

Proposition de protocole de suivi standardisé des Orthoptères du massif vosgien

Jean-Pierre Vacher et Roberto d'Agostino

Première version Mars, 2021. Mise à jour Juin, 2021

Contexte, objectif du suivi et indicateur mesuré

Une récente étude basée sur des données récoltées de manière opportuniste a montré des tendances de modification de l'aire de répartition de plusieurs espèces d'Orthoptères dans le massif des Vosges, avec une contraction de l'aire d'espèces montagnardes et montée en altitude d'espèces de plaine (D'Agostino and Vacher 2021). Ces changements d'aire de répartition sont probablement liés aux changements climatiques survenus ces vingt dernières années, mais les données utilisées étaient insuffisantes pour tester cette hypothèse. Afin d'apprécier l'effet de ces changements sur les Orthoptères du massif vosgien, nous proposons un protocole de suivi standardisé inspiré d'une étude récente menée sur les communautés d'Orthoptères d'un gradient altitudinal dans le Parc national du Mercantour (Mourguiart et al. 2020). Le protocole proposé dans le présent document a pour objectif de mesurer la variation de la répartition altitudinale du cortège d'Orthoptères du massif des Vosges. Pour cela, la stratégie d'échantillonnage est de disposer des placettes de suivi à différentes altitudes et de mesurer l'occurrence des espèces en prenant en compte la détection imparfaite. L'indicateur mesuré sera donc la variation temporelle de l'occurrence multi-espèces.

Stratégie d'échantillonnage

- Population statistique : Cortège des Orthoptères du massif vosgien et de la plaine
- Échantillon : 450 mailles de 500m x 500m réparties aléatoirement dans l'aire d'étude
- Unité d'échantillonnage : une maille de 500m × 500m

Nous n'avons pas opéré de simulations pour définir le nombre de mailles dans l'échantillon. Nous pensons cependant que 450 mailles représente un nombre suffisant d'unités d'échantillonnage pour obtenir une bonne robustesse statistique.

Procédure automatisée dans R pour générer l'échantillon

Charger les packages

Tout d'abord, nous chargeons une série de packages qui seront nécessaires pour la suite du script :

```
x = c("dismo", "ggmap", "rgdal", "rgeos", "maptools", "plyr", "dplyr", "tidyr", "raster",  
      "mapdata", "sp", "spdep")  
lapply(x, library, character.only = TRUE)
```

Définir l'aire d'étude

Dans un premier temps, nous définissons un polygone qui correspond à l'aire d'étude. Dans l'exemple ci-dessous, nous avons noté des coordonnées géographiques sur [Google maps](#) qui délimitent le massif des

Vosges plus la plaine de part et d'autre, jusqu'à la Moselle à l'ouest et jusqu'au Rhin à l'est. Ces coordonnées sont bien entendu interchangeables, nous les utilisons seulement pour l'exemple.

```
lon = data.frame(c(6.11, 6.11, 8.37, 8.37))
lat = data.frame(c(47.64, 49.17, 49.17, 47.64))
p = cbind(lon, lat)
colnames(p) = c("lon", "lat")
p = Polygon(p)
p = Polygons(list(p), 1)
p = SpatialPolygons(list(p))
proj4string(p) = CRS("+init=epsg:4326")
p = spTransform(p, CRS("+init=epsg:2154"))
```

Nous disposons maintenant d'un polygone en Lambert 93 (epsg:2154), donc des coordonnées métriques, qui correspond à l'aire d'étude. Nous allons transformer ce polygone en raster, avec des mailles de 500m x 500m, auxquelles nous allons associer un numéro qui sera séquentiel (1, 2, 3, 4,...)

```
r = raster(p) #s'assurer que p est en Lambert 93 (epsg:2154)
res(r) = 500 #500 correspond à la taille des mailles en mètres.
data = as.data.frame(matrix(1:(nrow(r) * ncol(r)), nrow = ncol(r), ncol = nrow(r)))
data = as.data.frame(t(as.matrix(data)))
rownames(data) = data$V1
values(r) = as.matrix(data)
```

Plus loin dans le script on voudra retirer les régions qui ne sont pas dans le Grand Est (Allemagne, Haute-Saône). Pour cela, on télécharge les contours de la régions depuis la couche OPENSTREETMAP disponible sur le l'URL <https://www.data.gouv.fr/en/datasets/r/01fdab09-ed86-4259-b863-69913a3e04d1>, puis on dépose la couche dans le dossier de travail

```
download.file(url = "http://osm13.openstreetmap.fr/~cquest/openfla/export/departements-20140306-100m-shp",
  destfile = "shapefiles/departements-20140306-100m-shp.zip")
unzip(zipfile = "shapefiles/departements-20140306-100m-shp.zip", exdir = "shapefiles/departements-20140306-100m-shp")
```

Une fois cette couche récupérée, on l'ouvre puis on sélectionne les départements du Grand Est :

```
dpt = readOGR(dsn = "departements-20180101.shp", layer = "departements-20180101")
gdEst = rbind(dpt[dpt@data$code_insee == "08", ], dpt[dpt@data$code_insee == "10",
  ], dpt[dpt@data$code_insee == "51", ], dpt[dpt@data$code_insee == "52", ], dpt[dpt@data$code_insee == "54", ],
  dpt[dpt@data$code_insee == "55", ], dpt[dpt@data$code_insee == "57", ],
  dpt[dpt@data$code_insee == "67", ], dpt[dpt@data$code_insee == "68", ], dpt[dpt@data$code_insee == "88", ])
gdEst = spTransform(gdEst, CRS("+init=epsg:2154"))
```

Puis, nous allons transformer de nouveau ce raster en couche vecteur avec polygones, pour obtenir une grille de mailles 500m x 500m qui recouvrira l'aire d'étude.

```
grid.poly = rasterToPolygons(r)
grid.poly = spTransform(grid.poly, CRS("+init=epsg:2154"))
grid.poly@data$ID = values(r)
grid.poly@data$layer = NULL
grid.poly = crop(grid.poly, gdEst)
raster::shapefile(grid.poly, file = "grille_globale.shp")
```

Sélectionner au hasard 150 lignes de la grille

Afin d'avoir un échantillon qui soit raisonnable en termes de prospections, nous allons sélectionner 150 lignes de la grille, qui en compte 358 (voir plus haut)

```
d = data.frame(data[, 1])
colnames(d) = "ID"
sample = sample(d$ID, 150, replace = F)
```

Compartimenter l'aire d'étude

Dans l'objectif d'avoir un échantillonnage qui soit équilibré entre les deux zones de plaine et la zone de montagne, nous allons compartimenter l'aire d'étude en trois sous-ensembles. Pour cela, nous divisons simplement le raster en trois sous-rasters.

Zone 1 : plaine côté Lorraine

```
rsub1 = r[1:nrow(r), 1:117, drop = F]
rsub1
d.sub1 = as.data.frame(matrix(values(rsub1), nrow = round(ncol(r)/3), ncol = nrow(r)))
d.sub1 = as.data.frame(t(as.matrix(d.sub1)))
d2 = d.sub1
rownames(d2) = data$V1
d3 = d2[which(rownames(d2) %in% sample), ]
d4 = gather(d3)
sample1 = sample(d4$value, 150, replace = F)
grid.poly1 = rasterToPolygons(rsub1)
grid.poly1 = spTransform(grid.poly1, CRS("+init=epsg:2154"))
grid.poly1@data$ID = values(rsub1)
grid.poly1@data$layer = NULL
grid.sample1 = grid.poly1[grid.poly1@data$ID %in% sample1, ]
```

Zone 2 : massif vosgien

```
rsub2 = r[1:nrow(r), (round(ncol(r)/3) + 1):((round(ncol(r)/3) + 1) + round(ncol(r)/3)),
drop = F]
rsub2
d.sub2 = as.data.frame(matrix(values(rsub2), nrow = 118, ncol = 358))
d.sub2 = as.data.frame(t(as.matrix(d.sub2)))
d2 = d.sub2
rownames(d2) = data$V1
d3 = d2[which(rownames(d2) %in% sample), ]
d4 = gather(d3)
sample2 = sample(d4$value, 150, replace = F)
grid.poly2 = rasterToPolygons(rsub2)
grid.poly2 = spTransform(grid.poly2, CRS("+init=epsg:2154"))
grid.poly2@data$ID = values(rsub2)
grid.poly2@data$layer = NULL
grid.sample2 = grid.poly2[grid.poly2@data$ID %in% sample2, ]
```

Zone 3 : plaine côté Alsace

```
rsub3 = r[1:nrow(r), (((round(ncol(r)/3) + 1) + round(ncol(r)/3)) + 1):ncol(r), drop = F]
rsub3
d.sub3 = as.data.frame(matrix(values(rsub3), nrow = 117, ncol = 358))
d.sub3 = as.data.frame(t(as.matrix(d.sub3)))
d2 = d.sub3
```

```

rownames(d2) = data$V1
d3 = d2[which(rownames(d2) %in% sample), ]
d4 = gather(d3)
sample3 = sample(d4$value, 150, replace = F)
grid.poly3 = rasterToPolygons(rsub3)
grid.poly3 = spTransform(grid.poly3, CRS("+init=epsg:2154"))
grid.poly3@data$ID = values(rsub3)
grid.poly3@data$layer = NULL
grid.sample3 = grid.poly3[grid.poly3@data$ID %in% sample3, ]

```

Assemblage de l'échantillon total

Une fois que les subset ont été générés, il suffit de les assembler pour avoir l'échantillon :

```

grid.sample = rbind(grid.sample1, grid.sample2, grid.sample3)
raster::shapefile(grid.sample, file = "grille_echantillon.shp")

```

Définir des répliquats spatiaux

Nous allons maintenant placer aléatoirement cinq répliquats spatiaux au sein de chaque maille de l'échantillon, qui correspondront à des mailles de 100m x 100m. Pour cela, nous définissons dans un premier temps une grille de 100m x 100m qui se superpose à la grille des échantillons.

```

rsubsample = raster(grid.sample)
res(rsubsample) = 100
rsubsample = rasterToPolygons(rsubsample)
subsample = raster::intersect(subsample, grid.sample)

```

Cependant, cette opération s'avère très coûteuse en mémoire et ne fonctionnera sans doute pas sur un ordinateur ordinaire. Deux solutions sont possibles. La première consiste à réaliser cette opération dans un logiciel de SIG de type QGIS ou ArcGIS sur un ordinateur possédant un processeur puissant et beaucoup de RAM (de 16 à 32 GO de RAM). Autrement, il est également possible de continuer dans R en créant des sous-ensembles, puis nous les assemblerons ensuite. C'est ce que nous déroulons ci-après :

Sous-grille n°1

```

rsubsample1 = raster(grid.sample1)
res(rsubsample1) = 100
subsample1 = rasterToPolygons(rsubsample1)
subsample1 = raster::intersect(subsample1, grid.sample1)

```

Sous-grille n°2

```

rsubsample2 = raster(grid.sample2)
res(rsubsample2) = 100
subsample2 = rasterToPolygons(rsubsample2)
subsample2 = raster::intersect(subsample2, grid.sample2)

```

Sous-grille n°3

```

rsubsample3 = raster(grid.sample3)
res(rsubsample3) = 100

```

```
subsample3 = rasterToPolygons(rsubsample3)
subsample3 = raster::intersect(subsample3, grid.sample3)
```

Une fois les trois sous-ensembles obtenus, nous les rassemblons :

```
subsample = rbind(subsample1, subsample2, subsample3)
subsample$ID_plot = subsample@data$ID
subsample$ID_subplot = seq.int(nrow(subsample))
raster::shapefile(subsample, file = "sous_grille_globale.shp")
```

Enfin, nous procédons à une sélection aléatoire des cinq réplicats spatiaux au sein de chaque maille de l'échantillon. Chaque réplicat sera numéroté de 1 à 5 dans les mailles, et contiendra le numéro de la maille dans la table de métadonnées.

```
spac.rep = ddply(as.data.frame(subsample), .(ID), function(x) x[sample(nrow(x), 5),
])
spac.rep = subsample[match(spac.rep$ID_subplot, subsample@data$ID_subplot, nomatch = 0),
]
spac.rep@data$ID_rep = ave(spac.rep@data$ID_subplot, spac.rep@data$ID_plot, FUN = seq_along)
spac.rep@data$layer = NULL
spac.rep@data$ID = NULL
spac.rep
raster::shapefile(spac.rep, file = "replicats_spatiaux.shp")
```

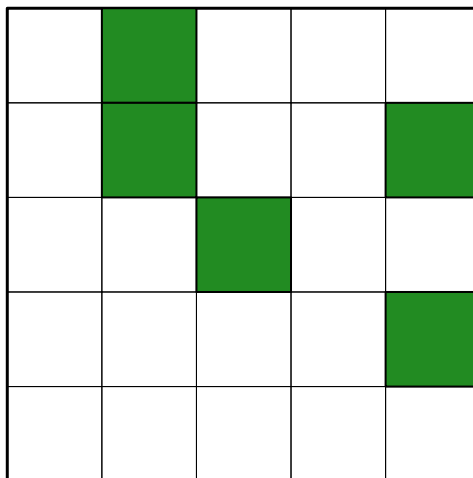


Figure 1: Schéma d'une maille échantillon de 500m x 500m avec cinq réplicats spatiaux de 100m x 100m en vert.

Protocole de récolte des données

L'observateur aura à charge de noter la présence de toutes les espèces d'Orthoptères au sein des réplicats spatiaux.

Nombre de passages

Un passage par maille est prévu par an.

Durée de prospection

Le temps de prospection par réplicat spatial sera de 30 minutes. Cette durée pourra être ajustée les années suivantes selon les résultats de la première année de suivi.

Covariables de session

- Température, mesurée à hauteur d’homme
- Nébulosité, selon trois classes : ensoleillé, variable, couvert

Covariables d’occurrence

- Altitude

Fréquence du suivi

Le suivi sera mené durant deux années d’affilée au début, puis une fois tous les cinq ans.

Références

- D’Agostino, Roberto, and Jean-Pierre Vacher. 2021. “Évolution temporelle de la répartition des Orthoptères dans le massif vosgien : possibles effets des changements climatiques sur la biodiversité.” *Bulletin de La Société d’Histoire Naturelle Et d’Ethnographie de Colmar* 77 (2): 10–17.
- Mourguiart, Bastien, Thibaut Couturier, Yoan Braud, Jérôme Mansons, Damien Combrisson, and Aurélien Besnard. 2020. “Multi-species occupancy models: an effective and flexible framework for studies of insect communities.” *Ecological Entomology*, December, een.12991. <https://doi.org/10.1111/een.12991>.