#2: Decreasing calculation time

Nadege Belouard*

2024-08-14

Contents

Aim and setup

In case of extremely large species occurrence datasets, it may take a long time to run the analyses. Any number of sectors will provide the accurate results. However, computational time may be decreased by increasing the number of sectors considered. The higher the number of sectors, the larger the invasion radius at which points are compared by pairs in `find_thresholds`, so the fewer distances need to be calculated. However, the lower the number of sectors, the better pre-identification of spatial discontinuities and the more pruned the list of potential jumps, so the faster `find_jumps`. The lowest computational time is therefore obtained by a trade-off between dataset size, invasion radius, and number of sectors.

We demonstrate the effect of the number of sectors on computational time on the SLF dataset.

```
library(magrittr)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'

## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag

## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(jumpID)
```

Load the grid data created in the first vignette

*iEco lab at Temple University, Ecobio lab at the University of Rennes, nadege.belouard@gmail.com

```
grid_data <- read.csv(file.path(here::here(), "exported-data", "grid_data.csv"))
```

Compare calculation times

Run the jumpID functions successively for 16, 40, and 80 sectors and compare computation times.

```
sectors = c(16, 40, 80)

optim <- data.frame(s = NULL,
                       Time_sectors = NULL,
                       Time_thresholds = NULL,
                       potJumps = NULL,
                       Time_jumps = NULL,
                       Jumps = NULL,
                       Time_secDiff = NULL)

for (s in sectors){
  print(paste0("Sectors: ", s))

  #1 Attribute sectors
  start.time.attribute_sectors <- Sys.time()
  grid_data_sectors <- jumpID::attribute_sectors(dataset = grid_data,
                                                     nb_sectors = s,
                                                     centroid = c(-75.675340, 40.415240))
  end.time.attribute_sectors <- Sys.time()
  time.taken.attribute_sectors <- end.time.attribute_sectors - start.time.attribute_sectors



  #2 Find thresholds
  start.time.find_thresholds <- Sys.time()
  Results_thresholds <- jumpID::find_thresholds(dataset = grid_data_sectors,
                                                    gap_size = 15,
                                                    negatives = T)
  preDist <- Results_thresholds$preDist
  potJumps <- Results_thresholds$potJumps
  end.time.find_thresholds <- Sys.time()
  time.taken.find_thresholds <- end.time.find_thresholds - start.time.find_thresholds



  #3 Find jumps
  start.time.find_jumps <- Sys.time()
  Results_jumps <- jumpID::find_jumps(grid_data = grid_data,
                                        potJumps = potJumps,
                                        gap_size = 15)
  Jumps <- Results_jumps$Jumps
  diffusers <- Results_jumps$diffusers
  potDiffusion <- Results_jumps$potDiffusion
  end.time.find_jumps <- Sys.time()
  time.taken.find_jumps  <- end.time.find_jumps - start.time.find_jumps
```

```r
  #4 Find sec diff
  start.time.find_secDiff <- Sys.time()
  Results_secDiff <- jumpID::find_secDiff(potDiffusion = potDiffusion,
                                          Jumps = Jumps,
                                          diffusers = diffusers,
                                          Dist = preDist,
                                          gap_size = 15)
  end.time.find_secDiff <- Sys.time()
  time.taken.find_secDiff  <- end.time.find_secDiff - start.time.find_secDiff



  result <- data.frame(s = s,
                       Time_sectors = time.taken.attribute_sectors,
                       Time_thresholds = time.taken.find_thresholds,
                       potJumps = dim(potJumps)[1],
                       Time_jumps = time.taken.find_jumps,
                       Jumps = dim(Jumps)[1],
                       Time_secDiff = time.taken.find_secDiff,
                       Total_time = time.taken.attribute_sectors + time.taken.find_thresholds +
                         time.taken.find_jumps + time.taken.find_secDiff)
  optim <- rbind(optim, result)
}
```

```
## [1] "Sectors: 16"
## 2024-08-14 17:22:21.248011 Start sector attribution... Sector attribution completed.
## 2024-08-14 17:22:21.298873 Start finding thresholds... Sector 1/16...  2/16...  3/16...  4/16...  5/
## Threshold analysis done. 4243 potential jumps were found.
## 2024-08-14 17:28:58.736689 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-14 17:29:21.456446 Start finding secondary diffusion... Year 2017 ...Year 2018 ...Year 2019
## [1] "Sectors: 40"
## 2024-08-14 17:30:29.635982 Start sector attribution... Sector attribution completed.
## 2024-08-14 17:30:29.656277 Start finding thresholds... Sector 1/40...  2/40...  3/40...  4/40...  5/
## Threshold analysis done. 3747 potential jumps were found.
## 2024-08-14 17:32:33.720621 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-14 17:32:54.405791 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018
## [1] "Sectors: 80"
## 2024-08-14 17:33:42.11299 Start sector attribution... Sector attribution completed.
## 2024-08-14 17:33:42.140418 Start finding thresholds... Sector 1/80...  2/80...  3/80...  4/80...  5/
## Warning: no negative survey in the gap identified in sector 23 and year 2022 after 106 km. The spati
##  24/80...  25/80...  26/80...  27/80...  28/80...  29/80...  30/80...  31/80...  32/80...  33/80...
## Warning: no negative survey in the gap identified in sector 33 and year 2020 after 113 km. The spati
##  34/80...  35/80...  36/80...  37/80...  38/80...  39/80...  40/80...  41/80...  42/80...  43/80...
## Threshold analysis done. 5034 potential jumps were found.
## 2024-08-14 17:34:02.213285 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-14 17:34:36.829112 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018
```

```r
optim
```

```
##    s    Time_sectors Time_thresholds potJumps    Time_jumps Jumps
## 1 16 0.05011010 secs  6.6239708 mins     4243 22.71957 secs   387
```

```
## 2 40 0.02022815 secs  2.0677376 mins     3747 20.68506 secs   387
## 3 80 0.02735901 secs  0.3345464 mins     5034 34.61572 secs   387
##     Time_secDiff    Total_time
## 1 1.1363157 mins 488.3869 secs
## 2 0.7950928 mins 192.4751 secs
## 3 1.2803433 mins 131.5365 secs
```

For this dataset, all computational times are decreased by dividing space into 40 sectors instead of 16. Data is not dense enough for dividing space into 80 sectors, as indicated by multiple warning messages from `find_threshold`.

– end of vignette –