

'Procédure de sélection des flux à cartographier selon un critère spatial – Le voisinage spatial binaire : la contiguïté (Cij=1)

F. Bahoken

Janvier, 2016

1. Présentation :

Cette annexe fait partie d'une série de procédures permettant de sélectionner les flux selon des critères liés à l'espacement des lieux. Le présent document (1/4) porte sur la construction d'une matrice de voisinage ordinaire, notée (Cij=1). Il s'agit d'une matrice de contiguïté d'ordre 1 entre les couples de lieux (i,j), ou la valeur 1 correspond aux couples de lieux (i,j) qui partagent une frontière de longueur non nulle, sinon elle est à 0.

On mobilise pour cela un fond de carte surfacique. Le cas d'illustration est le découpage communal de l'Isère. La matrice de flux associée est celle des flux domicile-travail de l'Isère que l'on réduira aux seuls flux s'exprimant entre des couples de lieux contigus (ou limitrophes): (Fij)<- (Cij=1)

Données exemple

isere_com.shp : fond de carte des communes de l'Isère **matrice_isere.csv** : matrice de flux de l'Isère (format matriciel)

2. Préparation

Déclaration des packages et connexion

```
#purge environnement
rm(list=ls())
```

```
library(rgeos)
```

```
## rgeos version: 0.3-8, (SVN revision 460)
## GEOS runtime version: 3.4.2-CAPI-1.8.2 r3921
## Polygon checking: TRUE
```

```
library(maptools)
```

```
## Loading required package: sp
## Checking rgeos availability: TRUE
```

```
library(proj4)
library(reshape2)
library(rCarto)
```

```
## Loading required package: RColorBrewer
## Loading required package: classInt
```

```
library(rgdal)
```

```
## rgdal: version: 0.9-1, (SVN revision 518)
## Geospatial Data Abstraction Library extensions to R successfully loaded
## Loaded GDAL runtime: GDAL 1.11.0, released 2014/04/16
## Path to GDAL shared files: C:/Users/bahoken.AD-IFSTTAR/Documents/R/win-library/3.1/rgdal/gdal
## GDAL does not use iconv for recoding strings.
## Loaded PROJ.4 runtime: Rel. 4.8.0, 6 March 2012, [PJ_VERSION: 480]
## Path to PROJ.4 shared files: C:/Users/bahoken.AD-IFSTTAR/Documents/R/win-library/3.1/rgdal/proj
##
## Attaching package: 'rgdal'
##
## The following object is masked from 'package:proj4':
##
##   project
```

Chargement du fond de carte

```
getwd() #où suis-je ?
```

```
## [1] "C:/FRANCOISE/R/2-1_contig_ok/v_1"
```

```
setwd("C:/FRANCOISE/R/2-1_contig_ok")  
  
# Chargement du fond de carte  
fdc<- readShapeSpatial("./isere_com.shp")  
colnames(fdc@data)[4]<-"INSEE_COM"
```

3. Construction de la matrice de voisinage ordinaire (Cij=1)

Création de la matrice de contigüité (contig)

contig=(Cij=1). La valeur 1 correspond à l'existence d'une frontière de longueur non nulle entre (i) et j, sinon (Cij=0).

```
# la fonction gintersect agit sur la geometrie du fdc  
contig<-gIntersects(fdc, byid = TRUE, prepared=TRUE)  
row.names(contig)<-fdc@data$INSEE_COM  
colnames(contig)<-fdc@data$INSEE_COM  
  
# mise à 1 ou 0 de la matrice  
for (i in 1:nrow(contig))  
  for (j in 1:ncol(contig))  
    {if (contig[i,j]==TRUE) {contig[i,j]<-1}  
      if (contig[i,i]!=0) {contig[i,i]<-0}  
    }
```

Transformation de contig en liste

```
tab <-melt(contig)  
names(tab) = c("CODE_i", "CODE_j", "cij")  
head(tab)
```

```
##   CODE_i CODE_j cij  
## 1  38226  38226  0  
## 2  38206  38226  0  
## 3  38301  38226  0  
## 4  38448  38226  0  
## 5  38287  38226  0  
## 6  38153  38226  0
```

La représentation de cette matrice conduit à un graphe de voisinage d'ordre 1, qu'il est possible de visualiser.

4. Visualisation du graphe de voisinage (Cij=1)

Extraction des centroïdes (du fond de carte (fdc))

```
pt <- cbind(fdc@data[, "INSEE_COM"], as.data.frame(coordinates(fdc)))  
colnames(pt) <- c("CODE", "X", "Y")  
head(pt)
```

```
##   CODE      X      Y  
## 0 38226 917543.1 6416638  
## 1 38206 933896.6 6464035  
## 2 38301 907669.5 6413449  
## 3 38448 840772.2 6484481  
## 4 38287 893903.3 6483937  
## 5 38153 904873.1 6458243
```

Jointure habituelle avec le fond de carte

```
tabflow<-tab  
tabflow = data.frame(tabflow, pt[match(tabflow[, "CODE_i"], pt[, "CODE"]), 2:3])  
tabflow = data.frame(tabflow, pt[match(tabflow[, "CODE_j"], pt[, "CODE"]), 2:3])  
colnames(tabflow) <- c("i", "j", "Cij", "X1", "Y1", "X2", "Y2")  
head(tabflow)
```

```
##   i     j Cij      X1      Y1      X2      Y2  
## 0 38226 38226  0 917543.1 6416638 917543.1 6416638  
## 1 38206 38226  0 933896.6 6464035 917543.1 6416638  
## 2 38301 38226  0 907669.5 6413449 917543.1 6416638
```

```
## 3 38448 38226 0 840772.2 6484481 917543.1 6416638
## 4 38287 38226 0 893903.3 6483937 917543.1 6416638
## 5 38153 38226 0 904873.1 6458243 917543.1 6416638
```

Nettoyage de la table en supprimant les valeurs nulles

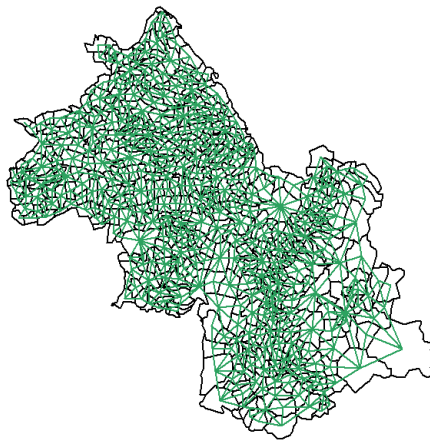
```
contig_1<-tabflow[tabflow[,"Cij"]!=0,]
```

Export éventuel de la table (utile pour la création des matrices de contiguité ordinale, voir billet (2/4))

```
write.table(contig_1,file="CKij_1.csv",sep=";",eol="\n")
```

Tracé du graphe de voisinage

```
plot(fdc)
trace<-segments(contig_1$X1, contig_1$Y1, contig_1$X2, contig_1$Y2,col="#2ca25f")
```



5. Restriction de (Fij) en fonction de (Cij)

Pour cartographier les flux qui se produisent entre des zones adjacentes - les seuls couples de lieux correspondant à ce graphe de voisinage -, il convient de réduire la matrice (Fij) en fonction de (Cij)=1. Cela revient à comparer terme à terme deux matrices de taille similaire.

On va donc créer à nouveau les liens de la matrice de flux, avant de la réduire.

Préparation du tracé des liens de la matrice (jointures, ...)

On repart du fichier de points initial

```
head(pt)
```

```
##      CODE      X      Y
## 0 38226 917543.1 6416638
## 1 38206 933896.6 6464035
## 2 38301 907669.5 6413449
## 3 38448 840772.2 6484481
## 4 38287 893903.3 6483937
## 5 38153 904873.1 6458243
```

et de Cij, qui correspond au tabflow, on renomme ce dernier pour éviter les confusions

```
Cij<-tabflow
head(Cij)
```

```
##      i      j Cij      X1      Y1      X2      Y2
## 0 38226 38226 0 917543.1 6416638 917543.1 6416638
## 1 38206 38226 0 933896.6 6464035 917543.1 6416638
```

```
## 2 38301 38226 0 907669.5 6413449 917543.1 6416638
## 3 38448 38226 0 840772.2 6484481 917543.1 6416638
## 4 38287 38226 0 893903.3 6483937 917543.1 6416638
## 5 38153 38226 0 904873.1 6458243 917543.1 6416638
```

On importe la matrice de flux, au format liste

```
Fij<-read.csv2("./liste_isere.csv",header=TRUE,sep=";",stringsAsFactors=FALSE,encoding="UTF-8",dec=".",check.names=FALSE)
```

```
head(Fij)
```

```
##      i      j Fij
## 1 38001 38001  0
## 2 38002 38001  0
## 3 38003 38001  0
## 4 38004 38001  0
## 5 38005 38001  0
## 6 38006 38001  0
```

On réalise la traditionnelle jointure entre (Fij) et (pt) pour récupérer les codes

```
Fij = data.frame(Fij, pt[match(Fij[, "i"], pt[, "CODE"]), 2:3])
Fij = data.frame(Fij, pt[match(Fij[, "j"], pt[, "CODE"]), 2:3])
colnames(Fij) <- c("i", "j", "Fij", "X1", "Y1", "X2", "Y2")
head(Fij)
```

```
##      i      j Fij      X1      Y1      X2      Y2
## 253 38001 38001  0 902347.2 6496924 902347.2 6496924
## 63  38002 38001  0 934589.3 6466585 902347.2 6496924
## 260 38003 38001  0 845671.2 6472517 902347.2 6496924
## 381 38004 38001  0 892928.9 6460737 902347.2 6496924
## 51  38005 38001  0 937063.9 6456949 902347.2 6496924
## 47  38006 38001  0 943664.5 6480414 902347.2 6496924
```

Application d'une variable (Cij) de restriction sur la matrice Fij: [(Fij)<- (Cij=1)]

On crée au passage une nouvelle variable (Cij_Fij) qui sera ensuite cartographiée. Attention : le temps de calcul peut être très long

```
for (k in 1:nrow(Fij))
{
  if (Cij$Cij[k]!=0) Fij$Cij_Fij[k]<-Fij$Fij[k]
  else Fij$Cij_Fij[k]<-0
}
```

```
head(Fij)
```

```
##      i      j Fij      X1      Y1      X2      Y2 Cij_Fij
## 253 38001 38001  0 902347.2 6496924 902347.2 6496924  0
## 63  38002 38001  0 934589.3 6466585 902347.2 6496924  0
## 260 38003 38001  0 845671.2 6472517 902347.2 6496924  0
## 381 38004 38001  0 892928.9 6460737 902347.2 6496924  0
## 51  38005 38001  0 937063.9 6456949 902347.2 6496924  0
## 47  38006 38001  0 943664.5 6480414 902347.2 6496924  0
```

6. Cartographie de la sélection de flux qui s'expriment entre des zones limitrophes (adjacentes, contigües d'ordre 1).

Pour cartographier cette matrice (Fij) réduite, voir billet [Carto.Flux_1](http://elementr.hypotheses.org/370). Procédure de cartographie d'une matrice de flux

<http://elementr.hypotheses.org/370>