



{matreex} R package

Maxime Jaunatre

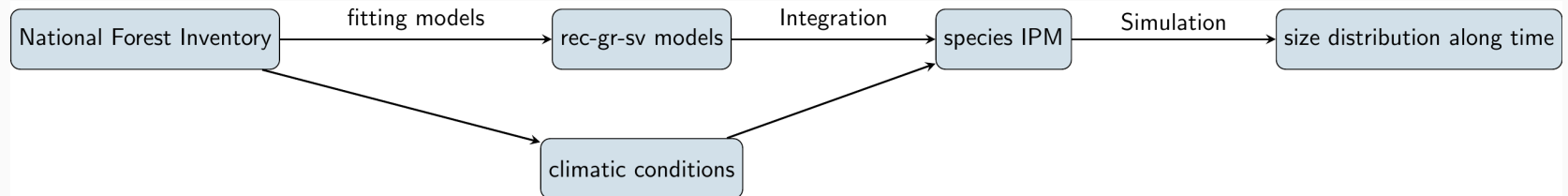
UR LESSEM | INRAE

2023-02-22 (updated: 2023-02-21)

Objectif : Créer un package



Partir des codes de Georges et Arnaud

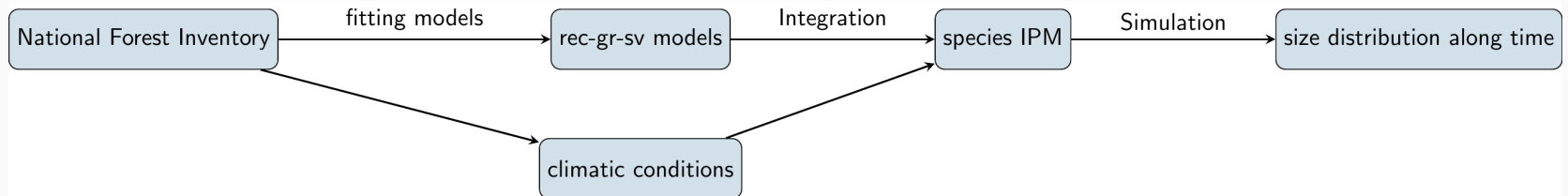


- Code permettant des simulations jusqu'à l'équilibre monospécifique pour un climat donné.
- Code publié avec l'article : *Demographic performance of European tree species at their hot and cold climatic edges* sous la forme d'un pipeline `{targets}`

Objectif : Créer un package



Partir des codes de Georges et Arnaud



- Code permettant des simulations jusqu'à l'équilibre monospécifique pour un climat donné.
- Code publié avec l'article : *Demographic performance of European tree species at their hot and cold climatic edges* sous la forme d'un pipeline `{targets}`

Deux besoins :

Mettre en format package pour faciliter l'utilisation du code

Ajouter des fonctionnalités dans le cadre du projet **FUNPOTENTIAL**

Contraintes :

- Besoin de faire beaucoup de simulations, différents scénariis ou climats.
- Une espèce sous format IPM est relativement lourde et longue à obtenir (2-3min)

Bonus :

- Faciliter l'usage pour les utilisateurs avec des fonctions communes

Choix d'utiliser plusieurs classes **S3**

Les fonctions sont simplifiées et les classes contiennent une structure définie. Mais tout peut être étendu plus tard par d'autres modules.

Fitted models



```
library(matreex) # load pkg  
matreex::fit_species[1:3] # list fitted species list
```

```
## [1] "Abies_alba"           "Acer_campestre"       "Acer_pseudoplatanus"
```

```
data("fit_Picea_abies") # load a single species model
```

Fitted models



```
library(matreex) # load pkg
matreex::fit_species[1:3] # list fitted species list
```

```
## [1] "Abies_alba"          "Acer_campestre"      "Acer_pseudoplatanus"
```

```
data("fit_Picea_abies") # load a single species model
```

```
names(fit_Picea_abies)
```

```
## [1] "sv"    "gr"    "rec"   "info"
```

```
fit_Picea_abies$sv
```

```
## $params_m
```

##	BATOTcomp	BATOTcomp:sgdd	BATOTcomp:wai	intercept	logsize
##	3.614605e-02	-3.333514e-06	-1.002760e-02	-2.104071e+01	3.298095e+00
##	logsize:sgdd	logsize:wai	sgdd	sgdd2	size
##	-2.926404e-03	-1.600203e+00	1.241456e-02	5.492081e-07	-4.927732e-03
##	size:sgdd	size:wai	wai	wai2	
##	5.128908e-06	2.955988e-03	7.758610e+00	2.612837e-01	

```
##
```

```
## $family
```

```
##
```

Integration

Un IPM est définis pour un climat et un niveau de compétition donné.

```
data("climate_species")
climate ← subset(climate_species, N = 2 & sp = "Picea_abies", select = c(1:6, 9
climate
```

```
##          sgdd          wai          sgddb          waib          wai2    sgdd2 N
## 62 1444.667 0.4519387 0.0006922012 0.6887343 0.2042486 2087062 2
```

```
Picea_ipm ← make_IPM(
  species = "Picea_abies",
  climate = climate,
  fit = fit_Picea_abies,
  clim_lab = "optimum clim",
  mesh = c(m = 700, L = 90, U = 1095 * 1.1),
  BA = 0:120
)
```

Summary

```
summary(Picea_ipm)
```

```
## IPM object for species Picea_abies at climate 'optimum clim'
```

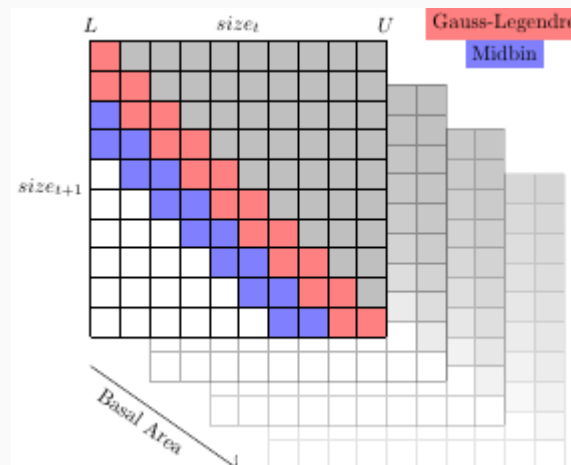
```
##
```

```
## Integation was done on a mesh from 90.80 to 1203.70 with 700 cells for BA between 0
```

```
## Gauss-Legendre was used on 32 cells with 3 x 140 levels (size at  $t * t+1$ )
```

```
## Midbin was used on 25 cells with 5 levels.
```

```
## The correction was constant, the IPM does contain survival and the recruitment delay
```



Species and Forest



Create a species

```
Picea_sp ← species(IPM = Picea_ipm,  
  init_pop = def_init,  
  harvest_fun = def_harv,  
  disturb_fun = def_disturb,  
  harv_lim = c(dth = 175, dha = 575, hmax = 1),  
  rdi_coef = NULL,  
  disturb_coef = NULL  
)
```

Species and Forest

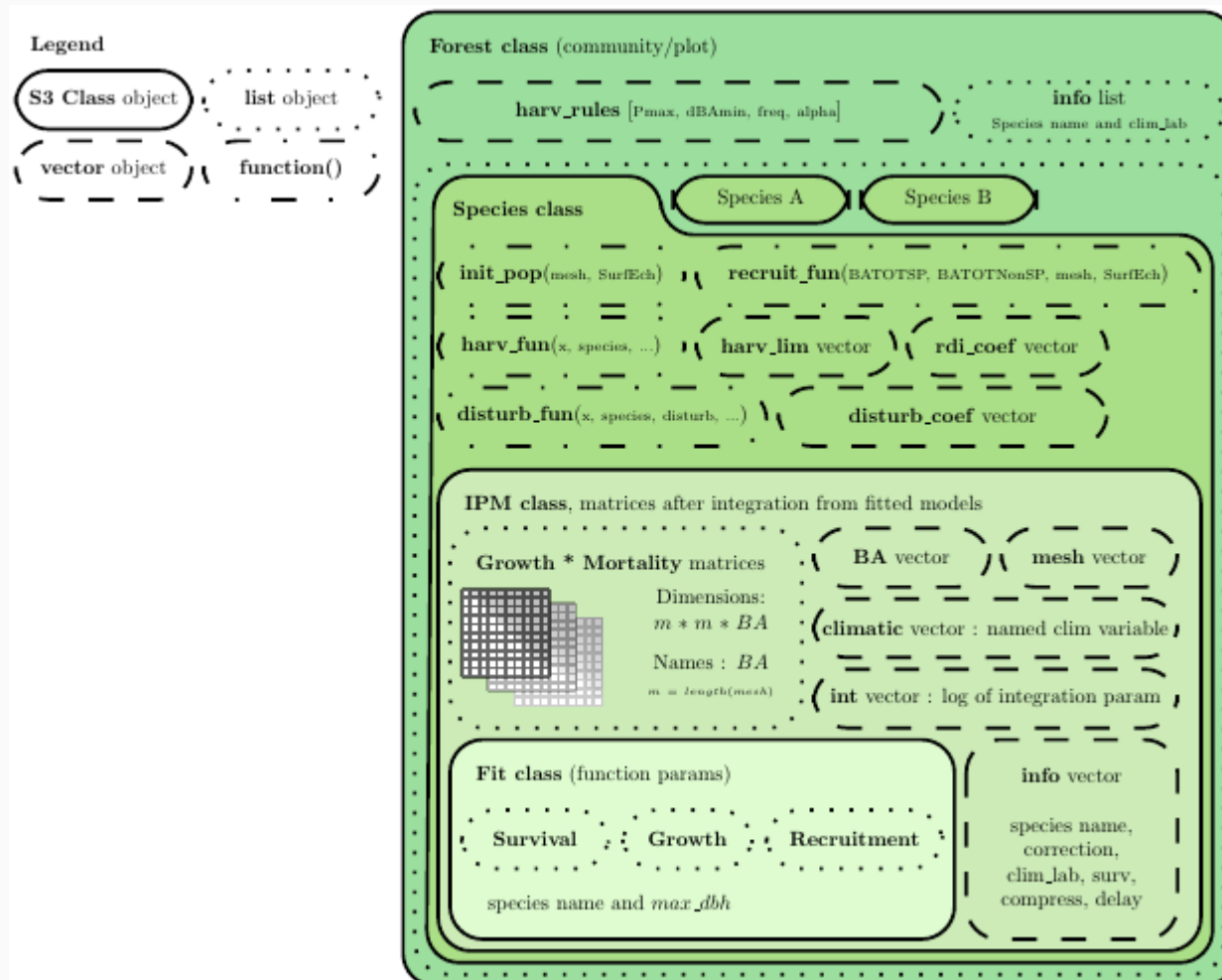


Create a species

```
Picea_sp ← species(IPM = Picea_ipm,  
  init_pop = def_init,  
  harvest_fun = def_harv,  
  disturb_fun = def_disturb,  
  harv_lim = c(dth = 175, dha = 575, hmax = 1),  
  rdi_coef = NULL,  
  disturb_coef = NULL  
)
```

Create a forest

```
Picea_for ← forest(species = list(Picea = Picea_sp),  
  harv_rules = c(Pmax = 0.25, dBamin = 3,  
    freq = 1, alpha = 1))
```



Run simulations



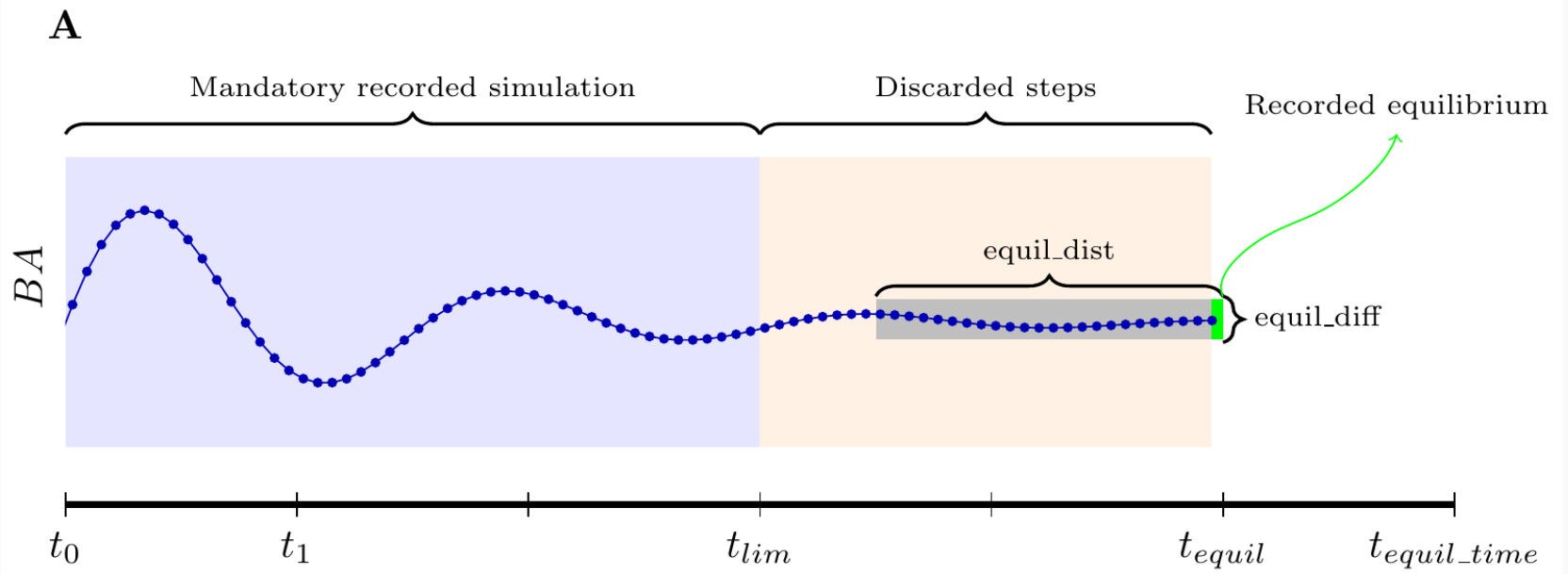
```
set.seed(42) # The seed is here for initial population random functions.
Picea_sim <- sim_deter_forest(
  Picea_for,
  tlim = 200,
  equil_time = 300, equil_dist = 50, equil_diff = 1,
  SurfEch = 0.03,
  verbose = TRUE
)
```

```
## Starting while loop. Maximum t = 300
```

```
## Simulation ended after time 300
```

```
## BA stabilized at 38.96 with diff of 6.61 at time 300
```

```
## Time difference of 1.34 secs
```



Results



Format de sortie de type long prévu pour les plots avec `{ggplot2}` et des filtres `{dplyr}`

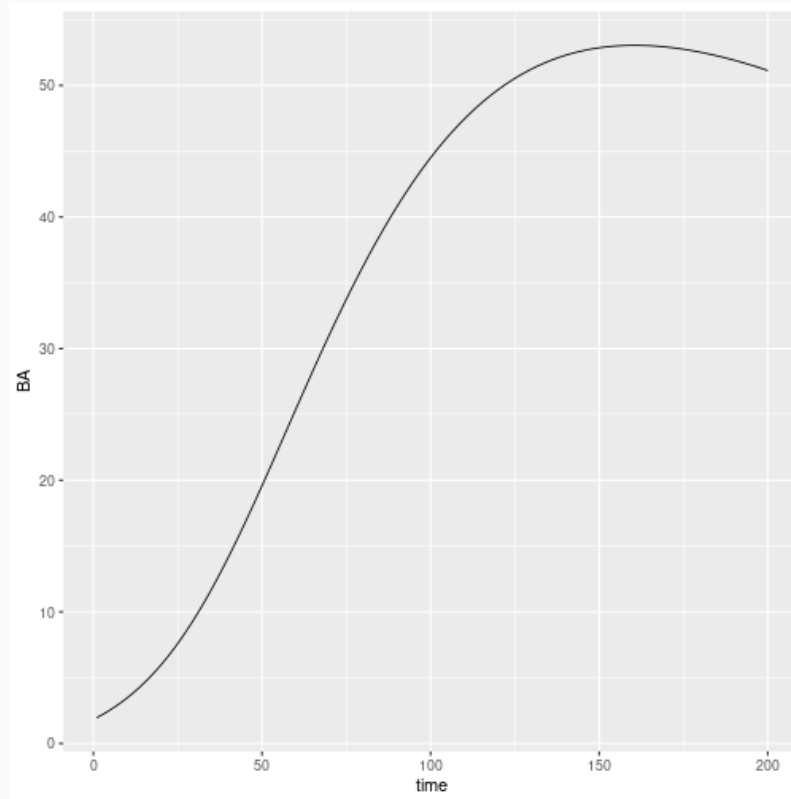
```
Picea_sim %>%  
  dplyr::filter(var = "BAsp", ! equil) %>% head()
```

```
## # A tibble: 6 × 7  
##   species    var    time  mesh  size equil value  
##   <chr>      <chr> <dbl> <dbl> <dbl> <lgl> <dbl>  
## 1 Picea_abies BAsp     1    NA    NA FALSE  1.96  
## 2 Picea_abies BAsp     2    NA    NA FALSE  2.10  
## 3 Picea_abies BAsp     3    NA    NA FALSE  2.24  
## 4 Picea_abies BAsp     4    NA    NA FALSE  2.39  
## 5 Picea_abies BAsp     5    NA    NA FALSE  2.55  
## 6 Picea_abies BAsp     6    NA    NA FALSE  2.72
```

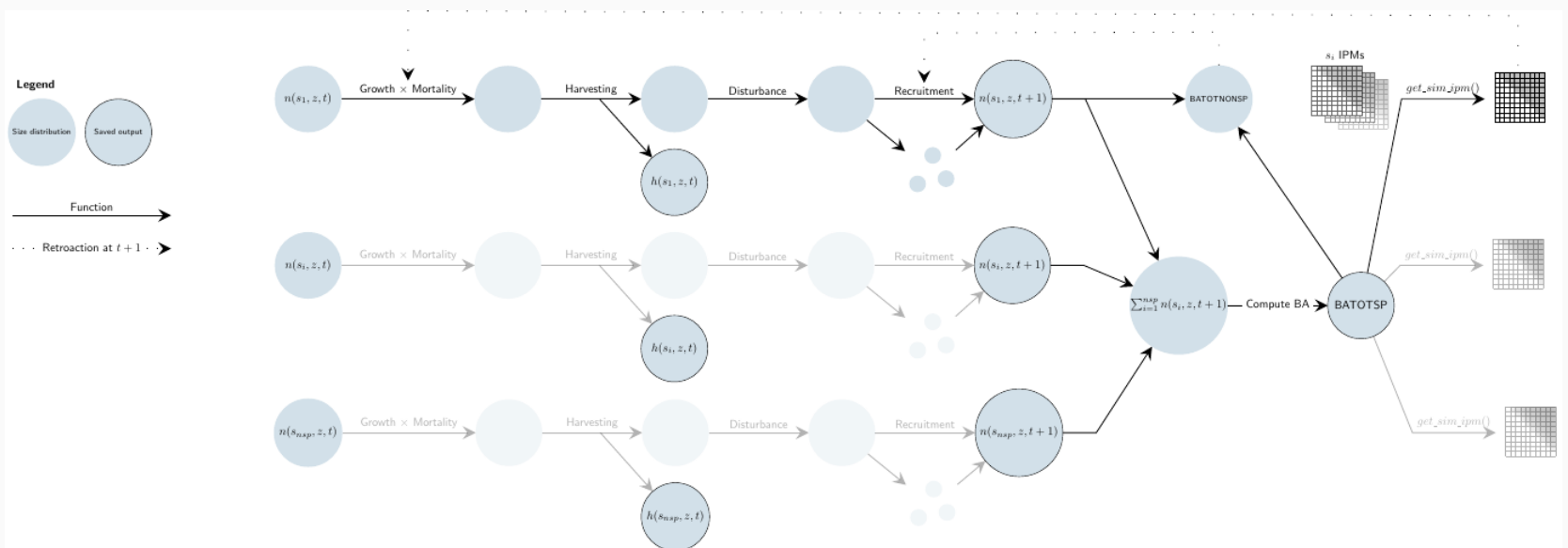
Plots



```
library(ggplot2)
Picea_sim %>% dplyr::filter(var = "BAsp", ! equil) %>%
  ggplot(aes(x = time, y = value)) + geom_line(linewidth = .4) + ylab("BA")
```



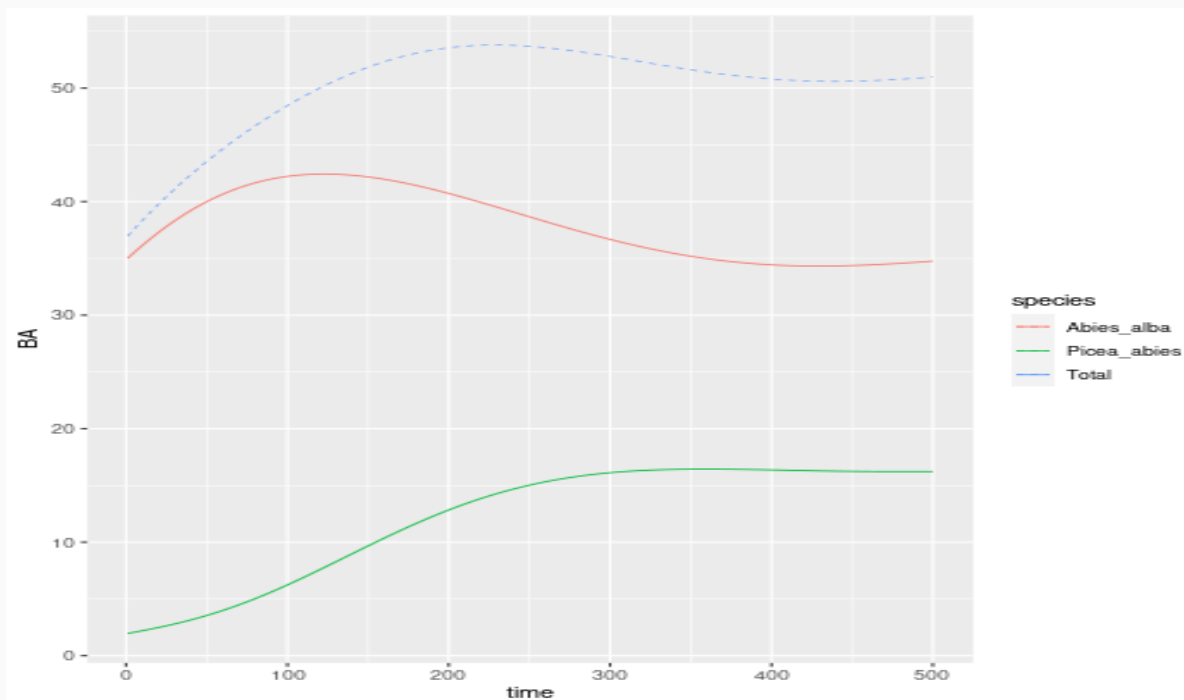
Loop algorithm

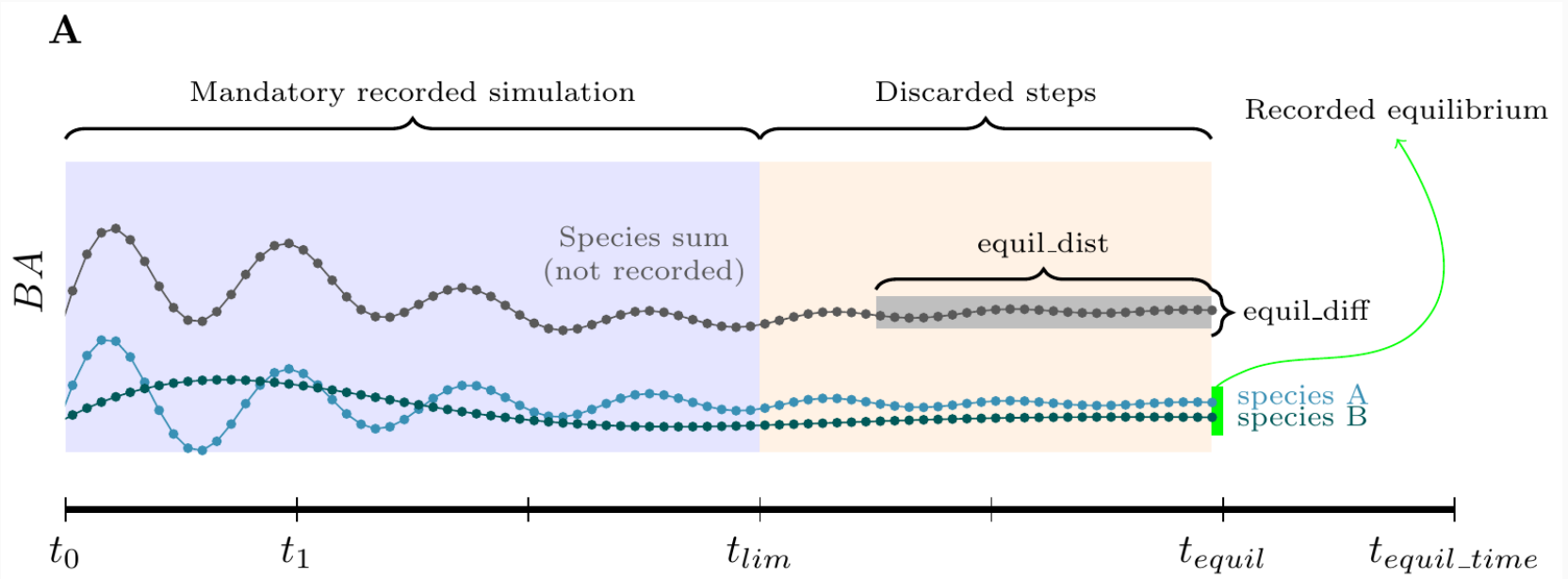


Multiple species




```
Picea_Abies_for ← forest(species = list(Picea = Picea_sp, Abies = Abies_sp))
set.seed(42)
Picea_Abies_sim ← sim_deter_forest(
  Picea_Abies_for,
  tlim = 500,
  equil_time = 500, equil_dist = 50,
  SurfEch = 0.03
)
```





Merci! Des questions ?

 [gowachin](#)  [gowachin](#)

Slides created via the R package [xaringan](#).