#2: Decreasing calculation time

Nadege Belouard*

2024-08-20

Contents

Aim and setup **1**

Compare calculation times **2**

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 <- Sys.time()
  grid_data_sectors <- jumpID::attribute_sectors(dataset = grid_data,
                                                   nb_sectors = s,
                                                   centroid = c(-75.675340, 40.415240))



  #2 Find thresholds
  Results_thresholds <- jumpID::find_thresholds(dataset = grid_data_sectors,
                                                  gap_size = 15,
                                                  negatives = T)
  preDist <- Results_thresholds$preDist
  potJumps <- Results_thresholds$potJumps



  #3 Find jumps
  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



  #4 Find sec diff
  Results_secDiff <- jumpID::find_secDiff(potDiffusion = potDiffusion,
                                            Jumps = Jumps,
                                            diffusers = diffusers,
                                            Dist = preDist,
                                            gap_size = 15)
```

```r
  end.time <- Sys.time()
  time.taken  <- end.time - start.time



  result <- data.frame(s = s,
                       potJumps = dim(potJumps)[1],
                       Jumps = dim(Jumps)[1],
                       Total_time = time.taken)
  optim <- rbind(optim, result)
}
```

```
## [1] "Sectors: 16"
## 2024-08-20 21:36:26.050098 Start sector attribution... Sector attribution completed.
## 2024-08-20 21:36:26.097713 Start finding thresholds... Sector 1/16...  2/16...  3/16...  4/16...  5/1
## Threshold analysis done. 4243 potential jumps were found.
## 2024-08-20 21:42:47.133793 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-20 21:43:10.02841 Start finding secondary diffusion... Year 2017 ...Year 2018 ...Year 2019 .
## [1] "Sectors: 40"
## 2024-08-20 21:44:17.490457 Start sector attribution... Sector attribution completed.
## 2024-08-20 21:44:17.509916 Start finding thresholds... Sector 1/40...  2/40...  3/40...  4/40...  5/4
## Threshold analysis done. 3747 potential jumps were found.
## 2024-08-20 21:46:11.271283 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-20 21:46:31.751719 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018
## [1] "Sectors: 80"
## 2024-08-20 21:47:19.156188 Start sector attribution... Sector attribution completed.
## 2024-08-20 21:47:19.180472 Start finding thresholds... Sector 1/80...  2/80...  3/80...  4/80...  5/8
## Warning: no negative survey in the gap identified in sector 23 and year 2022 after 106 km. The spatia
##  24/80...  25/80...  26/80...  27/80...  28/80...  29/80...  30/80...  31/80...  32/80...  33/80... W
## Warning: no negative survey in the gap identified in sector 33 and year 2020 after 113 km. The spatia
##  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.
## 2024-08-20 21:47:38.991352 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-20 21:48:14.125123 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018
```

```r
optim
```

```
##    s potJumps Jumps    Total_time
## 1 16     4243   387 7.857331 mins
## 2 40     3747   387 3.027747 mins
## 3 80     5034   387 2.187281 mins
```

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 –