

Boreal_LBMRDataPrep

Yong Luo & Eliot McIntire

6 October 2017

Overview

This module converts open datasets that are available for all of Canada's forests, into the input requirements for Land-R-Biomass, a forest landscape succession model derived from the Landis-II Biomass Succession Model. This has been partially tested for some parts of the Western Boreal Forest.

Specifically, it takes the Eco* maps (Ecozone, Ecoprovince, Ecoregion) maps of Canada, and species specific biomass maps of Canada (from Beaudoin *et al.* 2014).

Keeping these data preparations outside of the core LBMR module maintains the modularity of LBMR.

Install packages, if necessary

```
if (packageVersion("SpaDES") < "2.0.0")  
  install.packages("SpaDES") # prepInputs with alsoExtract argument  
  
# The most recent version of reproducible package is necessary -- check if reproducible is at least 0.2  
if (packageVersion("reproducible") < "0.2.0.9002")  
  devtools::install_github("PredictiveEcology/reproducible@development", dependencies = FALSE) # uses v  
if (packageVersion("SpaDES.core") < "0.2.0")  
  devtools::install_github("PredictiveEcology/SpaDES.core@development", dependencies = FALSE) # uses ex
```

Load libraries

```
library(magrittr) # for %>% pipe  
library(SpaDES)
```

Set up paths

```
moduleName <- "Boreal_LBMRDataPrep"  
spadesModulesDirectory <- "modules" # where the module will be downloaded  
  
inputPath <- file.path(dirname(spadesModulesDirectory), "inputs") %>% checkPath(create = TRUE)  
outputPath <- file.path(dirname(spadesModulesDirectory), "outputs")  
cachePath = file.path(outputPath, "cache")  
  
setPaths(cachePath = cachePath,  
          modulePath = spadesModulesDirectory,  
          inputPath = inputPath,
```

```

        outputPath = outputPath)
paths <- getPaths()

```

Choose a study area

```

library(raster)
cachePath <- file.path("Boreal_LBMRDataPrep", "cache")
modulePath <- Cache(readline, paste0("Where is the module path? (e.g., ~/module, with no quotes).\n",
                                     "Press Enter to accept the path in getPaths()$modulePath: "),
                    cacheRepo = cachePath)
setPaths(cachePath = cachePath, modulePath = modulePath)

## do you want to hand-draw a map or use defaults?
# - note that large areas will take longer to compute
handDrawMap <- FALSE

if (handDrawMap) {
  dev()
  clearPlot()
  canadaMap <- Cache(getData, 'GADM', country = 'CAN', level = 1, path = "data/",
                    cacheRepo = getPaths()$cachePath, quick = FALSE)
  LIM_SA <- shapefile("../LandscapesInMotion/data/maps/Foothills_study_area.shp")
  Plot(canadaMap, speedup = 5, visualSqueeze = 0.9) # 5 seemed optimal
  Plot(LIM_SA, addTo = "canadaMap", col = "green")

  ## hand-drawn study area
  if(!exists("shpStudyRegionFull")) {
    message("Since there is no object called 'shpStudyRegionFull', please draw a study area with 10 points")
    severalrandompoints <- clickCoordinates(10)
    if(startsWith(attr(severalrandompoints, "tags"), "cache")) message("Taking shpStudyRegionFull from cache")
    shpStudyRegionFull <- SpatialPolygons(list(Polygons(list(Polygon(severalrandompoints$coords)), ID = "shpStudyRegionFull",
                                                         proj4string = crs(canadaMap)))
  }
}
Plot(shpStudyRegionFull, addTo = "canadaMap", col = "red")

times <- list(start = 0, end = 10)
modules <- list("Boreal_LBMRDataPrep")
objects <- if (handDrawMap) list("shpStudyRegionFull" = shpStudyRegionFull,
                                "shpStudySubRegion" = shpStudyRegionFull) else list()

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

```

Run spades

This module is about data preparation, so there is no stochastic elements. The `spades` call will only cause one event to occur (the `init` event)

```
simOut <- spades(mySim)
```

Visualize

The `Plot` function will visualize all known `.quickPlot` type objects, which includes `Raster*` and `SpatialPolygons*` objects. After running this module, these are the outputs, which would likely be used as inputs to LBMR.

```
dev()
clearPlot()

Plot(simOut)
```

Downloads

During the `simInit` call, if the user does not provide alternatives for the expected inputs, the module will download 3 large `.tar` files (~2 GB each) and 1 `.zip` file (~45 MB) from the internet.

Inputs

This module has several input requirements. One is a study area, which should be provided as a `SpatialPolygonsDataFrame`, and named `shpStudyRegionFull`. This should be inside the boundaries of the boreal forest of Canada. When first running the code in this `.Rmd` file, you will be prompted to draw a polygon if none is provided as an input.

Creates Inputs

Most of the inputs will be created automatically, if they are not provided by the user. The automatic creation will work in the boreal forest of Canada. These are zip files and tar files that are available from various Natural Resources Canada web pages. Also, this module gets its Species Traits table from `dcyr/LANDIS-II_IA_generalUseFiles`.

Outputs

This will show the outputs of this module, which can be used directly as the inputs for LBMR:

```
# List all objects
ls(simOut)

# Examine a few tables a visuals
simOut$speciesTable
Plot(simOut$biomassMap)
simOut$shpStudyRegionFull <- spTransform(simOut$shpStudyRegionFull, crs(simOut$biomassMap))
Plot(simOut$shpStudyRegionFull, addTo = "simOut$biomassMap")
```

References