Cartograflow

Filtering Origin-Destination Matrices For Flow Mapping Purposes in R

Françoise BAHOKEN,
IFSTTAR / AME-SPLOTT & ass. UMR Géographie-Cités

21st *European colloquium on theoretical and quantitative geography* - **ECTQG 2019**
September 07, 2019 - Mondorf-les-Bains (Luxembourg)

Outline

1. **Introduction**
2. **Technical choices**
   (R and relating packages)
3. **{Cartograflow} general overview**
4. **Case study**
5. **Conclusion**
   (To do and remaining challenges)

1. Introduction

Reducing the Visual complexity of the so-called flowmaps

- General aim: geovisualisation of spatial interactions patterns through flows on thematics maps, clearly (as Tobler's said)
- The first problem to solve is to reduce the **graphic complexity** ("spaghetti-effect") due to the representation of N(N-1) links and (N) places
- How to plot on a map - in am meaningfull way - origin-destination linear features ?

This comes after: Bahoken (2016), *Contribution à la cartographie d'une matrice de flux* appendices
https://halshs.archives-ouvertes.fr/tel-01273776

Introduction

Existing solutions

1. Graphic, retinan ;
2. Geo-cartographic information (background) ;
3. Statistical or **numeric** information:
   – on places of origine (i) and/or destination (j)) - punctual features ;
   – on **interactions** (Fij) between (i) and (j) - mostly linear features,

In order to reduce the flow data values or features by:
- clusterisation / aggregation ;
- **selection / thresholding**

2. Technical choices

2.1 The choice of R platform

The choice of R/Rstudio rather than GIS software or purely statistical tools:

- (relatively) easy to handle ;
- open access and (easily) accessible ;
- provide reproducible research in the specific cartographic area ;
- facility of writing documents (with .Rmd formats) for educational purposes (here it is *ioslides_presentation* rendering in html from .Rmd)

2.2. Relating R packages

MATRIX FILTERING

`{igraph}` and `{matrix}`
- general packages for mathematical playing with matrixes ;

`{flows}` Laurent Beauguitte et al. (2016)
- dedicated to local flow selection as a variant of Nyusten & Dacey (1961) dominant flow

`{migR}` Matthieu Garnier et al. (2019)
- focused on residential migrations indicators computation and local flow mapping

`{stplanR}` Robin Lovelace (2016)
- dedicated to some transportations issues, for planar graphs

2.2. Relating R packages

MATRIX MAPPING

`{igraph}` and `{ggplot2}` Hadley Wickham (en 2005)
- general packages for plotting and no geographic links features

`{cartography}` Timotée Giraut & Nicolas Lambert (2015)
- dedicated to thematic, mainly choropleth cartography with various way of representation and advanced thematic mapping rendering options
- limited for flow mapping to small and symmetric matrixes

2.3. Specificities of {cartograflow}

- Dedicated to general OD issues, i.e. for non planar networks that required global thresholding (to this day) ;
- Allows spatial filtering procedure, via an external matrix of (XY) positions: Compute continuous or discrete distance travelled by the flow :
- Filtering procedures with embedded flow mapping process:
  – provide integrated plotting function ;
  – allow to plot graduated and (non) oriented straight lines

3. {cartograflow} general overview

{Cartograflow}: available matrix formats

- **List format "L"**:
  a .csv 3 column flow dataset (origin, destination, flow_value)

- **Matrice format "M"**:
  a .csv [n*n] flow dataset.

`flowtabmat()`:
- to convert from "L" to "M" format - as {base::dcast} ;
- to check if your matrix is square or not ;

{Cartograflow}: useful functions

`flowcarre()`:
- to close and square [n,n] an asymetric matrix ;
- from "L" or "M" format.

`flowtype()`:
- to **compute the main types of flows** from descriptive analysis
(from an asymmetric flow matrix, in "M" or "L" format) ;
- result is (as Tobler's) bilateral gross or net flows (symetric and skew symetric) matrix.

`flowjointure()`:
- to perform an attribute spatial join - by (i) and (j) between a .csv flow dataset and a .shp spatial shape ;
- to transfert the OD centroïd coordinates (Xi, Yi, Xj, Yj) of a .shp areal background to the flow matrice.

{Cartograflow}: main functions

`flowreduct()`:
- for **filtering flows by an external matrix** (e.g. a matrix of continuous distance).
- the *select* criterion set as:
– *dmin* is for selecting the min value to plot - ie. up to x km ;
– *dmax* is for selecting values - ie. less than x km

`flowgini()`:
- performs a concentration analysis of a flow dataset ;
- computes *Gini coefficient* and plot interactive *Lorenz curve*
- to be use before `flowanalysis()`

`flowanalysis()`:
- to compute a global filter criterion based on
– flow's *significativity* (% of total interactions) ;
– and/or *flow's density* (% of total linear features).

{Cartograflow}: main functions

`flowdistance()`:
- for computing a *continuous distance* with several additional parameters ;
- for filtering flows by a distance travelled

`flowcontig()`;
- compute a *discrete (ordinal) distance* matrix based on (k) contiguïty ;
- where (k) is the rank parameter (1:n-1) defined as the number of borders to be crossed between origins and destinations places

`flowmap()`:
- is for plotting OD flows ;
- by filtering values or features, or not ;
- with straight features, oriented (arrows) or not

4. Case study

The greater Paris commuters

Data

- Statistical dataset :
  "*Base flux de mobilite*" MOBPRO - commuters - INSEE, 2015
- Geographical dataset :
  municipalities from IGN, data preparation by APUR & UMS Riate, 2017

```
## 'data.frame':    4692 obs. of  3 variables:
##  $ i  : int  75101 75101 75101 75101 75101 75101 75101 75101 75101 75101 ...
##  $ j  : int  75102 75105 75108 75109 75112 75113 75115 75116 92012 92026 ...
##  $ Fij: int  247 104 426 263 123 139 134 123 128 139 ...
```

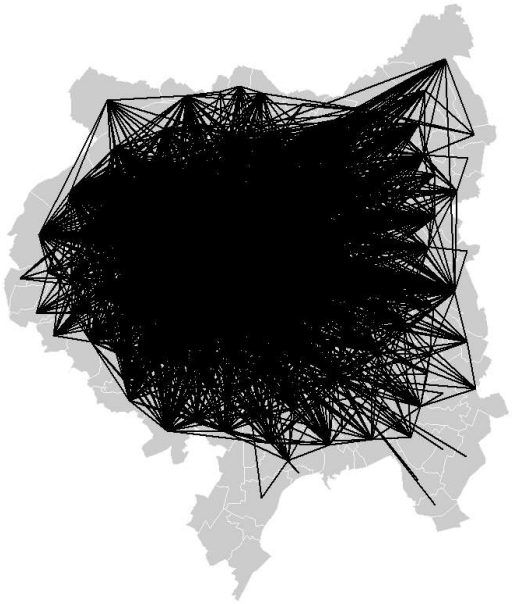
```
## Reading layer `MGP_communes' from data source `D:\R\ECTQG\fdc_data\MGP_communes.shp' using driver `ES
## Simple feature collection with 150 features and 10 fields
## geometry type:  POLYGON
## dimension:      XY
## bbox:           xmin: 637297 ymin: 6838631 xmax: 671756 ymax: 6879246
## epsg (SRID):    NA
## proj4string:    +proj=lcc +lat_1=49 +lat_2=44 +lat_0=46.5 +lon_0=3 +x_0=700000 +y_0=6600000 +ellps=GR
```

Geodata

Geovisualizing Greater Paris commuters

Revealing graphic complexity

Plotting these matrix on a map reveals the so-called *spaghetti-effect*
Hereby, plotting of all theoretical OD links (without filter)



4. Global filtering to clarify the map

4.1. Numerical filtering by a unique parameter

Global criteria means **unique** parameter to be apply to all the cells of the matrix

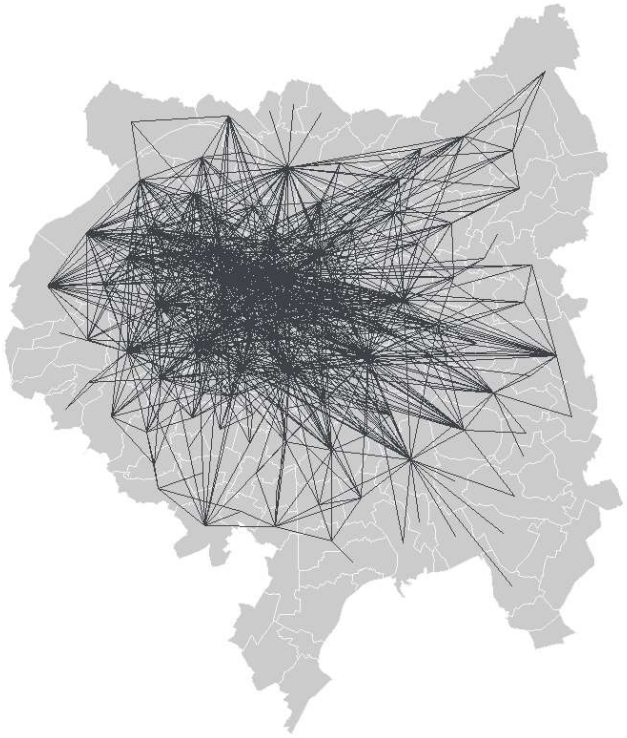
```
#eg : the mean value
tabflow <-tabflow %>%
  filter(tabflow$Fij!=0)  # matrix reduction to existing flow values

X<-mean(tabflow$Fij)

flowmap(tab = tabflow,
                format="L",
                fdc="./fdc_data/MGP_communes.shp",
                code="IDCOM",
                filter=TRUE,
                a.col="#3f4247",
                threshold= X   #flow value > X = 348
               )
```

4.2. Numerical filtering by a unique parameter

Plotting flow value up to the criteria Fij > alpha
alpha=348 (mean value)



4.3. Numerical filtering

Applying a concentration criteria

(1/3) Computes Gini's coefficent

```
tab_gini<-flowgini(tabflow.sq, format="L", origin="i",dest="j", valflow="ydata",
          fdc = "./fdc_data/MGP_communes.shp",code="IDCOM", lorenz.plot = FALSE)

head(tab_gini)
```

```
##          i     j ydata       X1      Y1       X2      Y2 link     flowcum
## 1067 75117 75108  4857 649165.4 6865479 649575.7 6863852    1 0.003117114
## 1065 75115 75108  4852 648097.2 6860237 649575.7 6863852    1 0.006231019
## 1066 75116 75108  4000 645848.7 6862519 649575.7 6863852    1 0.008798130
## 1068 75118 75108  3560 652206.2 6866034 649575.7 6863852    1 0.011082859
## 2281 92012 75116  3263 644144.9 6859859 645848.7 6862519    1 0.013176979
## 4515 75115 92012  3142 648097.2 6860237 644144.9 6859859    1 0.015193445
##           linkcum
## 1067 0.0002292001
## 1065 0.0004584002
## 1066 0.0006876003
```

4.3. Numerical filtering

Applying a concentration criteria

(2/3) Plot Lorenz curve

4.3. Numerical filtering

Applying a concentration criteria

(3/3) Compute critflow parameter and flowmap

```
flowanalysis(tab_gini, critflow = 0.02, result = "signif")
```

```
## [1] "threshold =  3070  ---  flows =  2 % ---  links =  0.18 %"
```

4.3. Numerical filtering

Applying a concentration criteria

(3-bis) Compute critlink parameter and flowmap

```
flowanalysis(tab_gini,critlink = 0.01,result = "density")
```

```
## [1] "threshold =  2045  ---  flows =  7.21 % ---  links =  1 %"
```

Plot 4 % of the total features, flow greater than 2015

4.4. Spatial filtering

- Spatial means filtering by the **distance** travelled between origin and destination ;
- Involves a matrix distance - so the criteria is **continuous**

4 steps involved:
- 1) compute a distance matrix ;
- 2) plot the corresponding graph ;
- 3) filter the matrix ;
- 4) Reduce the flow matrix by the filtered distance matrix

4.4. Spatial filtering

Continuous distance travelled

(1/3) Compute continuous distance matrix from a shapefile (through jointure)
Can be done with a .csv in {base}

```
tab.distance<-flowdist(tab, dist.method = "euclidian",result = "dist")
head(tab.distance)
```

```
##       i     j  distance
## 1 75101 75102  787.3099
## 2 75101 75105 2270.1051
## 3 75101 75108 2087.3999
## 4 75101 75109 1629.7126
## 5 75101 75112 6948.2107
## 6 75101 75113 4240.7855
```

4.4. Spatial filtering

Continuous distance travelled

(2/3) Reduce the distance matrix
Example: Compute the summary to choose the threshold criterion. The short distance (as Dij< Q1)

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0    3793    7016    7586   10619   27067
```

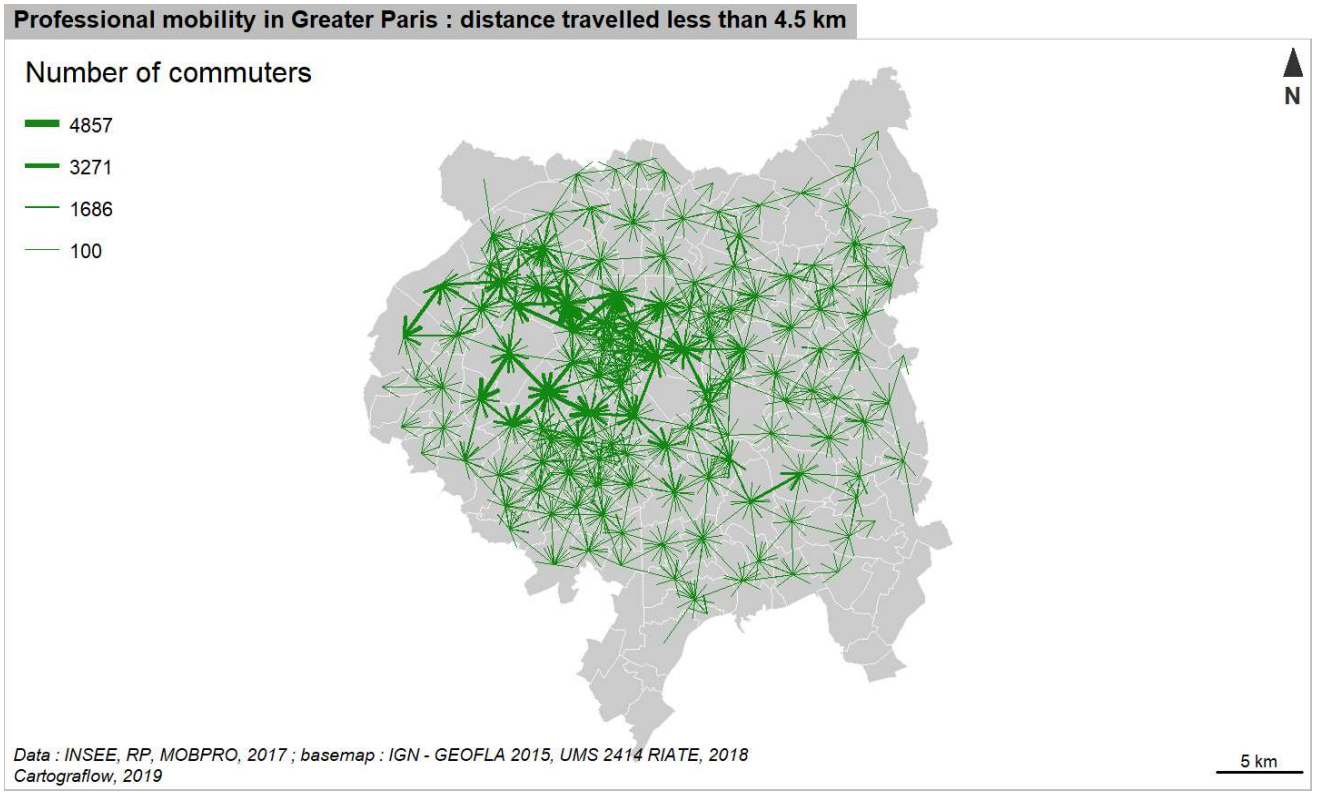
Reduce the flow dataset from a selected distance travelled Dij < 3793 m (Q1)

```
tab.flow.dmin<-flowreduct(tabflow.sq,tab.distance,metric = "continous",
                    select = "dmax", # max distance parameter to plot
                    d = 3793)        # corresponding max distance value to plot
#select for all i,j flow values up to 0
flow.dmin<-tab.flow.dmin%>%
        select(i,j,flowfilter)%>%
        filter(flowfilter !=0)
```

4.4. Spatial filtering

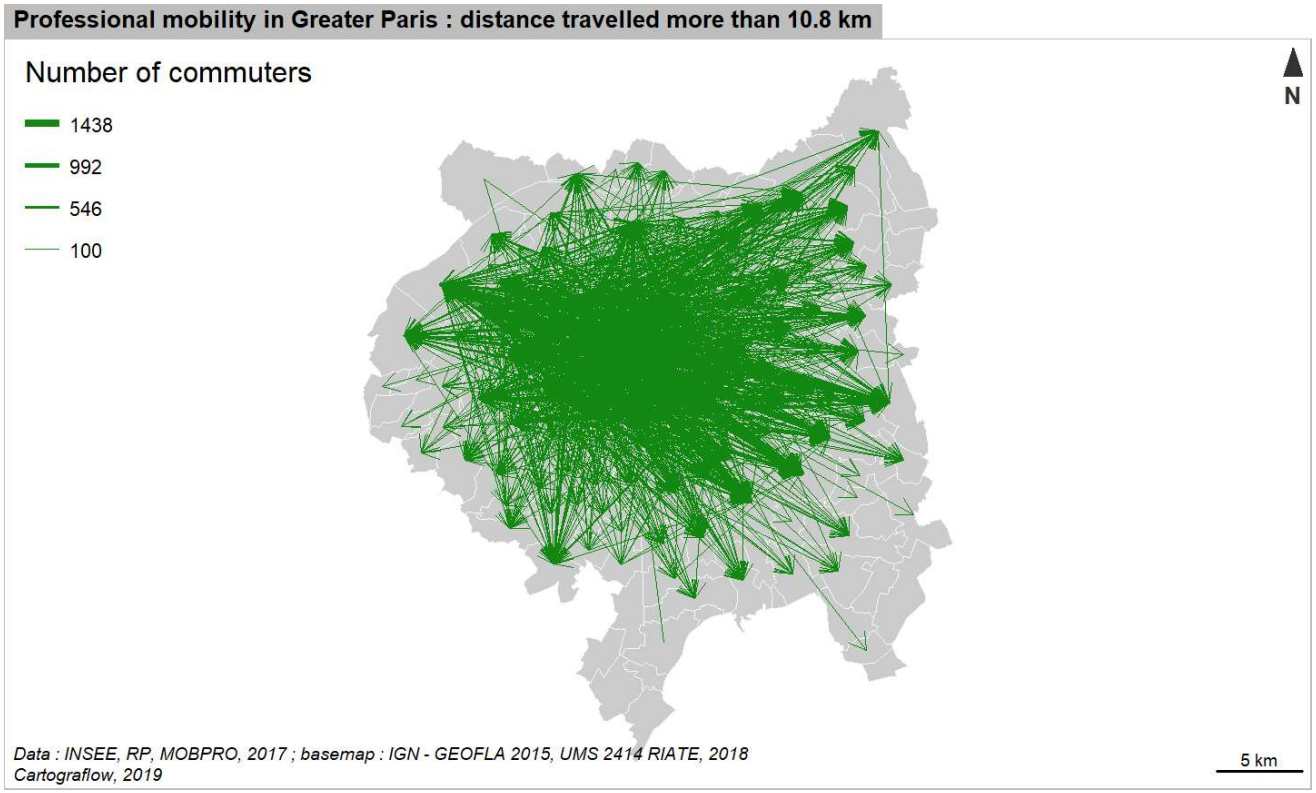
Continuous distance - short distance travelled (Dij) < Q1

(3/3) Reduce the flow matrix by the distance matrix filtered



4.4. Spatial filtering

Continuous distance - long distance travelled Dij > Q3



4.5. Territorial filtering

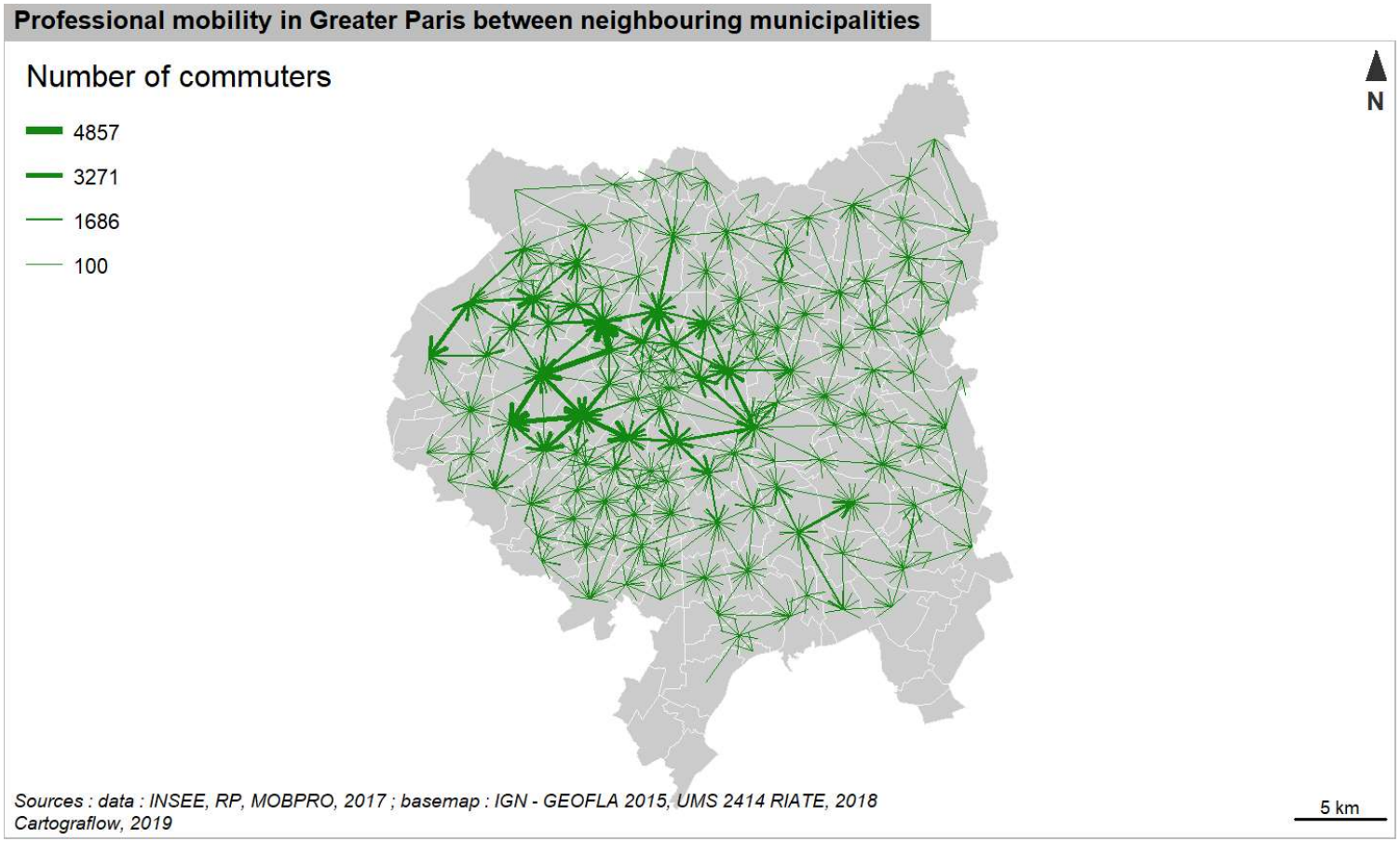
Discrete / Ordinal distance

(1/2) Building the neighbouring graph (ex. rank 1)

4.5. Territorial filtering

Discrete / Ordinal distance

Reduce the flow matrix to only selected neighbouring flow values



Conclusion

- This is a 1st version of {cartograflow} dedicated to **global** filtering ;
- It combines the application of a filtering criterion with a plotting procedure ;
- Two matrix format are available, for direct compatibility with statistical and thematic mapping packages ;
- A voluntary simple, general and generalizable approach for pedagogic issues

Conclusion

Todolist and remaining challenges

- **DATA FORMAT**

  – Loading an (X,Y) .csv file in flow mapping procedure ;
  – Trying to playing with complex data sets (i.e. temporal, categorial)

- **DRAWING FEATURES**

  – Dissociation and rendering of arrows in parallel ;
  – Rendering arrows as a curve ;
  – Trying edge bundling procedures ;

Conclusion

Todolist and remaining challenges

- **DATA FILTERING**

  – Adding **local filtering**:
  selecting nodes ;
  applying *dominant flows analysis* for drawing selected links ;
  – Adding complementary contiguity matrix, ex. Queen neighbours ;
  – Adding matrix reduction by **clustering**, especially after dominant analysis

THANKS FOR YOUR ATTENTION

Cartograflow
– CRAN: https://cran.r-project.org/web/packages/cartograflow/index.html
– Github: https://github.com/fbahoken/cartogRaflow

All collaborations are welcome !

Françoise BAHOKEN
IFSTTAR / AME-SPLOTT
Ass. UMR 8504 Géographie-Cités
francoise.bahoken@isffar.fr
@fbahoken