#2: Decreasing calculation time

Nadege Belouard*

2024-08-13

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

```r
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.

```r
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
```

```
  #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"
## 2024-08-13 11:40:50.31325 Start sector attribution... Sector attribution completed.
## 2024-08-13 11:40:50.4182 Start finding thresholds... Sector 1/16...  2/16...  3/16...  4/16...  5/16
## Threshold analysis done. 4243 potential jumps were found.
## 2024-08-13 11:48:32.449812 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-13 11:55:15.170252 Start finding secondary diffusion... Year 2017 ...Year 2018 ...Year 2019
## [1] "Sectors: 40"
## 2024-08-13 11:56:22.828935 Start sector attribution... Sector attribution completed.
## 2024-08-13 11:56:22.849666 Start finding thresholds... Sector 1/40...  2/40...  3/40...  4/40...  5/4
## Threshold analysis done. 3887 potential jumps were found.
## 2024-08-13 12:01:40.195414 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-13 12:07:17.884 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018 ...
## [1] "Sectors: 80"
## 2024-08-13 12:08:06.107858 Start sector attribution... Sector attribution completed.
## 2024-08-13 12:08:06.128443 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. 5096 potential jumps were found.
## 2024-08-13 12:12:48.003018 Start finding jumps... Year 2014 ... Year 2015 ... Year 2016 ... Year 201
## 2024-08-13 12:19:16.959926 Start finding secondary diffusion... Year 2016 ...Year 2017 ...Year 2018
```

```
optim
```

```
##    s    Time_sectors Time_thresholds potJumps    Time_jumps Jumps  Time_secDiff
## 1 16 0.07503319 secs   7.701029 mins     4243 6.712007 mins   387 1.127631 mins
```

```
## 2 40 0.02066112 secs   5.289094 mins     3887 5.628141 mins   387 0.803703 mins
## 3 80 0.02051687 secs   4.697907 mins     5096 6.482613 mins   387 1.295748 mins
##      Total_time
## 1 932.5150 secs
## 2 703.2769 secs
## 3 748.5966 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 –