Urban ecological corridors in Ouro Preto: removing STs from corridors

Tulaci Bhakti, Bernardo Niebuhr, Joao Carlos Pena

May 2019

```
# Clean everything before beginning

rm(list = ls())

# Data folder
corrdir <- 'simulated_corridors/'
stdir <- 'input/Raster_finais/'
mapdir <- 'input/Shapefiles/'

# Output folder
outdir <- 'output/'
```

Loading data

Load corridor rasters, vetors, and information already imported into R.

```
load(paste0(corrdir, 'corridors_loaded.RData'))

# species
sp.short <- c('aleuco', 'ccaudata', 'pleuco', 'sscans', 'xfuscus')
sp <- c('A. leucophthalmus', 'C. caudata', 'P. leucoptera', 'S. scansor', 'X. fuscus')
```

Remove ST from maps

Remove ST from RSFI maps

First, we check that all maps have the same extent We'll standardize the extent according to the first species of no zone scenario

```
# no zoning
corridors_nozone_nost <- list()

for(i in 1:length(corridors_no_zone)) {
  corridors_nozone_nost[[i]] <- raster::crop(raster::extend(corridors_no_zone[[i]],
                                                            corridors_no_zone[[1]]),
                                             corridors_no_zone[[1]])
}
corridors_nozone_nost

# just to check that nothing changed in the map, only the extent
#raster::writeRaster(corridors_zone[[4]], 'raster_antigo.tif', overwrite = TRUE)
#raster::writeRaster(corridors_zone_nost[[4]], 'raster_novo.tif')

# zoning
```

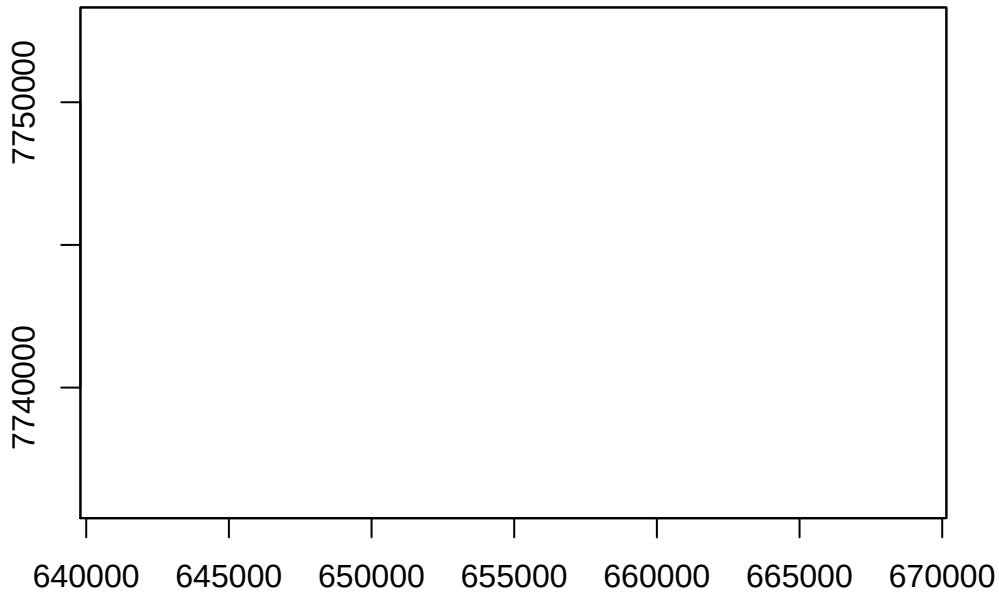
```
corridors_zone_nost <- list()

for(i in 1:length(corridors_zone)) {
  corridors_zone_nost[[i]] <- raster::crop(raster::extend(corridors_zone[[i]],
                                                          corridors_no_zone[[1]]),
                                           corridors_no_zone[[1]])
}
corridors_nozone_nost
```

Now we'll transform do the same for the ST map.

```
ST.map <- raster::crop(raster::extend(ST.map, corridors_no_zone[[1]]), corridors_no_zone[[1]])
ST.map[1:10]
```

Finally we'll remove corridors information from ST patches

```
# no zoning
for(i in 1:length(corridors_nozone_nost)) {
  # print(i)
  corridors_nozone_nost[[i]][] <- ifelse(!is.na(ST.map[]), NA, corridors_nozone_nost[[i]][])
}
corridors_nozone_nost

plot(corridors_nozone_nost[[1]])
```



Check if values have inceased. On the left (right) we have the range (minimum, maximum) of values before (after) removing STs from corridor RSFI maps.

```
for(i in 1:length(corridors_no_zone)) {
  rb <- raster::brick(corridors_no_zone[[i]], corridors_nozone_nost[[i]])
  names(rb) <- c('with.ST', 'without.ST')
  rb %>%
    cellStats(stat = 'range') %>%
    print
}
```

```
##      with.ST without.ST
```

```
## [1,]       1       -Inf
## [2,]     137        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]      14        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]      38        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]     140        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]      17        Inf
```

```
# zoning
for(i in 1:length(corridors_zone_nost)) {
  # print(i)
  corridors_zone_nost[[i]][] <- ifelse(!is.na(ST.map[]), NA, corridors_zone_nost[[i]][])
}
corridors_zone_nost
```

Check if values have inceased. On the left (right) we have the range (minimum, maximum) of values before (after) removing STs from corridor RSFI maps.

```
for(i in 1:length(corridors_zone)) {
  rb <- raster::crop(raster::extend(corridors_zone[[i]], corridors_no_zone[[1]]),
                corridors_no_zone[[1]]) %>%
    raster::brick(corridors_zone_nost[[i]])
  names(rb) <- c('with.ST', 'without.ST')
  rb %>%
    cellStats(stat = 'range') %>%
    print
}
```

```
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]     161        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]      39        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]     101        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]     170        Inf
##      with.ST without.ST
## [1,]       1       -Inf
## [2,]      38        Inf
```

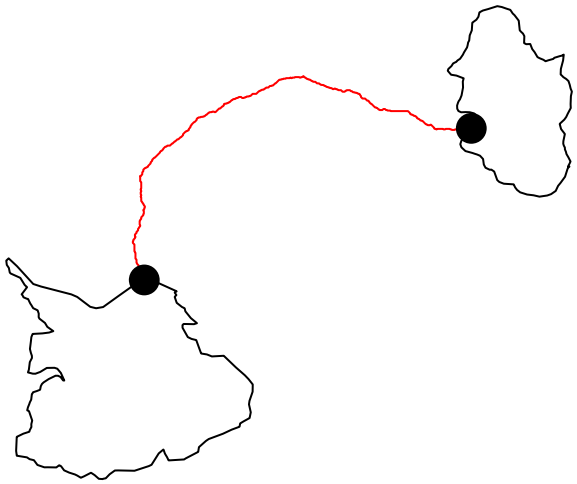
Write folder

```r
# Create new folder, if it does not exist
newdir <- paste0(corrdir, 'corredores_sem_st/')
dir.create(newdir, showWarnings = FALSE)

for(i in 1:length(corridors_no_zone)) {
  # no zoning
  corridors_nozone_nost[[i]] %>%
  raster::writeRaster(paste0(newdir, 'corridors_RSFI_nozoning_', sp.short[i], '.tif'),
                      overwrite = TRUE)

  # zoning
  corridors_zone_nost[[i]] %>%
    raster::writeRaster(paste0(newdir, 'corridors_RSFI_zoning_', sp.short[i], '.tif'),
                        overwrite = TRUE)
}
```

Remove STs from vector corridors

```r
# load ST vector
ST.pol <- rgdal::readOGR(paste0(mapdir, 'sources_targets.shp'))

# one example
# select ID polygons
(ids <- corr.shp.sts.sem.zona[[1]][[5]])
(st.ids <- ST.pol[ST.pol$gridcode == ids[1] | ST.pol$gridcode == ids[2],])

# remove area within STs
corr.cut <- rgeos::gDifference(corr.shp.sem.zona[[1]][[5]], st.ids)

# get initial and end coordinates
coords <- do.call('rbind', lapply(coordinates(corr.cut)[[1]], function(x) x[c(1, nrow(x)),])) %>%
  SpatialPoints(proj4string = crs(st.ids))

# initial coord
pos.init <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[1],], byid = TRUE) %>%
  apply(MARGIN = 2, min) %>%
  which.min()
init.pt <- coords[pos.init]

# final coord
pos.fin <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[2],], byid = TRUE) %>%
  apply(MARGIN = 2, min) %>%
  which.min()
fin.pt <- coords[pos.fin]

# Euclidean distance
(ED <- raster::pointDistance(init.pt, fin.pt) %>% as.numeric)

# calculate total corridor length
(Tot.len <- rgeos::gLength(corr.cut))

# Plot to check
```

```r
plot(st.ids)
plot(corr.cut, add = T, col = 2)
plot(init.pt, add = 2, cex = 2, pch = 19)
plot(fin.pt, add = 2, cex = 2, pch = 19)
```



```r
# Loop for all species

# No zoning
sp
(tab.sem.zona <- data.frame(sp = NA, scenario = NA, source = NA, target = NA, sim = NA,
                  tot.dist = NA, euc.dist = NA, cost = NA)[-1,])
scen <- 'Land cover'

# for each species
for(i in 1:length(corr.shp.sem.zona)) {

  # for each simulation
  for(j in 1:length(corr.shp.sem.zona[[i]])) {

    # select ID polygons
    (ids <- corr.shp.sts.sem.zona[[i]][[j]])
    (st.ids <- ST.pol[ST.pol$gridcode == ids[1] | ST.pol$gridcode == ids[2],])

    # remove area within STs
    corr.cut <- rgeos::gDifference(corr.shp.sem.zona[[i]][[j]], st.ids)

    # get initial and end coordinates
    coords <- do.call('rbind', lapply(coordinates(corr.cut)[[1]], function(x) x[c(1, nrow(x)),])) %>%
      SpatialPoints(proj4string = crs(st.ids))

    # initial coord
    pos.init <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[1],], byid = TRUE) %>%
      apply(MARGIN = 2, min) %>%
      which.min()
    init.pt <- coords[pos.init]

    # final coord
    pos.fin <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[2],], byid = TRUE) %>%
```

```
      apply(MARGIN = 2, min) %>%
      which.min()
    fin.pt <- coords[pos.fin]

    # Euclidean distance
    (ED <- raster::pointDistance(init.pt, fin.pt) %>% as.numeric)

    # calculate total corridor length
    (Tot.len <- rgeos::gLength(corr.cut))

    # Plot to check
    # plot(st.ids)
    # plot(corr.cut, add = T, col = 2)
    # plot(init.pt, add = 2, cex = 2, pch = 19)
    # plot(fin.pt, add = 2, cex = 2, pch = 19)

    # attach info
    tab.sem.zona <- rbind(tab.sem.zona,
                          data.frame(sp = sp[i], scenario = scen, source = ids[1], target = ids[2],
                                     sim = j, tot.dist = Tot.len, euc.dist = ED, cost = NA))

    # print
    print(paste(i, j, ids[1], ids[2]))
  }

}

# No zoning
sp
(tab.com.zona <- data.frame(sp = NA, scenario = NA, source = NA, target = NA, sim = NA,
                            tot.dist = NA, euc.dist = NA, cost = NA)[-1,])
scen <- 'Land cover + Urban zoning'

# for each species
for(i in 1:length(corr.shp.com.zona)) {

  # for each simulation
  for(j in 1:length(corr.shp.com.zona[[i]])) {

    # select ID polygons
    (ids <- corr.shp.sts.com.zona[[i]][[j]])
    (st.ids <- ST.pol[ST.pol$gridcode == ids[1] | ST.pol$gridcode == ids[2],])

    # remove area within STs
    corr.cut <- rgeos::gDifference(corr.shp.com.zona[[i]][[j]], st.ids)

    # get initial and end coordinates
    coords <- do.call('rbind', lapply(coordinates(corr.cut)[[1]], function(x) x[c(1, nrow(x)),])) %>%
      SpatialPoints(proj4string = crs(st.ids))

    # initial coord
    pos.init <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[1],], byid = TRUE) %>%
      apply(MARGIN = 2, min) %>%
```

```
      which.min()
    init.pt <- coords[pos.init]

    # final coord
    pos.fin <- rgeos::gDistance(coords, st.ids[st.ids$gridcode == ids[2],], byid = TRUE) %>%
      apply(MARGIN = 2, min) %>%
      which.min()
    fin.pt <- coords[pos.fin]

    # Euclidean distance
    (ED <- raster::pointDistance(init.pt, fin.pt) %>% as.numeric)

    # calculate total corridor length
    (Tot.len <- rgeos::gLength(corr.cut))

    # Plot to check
    # plot(st.ids)
    # plot(corr.cut, add = T, col = 2)
    # plot(init.pt, add = 2, cex = 2, pch = 19)
    # plot(fin.pt, add = 2, cex = 2, pch = 19)

    # attach info
    tab.com.zona <- rbind(tab.com.zona,
                          data.frame(sp = sp[i], scenario = scen, source = ids[1], target = ids[2],
                                     sim = j, tot.dist = Tot.len, euc.dist = ED, cost = NA))

    # print
    print(paste(i, j, ids[1], ids[2]))
  }

}
```

Save corridors loaded without STs

```
save.image(paste0(corrdir, 'corridors_RSFI_noST.RData'))
```