

Perturbations

Maxime Jaunatre

2022-12-08

Testing simulation with perturbations

This document show how to simulate a species with perturbation occuring in time. For now, the package is in experimental state and use the branch *tempest*. For this example we will use the species *Picea abies* as an example.

Only the Julien disturbance model is explained in this case.

```
# Libraries
library(ggplot2)
library(dplyr)

##
## Attachement du package : 'dplyr'
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
## L'objet suivant est masqué depuis 'package:testthat':
##
##     matches
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
library(devtools)

# Loading all functions of the package
devtools::load_all()

## i Loading treeforce
species <- "Picea_abies"
data(list = paste0("fit_", species))
fit <- eval(parse(text=paste0("fit_", species)))
climate <- subset(climate_species, sp == species & N == 2, select = -c(N, sp))
climate <- drop(as.matrix(climate))
ipm_Picea <- make_IPM(
  "Picea_abies", climate, "opt_Picab_clim", fit = fit_Picea_abies,
  mesh = c(m = 700, L = 90, U = get_maxdbh(fit_Picea_abies) * 1.1),
  BA = 0:100, verbose = TRUE
)

## Launching integration loop
## GL integration occur on 32 cells
```

```
## midbin integration occur on 25 cells
```

```
## Integration =>-----
```

```
3% | ETA: 1mIntegration =>-----
```

```
## Time difference of 1.25 mins
```

```
Picea_abies <- species(IPM = ipm_Picea, init_pop = def_initBA(40),  
                      harvest_fun = def_harv)
```

We want to start our simulations with an equilibrium size distribution so we compute here first

```
forest_ipm <- new_forest(species = list(Picea = Picea_abies))
```

```
time <- 3000
```

```
set.seed(42)
```

```
memor <- sim_deter_forest.forest(forest_ipm, tlim = time,  
                                equil_dist = 250, equil_time = time,  
                                verbose = TRUE, correction = "cut") %>%  
  tree_format()
```

```
## apply a IPM cut correction
```

```
## Starting while loop. Maximum t = 3000
```

```
## time 500 | BA diff : 21.00
```

```
## time 1000 | BA diff : 7.35
```

```
## time 1500 | BA diff : 2.35
```

```
## time 2000 | BA diff : 0.83
```

```
## time 2500 | BA diff : 0.32
```

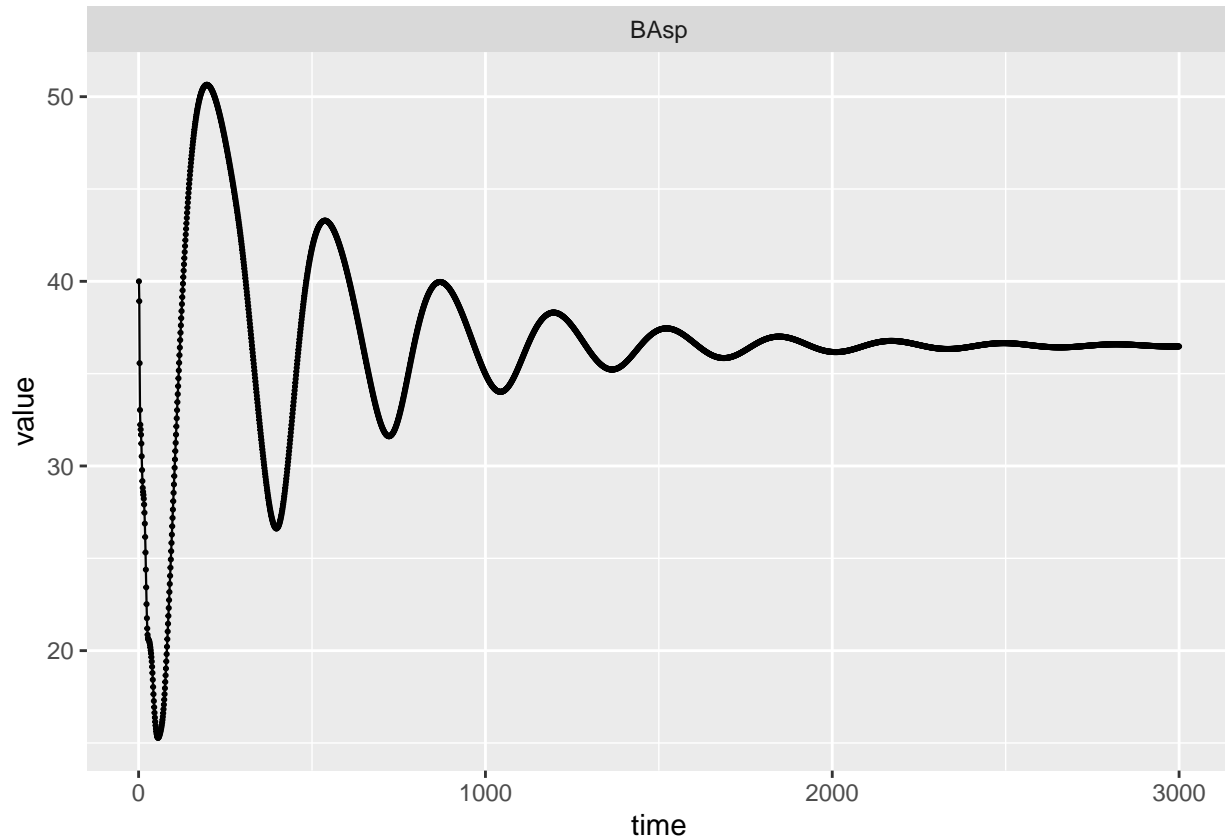
```
## time 3000 | BA diff : 0.12
```

```
## Simulation ended after time 3000
```

```
## BA stabilized at 36.47 with diff of 0.12 at time 3000
```

```
## Time difference of 14.6 secs
```

```
memor %>%  
  filter(var %in% c("BAsp"), ! equil, value != 0) %>%  
  ggplot(aes(x = time, y = value)) +  
  facet_wrap(~ var, scales = "free_y") +  
  geom_line(size = .4) + geom_point(size = .4) +  
  NULL
```



```
equil <- memor %>%
  filter(var == "m", equil) %>% pull(value)
```

Disturbance

Definition of disturbance

We define a disturbance by few parameters used later in the formula.

- I its intensity
- t the duration of the disturbance. Here we need to set it to 5 because it's the time between two census.
- *type* the class of disturbance. This is often a label in "**storm**", "**fire**" and "**biotic**". This is not used in the formula but to filter species parameters fitted.

We can set all of this in a `data.frame` object. We had a last column named **IsSurv**. It's used to tell the simulation if the survival part of the IPM is needed during a disturbance. In this case, the data do not allow differentiation between disturbance mortality and background mortality. Therefore, we need to deactivate baseline mortality so that it is not double counted.

```
ex_disturb <- data.frame(type = "storm", intensity = 0.5, duration = 5, IsSurv = FALSE)
```

Impact on population

The disturbance impact on the population result from parameters computed by Julien. They compose a function that takes the size distribution, quadratic diameter of the species, intensity (I) and duration (t) of the disturbance. A set of parameters was made for each type of disturbance and species.

$$dqm = \sqrt{\frac{\sum_{i=1}^m size_i^2 \times value_i}{\sum_{i=1}^m value_i}} \logratio = \log\left(\frac{size}{dqm}\right) dbh.scaled = dbh.intercept + size \times dbh.slope \logratio.scaled = \logratio.int$$

The parameters are estimated with Bayesian computations. The mean of all estimations are stored inside the package for each combination of species and disturbance type.

```
(coefs <- filter(treeforce::disturb_coef, species == "Picea_abies"))

##   disturbance    species      a0      a1      b      c
## 1      biotic Picea_abies -5.645296  0.0000000 5.831757 -0.079284149
## 2      storm Picea_abies -4.568872 -0.1489534 5.359453  0.003335358
##   dbh.intercept dbh.slope logratio.intercept logratio.slope
## 1      -0.7868063 0.007928421          0.4682191          2.923733
## 2      -0.7197517 0.007261211          0.1725871          2.722508

Picea_abies$disturb_coef <- coefs
```

Linked with this set of parameters, we need to provide a disturbance function to the species we want to simulate. *The species is initiated with an empty function that will throw warnings.*

```
#' Disturbance function
#'
#' @param x population state distribution at time t
#' @param species The species class object of interest to get mesh and RDIcoef
#' values from. RDIcoef is a one line dataframe with RDI coefficient for one
#' species.
#' @param disturb Disturbance parameters. Highly depend on the disturbance
#' impact parameters given to the species.
#' @param ... Not used in this case.
#'
disturb_fun <- function(x, species, disturb = NULL, ...){

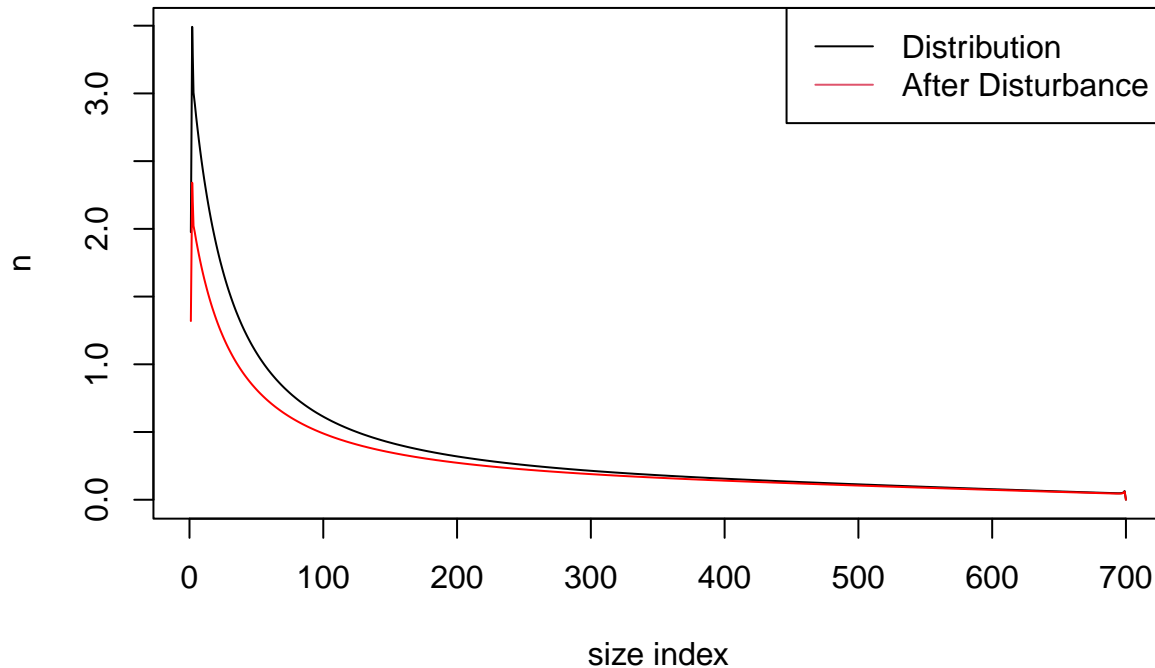
  size <- species$IPM$mesh
  coef <- species$disturb_coef
  if(any(disturb$type %in% coef$disturbance)){
    coef <- subset(coef, disturbance == disturb$type)
  } else {
    stop(sprintf("The species %s miss this disturbance type (%s) parameters",
                 sp_name(species), disturb$type))
  }

  qmd <- treeforce::QMD(size = size, n = x)
  logratio <- log(size / qmd)
  dbh.scaled = coef$dbh.intercept + size * coef$dbh.slope
  logratio.scaled = coef$logratio.intercept + logratio * coef$logratio.slope
  Pkill <- plogis(coef$a0 + coef$a1 * logratio.scaled +
                  coef$b * disturb$intensity^(coef$c * dbh.scaled)) ^ disturb$duration

  return(x* Pkill) # always return the mortality distribution
}

Picea_abies$disturb_fun <- disturb_fun
```

```
plot(equil, type= "l", xlab = "size index", ylab = "n")
lines(1:700, equil - disturb_fun(equil, Picea_abies, ex_disturb), col = "red")
legend("topright", c("Distribution", "After Disturbance"),
      lty = c(1, 1), col = c(1, 2))
```



Simulations

Running a simulation takes the same parameters as usual, with an added data.frame with disturbance along time. *We need to think about a clean way to build this table...*

```
time <- 2500
disturb <- data.frame(dist = FALSE, type = "none", intensity = 0,
                      duration = 1, IsSurv = TRUE, t = 1:time)

disturb[100, ] <- data.frame(dist = TRUE, type = "storm", intensity = 0.5,
                              duration = 5, IsSurv = FALSE, t = 100)
disturb[98:102, ]
```

```
##      dist  type intensity duration IsSurv  t
## 98 FALSE none      0.0         1  TRUE 98
## 99 FALSE none      0.0         1  TRUE 99
## 100 TRUE storm    0.5         5 FALSE 100
## 101 FALSE none      0.0         1  TRUE 101
## 102 FALSE none      0.0         1  TRUE 102
```

```
load_all()
```

```
## i Loading treeforce
```

```
Picea_abies$init_pop <- def_init_k(equil * 0.03)
forest_ipm <- new_forest(species = list(Picea = Picea_abies))
set.seed(42)
memor <- sim_deter_forest.forest(forest_ipm, tlim = time,
                                equil_dist = time, equil_time = time,
```

```

                                disturbance = disturb,
                                verbose = TRUE, correction = "cut") %>%

tree_format()

```

```

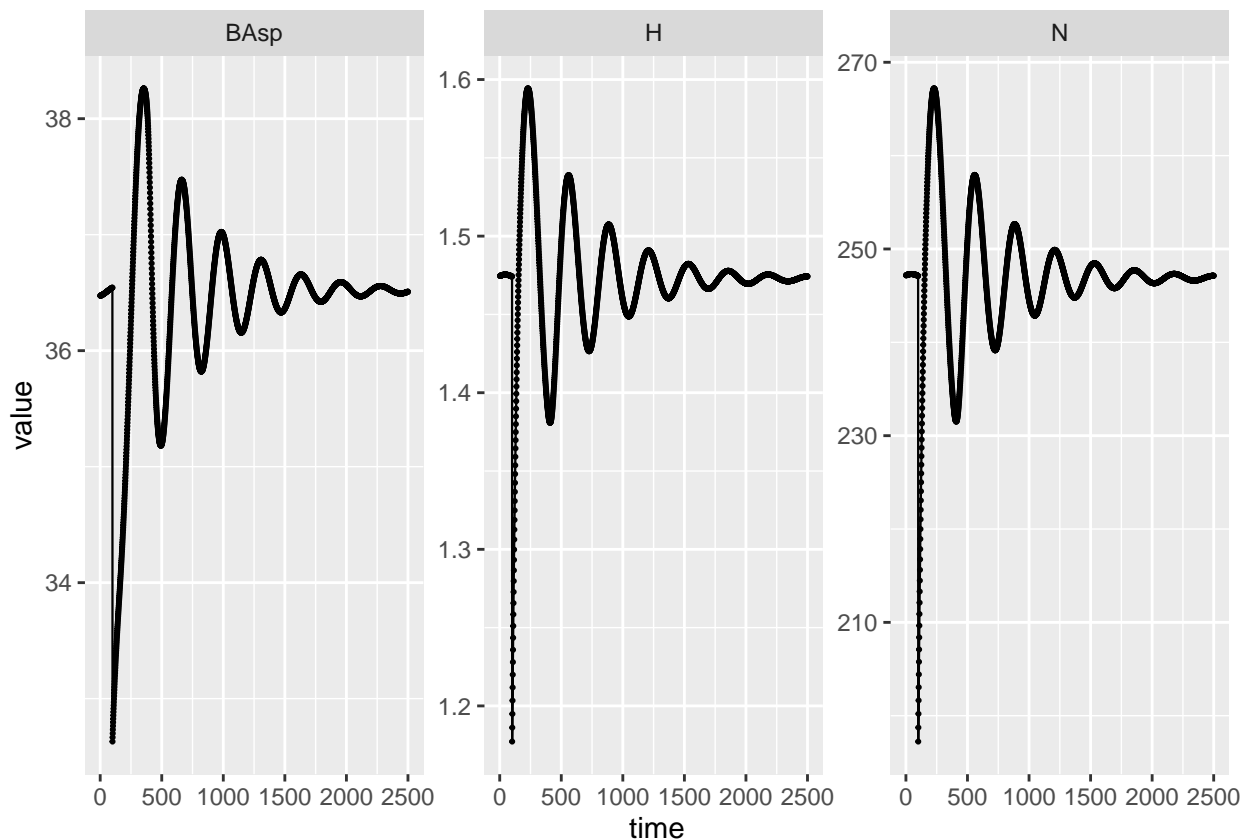
## apply a IPM cut correction
## Starting while loop. Maximum t = 2500
## time 100 | Disturbance : storm I = 0.50
## time 500 | BA diff : 5.63
## time 1000 | BA diff : 5.63
## time 1500 | BA diff : 5.63
## time 2000 | BA diff : 5.63
## time 2500 | BA diff : 5.63
## Simulation ended after time 2500
## BA stabilized at 36.51 with diff of 5.63 at time 2500
## Time difference of 13.3 secs

```

```

memor %>%
  filter(var %in% c("BAsp", "H", "N"), ! equil, value != 0) %>%
  ggplot(aes(x = time, y = value)) +
  facet_wrap(~ var, scales = "free_y") +
  geom_line(size = .4) + geom_point(size = .4) +
  NULL

```



Targets

- Multiple species simulations