is not complete. If there are discrepancies, omissions, or it is unclear, this is expected

Overview

A biomass succession model derived and modified from LANDIS-II Biomass Succession v3.2.1. Biomass_core simulates forest succession in a spatially explicit manner (per pixel) by reproducing the population dynamics (growth and mortality), dispersal and competition for light resources of tree species. Like in LANDIS-II, dynamics are simulated in terms of their biomass per cohort (a combination of species and age), following a `biomass ~ age` curve that is influenced by species-specific parameters such as growth, mortality, maximum biomass, and species traits such as longevity and tolerance to shade (see LANDIS-II Biomass Succession v3.2 User Guide and Scheller and Mladenoff (2004). One of three dispersal algorithms are available: 'no dispersal', 'universal dispersal'(every species can disperse across the entire landscape) and 'ward dispersal' (species disperse according to a dispersal kernel, with mean and maximum distances taken from LANDIS-II trait tables), the latter being used by default. The biggest differences between Biomass_core and LANDIS-II lie on how the model is parametrised and initialized and how different processes have been compartmentalised in order for higher flexibility.

Inputs and parametrisation

`Biomass_core` is capable of running on dummy datasets from which it estimates parameters linked to vegetation growth and seed germination (such as the maximum biomass per species, per pixel, and the probability of seed germination - *i.e.*, species establishment probability not due to resprouting), but also builds and initialises forest communities (based on biomass, age, species composition, land cover and ecological zones like ecodistricts. Ideally, however, the user should supply realistic versions of these data and the essential initialisation objects that `Biomass_core` requires to run. Below are two lists of input objects that the user should supply to `Biomass_core` in order for it to run on a realistic setting (this is not, however, an exhaustive list of inputs)

Required inputs	Description
`biomassMap`	map of total biomass in the study area
`ecoregion`	a table listing values in ecoregionMap and their state (active or not)
`minRelativeB`	table defining the cut points to classify stand shadiness
`rasterToMatch`	the map of the study area.
`species`	a table of species traits (based on LANDIS-II species traits for [v3.2.1] (https://github.com/LANDIS-II-Foundation/Extensions-Succession-Archive/blob /master/biomass-succession-archive/trunk/tests/v6.0-2.0/species.txt))
`sufficientLight`	table defining species responses to shade according to shade tolerance levels

Init. objects	Description
`cohortData`	table with cohort information by pixelGroup
`ecoregionMap`	map of ecological zones in the study area
`pixelGroupMap`	map of pixelGroup
`speciesEcoregion`	a table of species traits by for ecological zone in ecoregionMap

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

Besides the above mentioned inputs, `Biomass_core` uses the following parameters, which can be changed by the user if need be:

Parameter	Description
`calcSummaryBGM`	determines when biomass, growth and mortality will be summarized and plotted
`calibrate`	activates calibration if LANDIS-II-type spin-up is active (not implemented)
`initialBiomassSource`	determines how biomass will be initialised (only based on observed data at this point)
`seedingAlgorithm`	type of seeding algorithm (see above)
`spinupMortalityfraction`	if using LANDIS-II spin-up, this parameter increases mortality for all cohorts (not implemented)
`sppEquivCol`	column of species equivalency table (`sppEquiv`) to use to extract species names
`successionTimestep`	time step of succession dynamics (ideally 10, but growth/mortality will still happen every year)
`vegLeadingProportion`	relative proportion determining species dominance in a pixel (used to plot leading vegetation maps)
`.plotInitialTime`	controls first year of plotting
`.saveInitialTime`	controls first year of saving
`.useCache`	controls caching
`.useParallel`	controls parallelisation

Initialization

Unlike the initialisation in LANDIS-II, which "iterates the number of time steps equal to the maximum cohort age for each site", beginning at t - oldest cohort age and adding cohorts at the appropriate time (Scheller & Miranda 2016), Biomass_core initialises the simulation by deriving initial biomasses from available data.

Compartmentalisation and modularity

While in LANDIS-II phenomena like growth and mortality occur as part of the same process (or function from a coding perspective), in Biomass_core they occur separately and sequentially. This means that they can be more easily upgraded/changed. The same happens with post-disturbance regeneration. Unlike in LANDIS-II, post-disturbance regeneration is not part of Biomass_core *per se*, but rather belongs to a separate module, that needs to be loaded should the user want to simulate disturbance effect (*i.e.*, fire disturbances). Again, this enables higher flexibility when swapping between different approaches to regeneration. For instance, default (*i.e.* not climate sensitive) growth and mortality functions are part of the `LandR` R package, which needs to be loaded prior to running `Biomass_core`. Should the user wish to change the growth/mortality algorithms, they would need to provide compatible functions (with the same names) to the simulation via `simInit` - user-provided functions will replace those loaded with a package. + Note: The `LandR` package

provides other supporting functions and objects to the simulation, and still needs to be loaded prior to running `Biomass_core`.

Simulation flow

No disturbances

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

1. Preparation of necessary objects for the simulation - either by accessory data prep. modules, or Biomass_core itself (using LANDIS-II test parameters and dummy data for stand age, biomass and land cover and ecological zoning)
2. Seed dispersal - see Section 4.5.1 Seeding of the LANDIS-II Model v7.0 Description(https://drive.google.com/file/d/15gSueug-Rj9I2RZqdroDbad-k53Jq7j3/view) for details

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

3. Growth and mortality - based on Scheller and Mladenoff (2004)

- unlike dispersal, growth and mortality should occur every year

4. Ageing - based on Scheller and Mladenoff (2004)

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

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

With disturbances

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

1. Preparation of necessary objects for the simulation - either by accessory prep. data modules, or Biomass_core itself (using LANDIS-II test parameters and dummy data.)
2. Disturbances - simulated by a disturbance module
3. Post-disturbance regeneration - simulated by a regeneration module (`Biomass_regeneration` is an optional download)
4. Seed dispersal - see Section 4.5.1 Seeding of the LANDIS-II Model v7.0 Description(https://drive.google.com/file/d/15gSueug-Rj9I2RZqdroDbad-k53Jq7j3/view) for details
5. Growth, ageing and mortality - based on Scheller and Mladenoff (2004)
6. Preparation of visual/saved outputs . . . (repeat 2-6) . . .

Usage example

Load SpaDES

```
library(SpaDES)
library(LandR)

## April 1st, 2019:
##  growth/mortality processes still in separate module, but will be ported back to Biomass_core soon.
```

```
moduleName <- c("Biomass_core")
spadesModulesDirectory <- ".." # In general, a module code will be controlled at one level above the so
```

Get the module

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

Setup simulation

```
setPaths(inputPath = "inputs",
         cachePath = "cache",
         modulePath = spadesModulesDirectory,
         outputPath = "outputs")

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

## Usage example
modules <- as.list(moduleName)
objects <- list()
paths <- getPaths()

successionTimestep <- 1L

## keep default values for most parameters
## (ommitted from this list)
parameters <- list(
  Biomass_core = list(
    ".plotInitialTime" = times$start
    , "sppEquivCol" = "Boreal"
    , "successionTimestep" = successionTimestep*10
    , ".saveInitialTime" = 1
    , ".useCache" = FALSE
    , ".useParallel" = FALSE
  )
)

mySim <- simInit(times = times, params = parameters, modules = modules, objects = objects, paths = path
mySimOut <- spades(mySim, debug = TRUE)
```

Events

Events are scheduled as follows:

- Module initiation
- Seed dispersal (every 10 time steps)
- Mortality and growth
- Reclassification of age cohorts (every 10 time steps)
- Summary tables of regeneration (summaryRegen), biomass, age, growth and mortality (summaryBGM*)
- Plots of maps (plotMaps) and averages (plotAvgs)
- Save

Links to other modules

Intended to be used with other landscape modules, such as LandMine, FireSense, Biomass_borealDataPrep, and possibly many others.