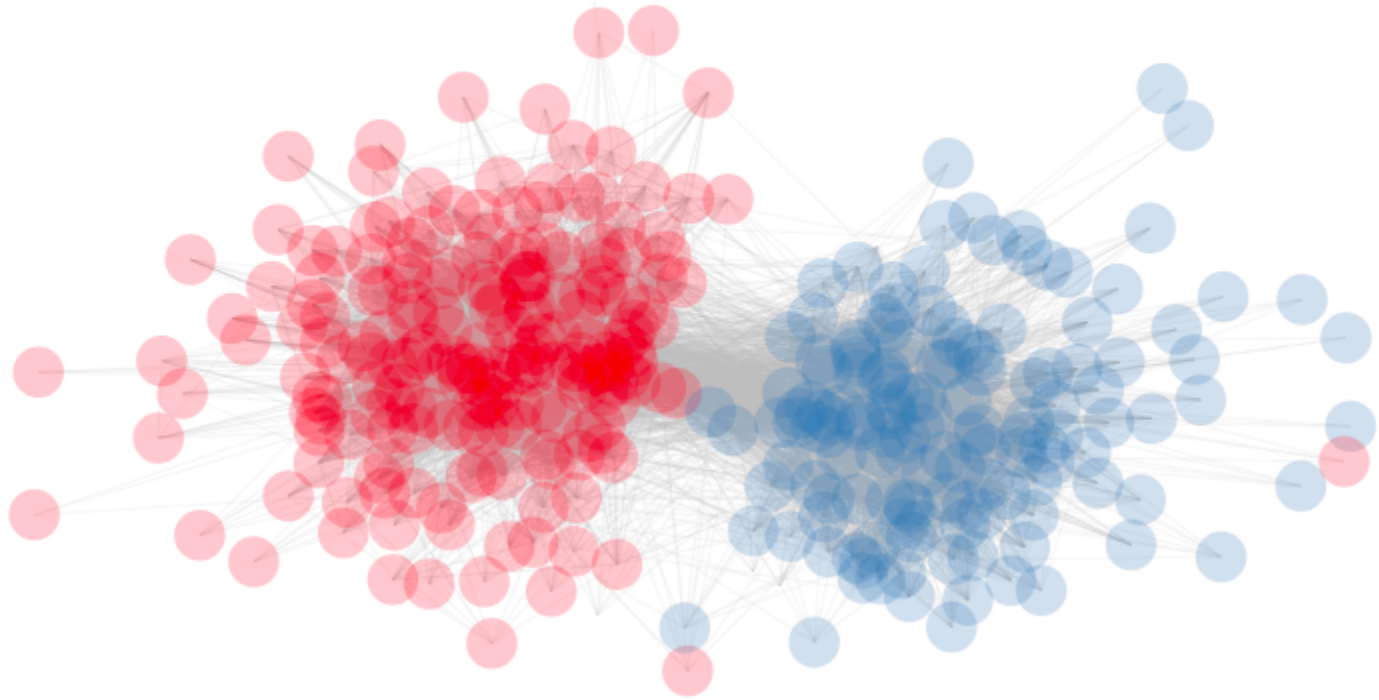
François Briatte

R Addicts Paris 2013

Collaboration Networks Among French Members of Parliament



Demo code: https://gist.github.com/briatte/7214253

To install the `GGally` package:

```
# from CRAN:
install.packages("GGally")

# from GitHub:
devtools::install_github("ggobi/ggally")
```

To replicate the examples:

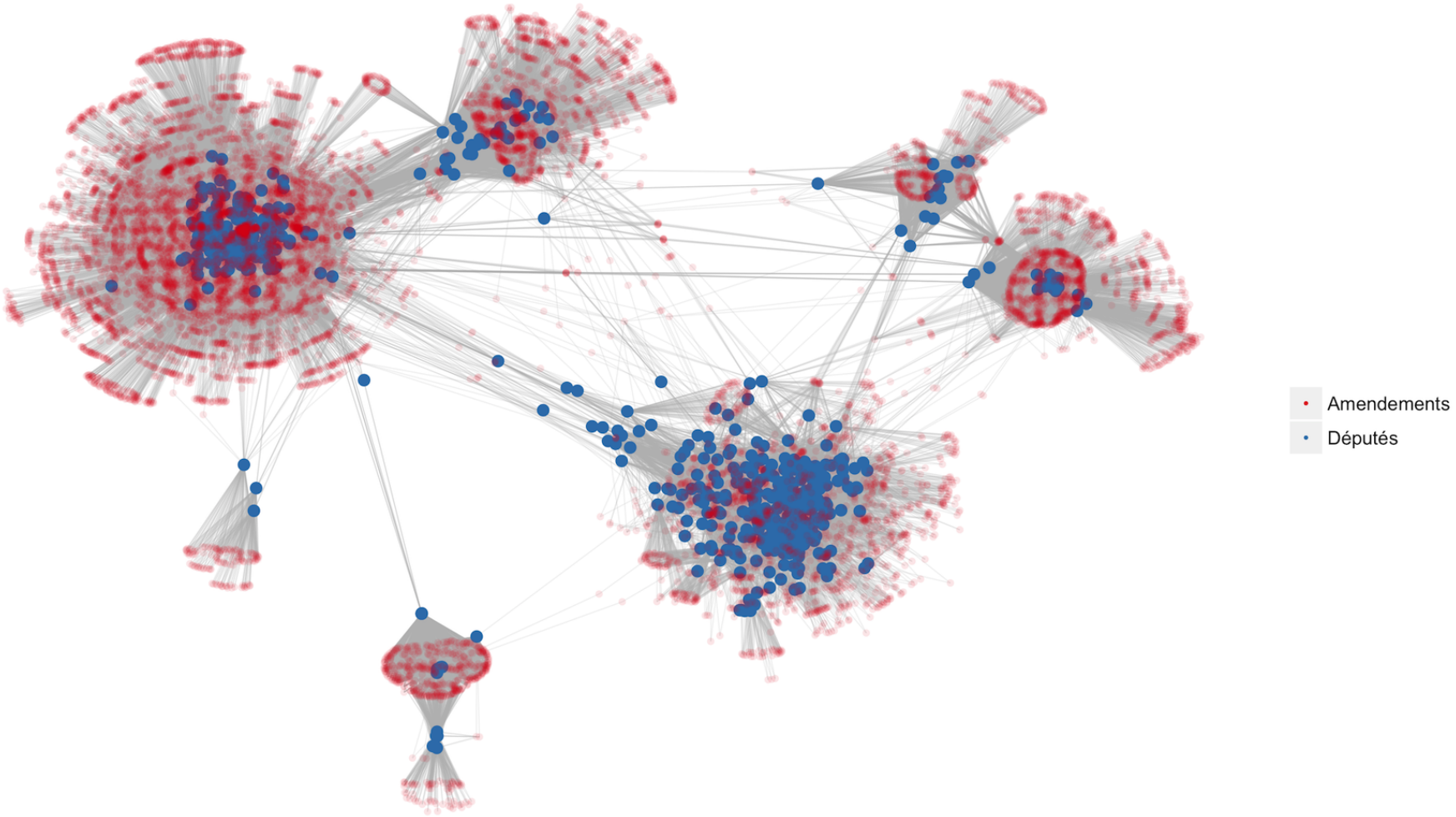
```
devtools::source_url("https://gist.github.
com/briatte/7214253/raw/601a28cd202720248d894b183fa0df
9e30d55d72/ggnet.demo.r", prompt = FALSE, verbose =
TRUE)
```

Network ties:

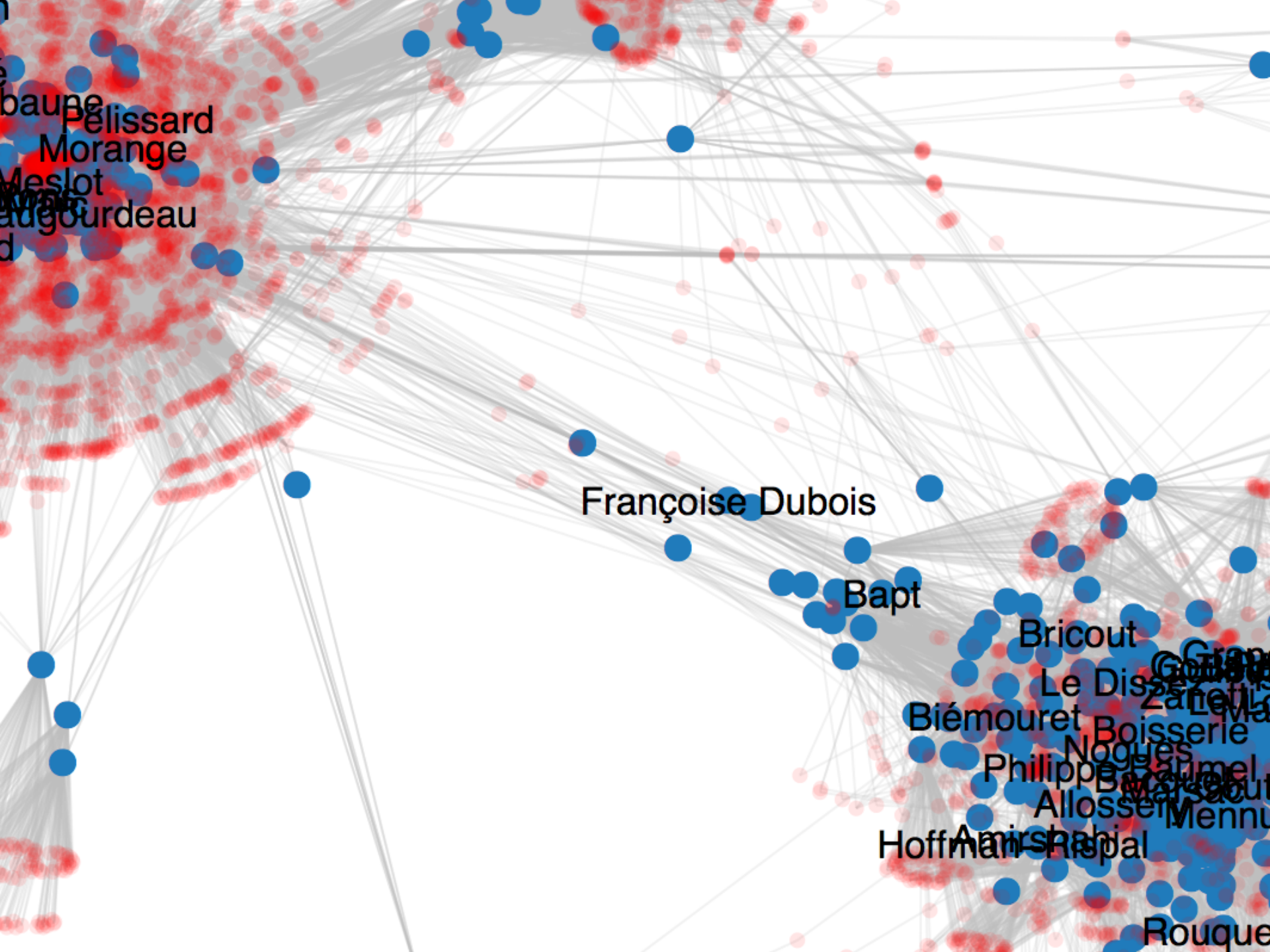
- Amendment cosponsorships
- Bill cosponsorships (Baptiste Coulmont)
- Twitter “follower/following”

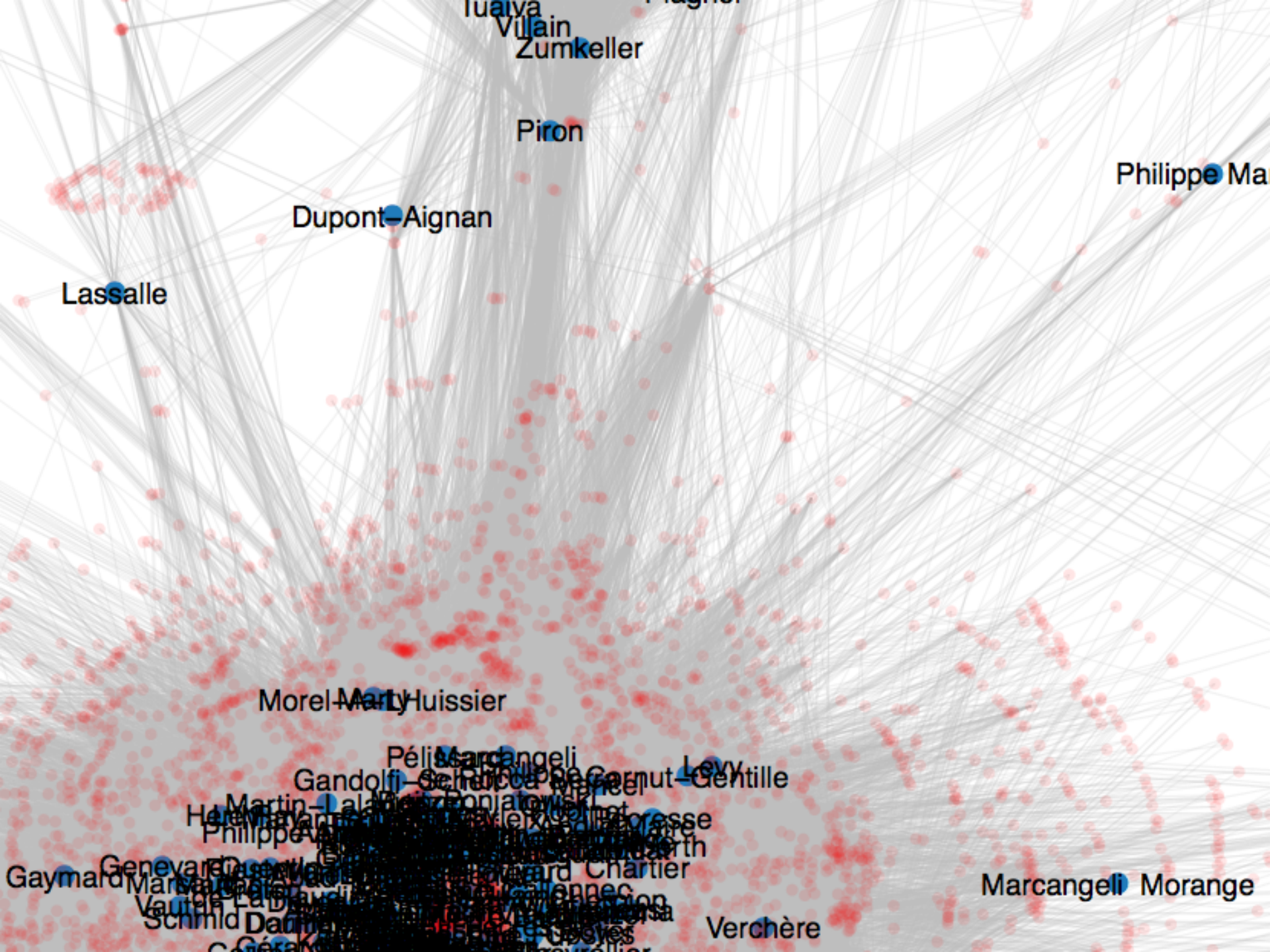
Code and data:

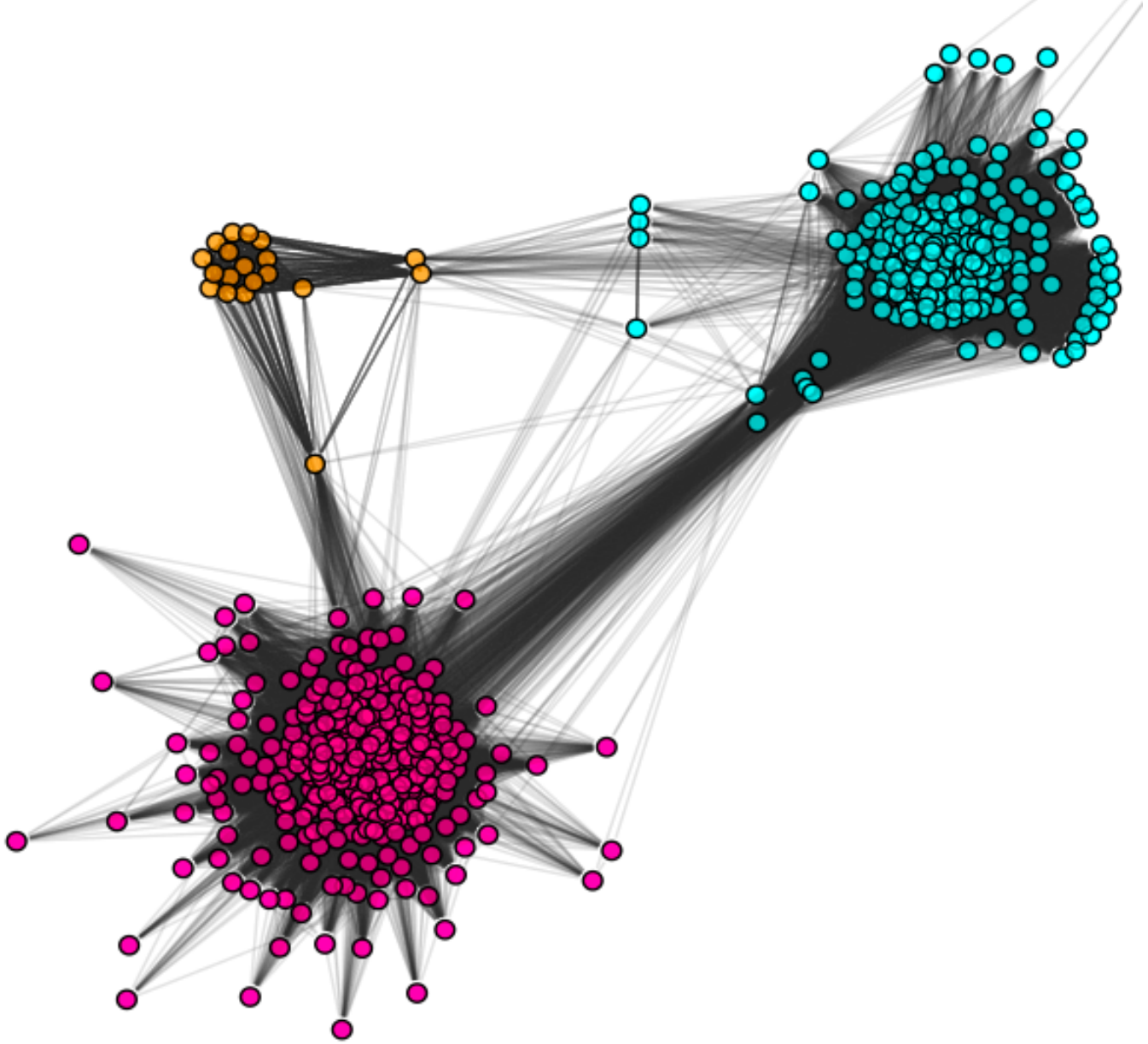
- github.com/briatte/ggnet
- github.com/briatte/neta
- nosdeputes.fr (Regards Citoyens)



Inspiration:
Solomon Messing, “Working with Bipartite/Affiliation Network Data in R”


Pélissard
Morange
Françoise Dubois
Bapt
Bricout
Biémouret
Boisserie
Hoffman-Rispal


Tuaiva
Villain
Zumkeller
Piron
Philippe Ma
Dupont-Aignan
Lassalle
Morel-Marly
Huissier
Gandolfi
Levy
Gentille
Martin
Chartier
Gaymard
Marcangeli
Morange
Verchère



Source:
Baptiste Coulmont, “Travail de députés”, “Travail de députés (suite)”

1. Packages

```
load.package = function(..., mute = TRUE) {
  sapply(c(...), function(x) {
    if(!require(x, quietly = mute, character.only = TRUE))
      install.packages(x, quiet = mute)
    library(x, character.only = TRUE, quietly = mute)
  })
}

# packages: viz
load.package("GGally", "RColorBrewer")

# packages: networks
load.package("intergraph", "sna")
```

2. Data

```
read.tsv = function(x, url = "https://raw.github.com/briatte/ggnet/master/") {
  if(!require(downloader)) install.packages("downloader")
  if(!file.exists(x)) downloader::download(paste0(url, x), x, mode = "wb")
  return(read.csv(x, sep = "\t"))
}

# data: MPs
ids = read.tsv("nodes.tsv")
names(ids)

# data: Twitter
df = read.tsv("network.tsv")
names(df)
```

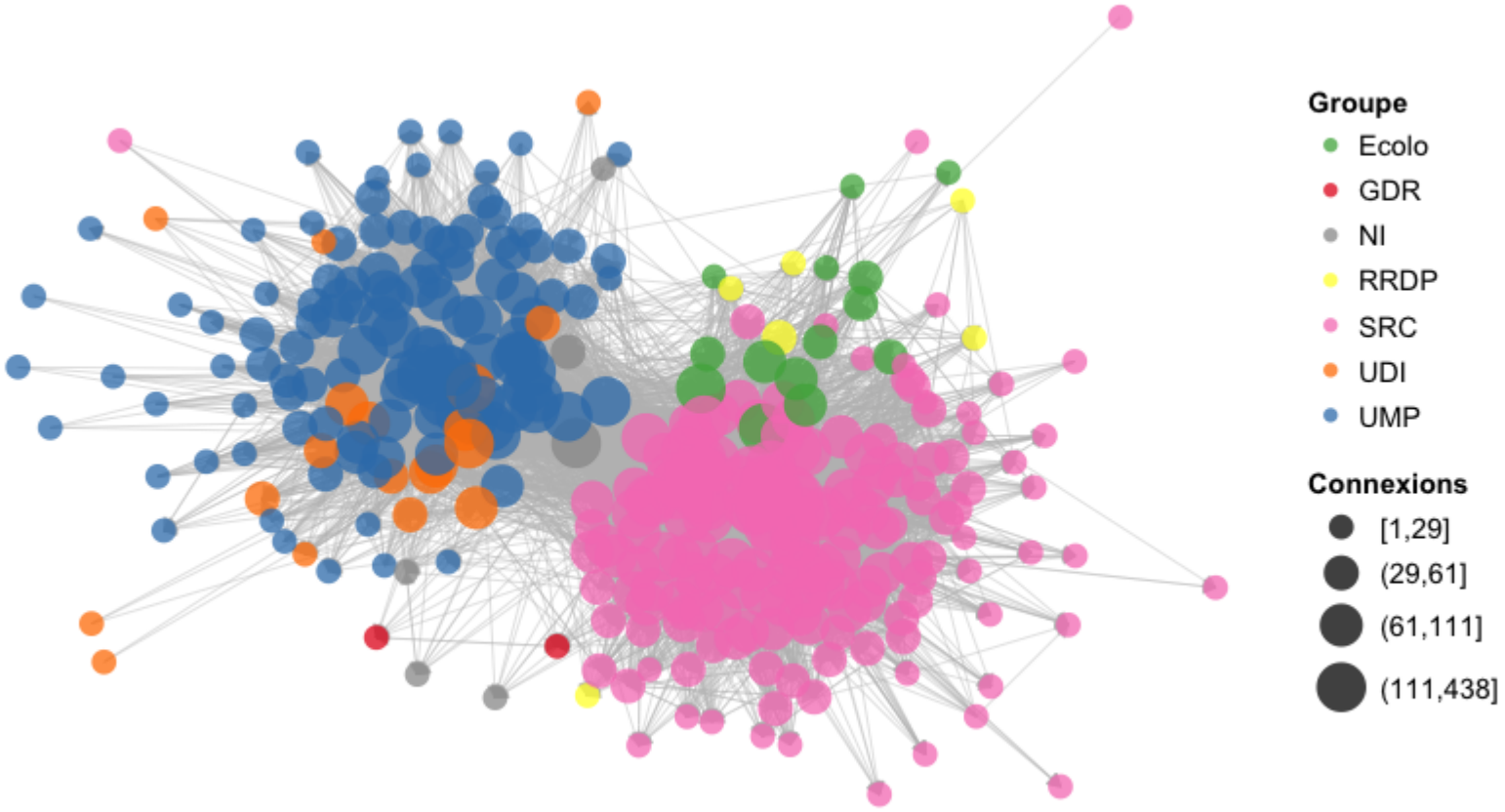
3. Plot

```
# build: network
net = network::network(df)
mps = data.frame(Twitter = network::network.vertex.names(net))

# build: colours
mp.groups = merge(mps, ids, by = "Twitter")$Groupe
mp.colors = RColorBrewer::brewer.pal(9, "Set1")[c(3, 1, 9, 6, 8, 5, 2)]

# plot: network
ggnet(net,
      weight = "degree",
      quantize = TRUE,
      node.group = mp.groups,
      node.color = mp.colors)
```

```
# plot: network
ggnet(net,
      weight = "degree",        # inlinks + outlinks
      quantize = TRUE,          # weight by quartile
      node.group = mp.groups,   # assign node groups
      node.color = mp.colors)   # assign node colors
```



Questions?

details ↓

Arguments

```
# network

ggnet(
  net,                                    # an object of class network
  mode              = "fruchtermanreingold", # a placement algorithm name

# nodes

weight.method    = "none",
quantize.weights = FALSE,     # break weights to quartiles
subset.threshold = 0,         # exclude x ≥ weighted nodes
```

Internals

```
# arguments include a network and a force-based method

ggnet(net, mode = "fruchtermanreingold", size = 12,
    alpha = 0.75, weight.method = "none", ...

# extract x-y coordinates from the placement algorithm

placement <- paste0("gplot.layout.", mode)
if(!exists(placement)) stop("Unsupported placement method.")

# paste edge list attributes from the adjacency matrix

edglist <- as.matrix.network.edgelist(net)
edges   <- data.frame(plotcord[edglist[, 1], ], plotcord[edglist[, 2], ])
```

Aesthetics

```
# size

size              = 12,       # node size
segment.size      = .25,      # set to 0 to remove from plot
arrow.size        = 0,        # set to 0 to remove from plot

# alpha

alpha             = .75,      # node transparency
node.alpha        = NULL,     # transparency for nodes (inherits from alpha)
segment.alpha     = NULL,     # transparency for links (inherits from alpha)

# color

node.group        = NULL,     # what to color the nodes with
node.color        = NULL,     # what colors to use for the node classes
segment.color     = "grey",   # default links are rgb(190, 190, 190)
```

Advanced

```
# tweak labels

label.nodes      = FALSE,     # add vertex names in small print
trim.labels      = TRUE,      # clean vertex names
...                           # passed to geom_text for node labels

# tweak legend

legend.position  = "right",   # set to "none" to remove from plot
names            = c("", ""), # what to call the node color and node weight
legends
top8.nodes       = FALSE,     # color the top 8 nodes with ColorBrewer Set1
```

they see me trollin ↓

f.briatte@ed.ac.uk