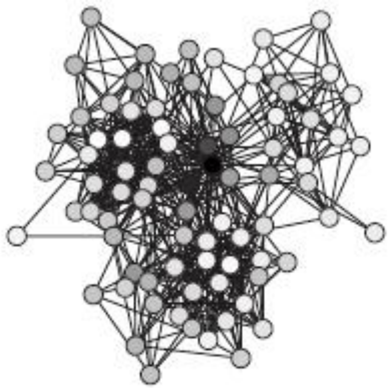
Analyse des données d’échange

1ère partie : statistiques résumées

Plan

0. Jeu de données en exemple “UKfaculty”
1. Les “statistiques résumées”
2. Degrés, degrés entrants, degrés sortants
3. Centralité
4. Modules et modularité
5. Comparaison de classifications
6. Approche RDPG

Jeu de données UKfaculty



PHYSICAL REVIEW E **77**, 016107 (2008)

Fuzzy communities and the concept of bridgeness in complex networks

Tamás Nepusz*
Department of Measurement and Information Systems, Budapest University of Technology and Economics, P. O. Box 91, H-1521 Budapest, Hungary

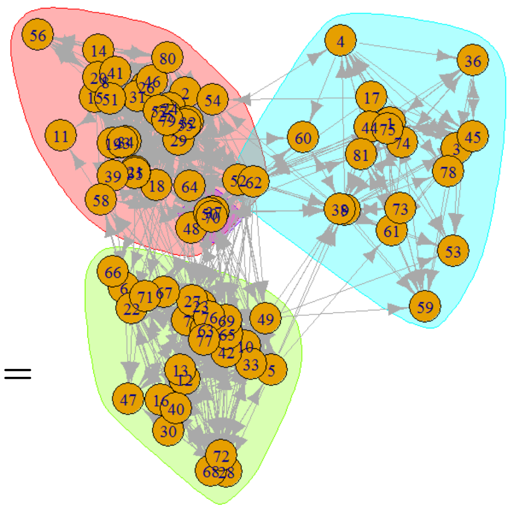
Andrea Petróczi
School of Life Sciences, Kingston University, Kingston-upon-Thames, Surrey, KT1 2EE, United Kingdom

László Négyessy
Neurobionics Research Group, Hungarian Academy of Sciences–Péter Pázmány Catholic University–Semmelweis University, Tűzoltó Utca 58, H-1094 Budapest, Hungary

Fülöp Bazsó[†]
Department of Biophysics, KFKI Research Institute for Particle and Nuclear Physics of the Hungarian Academy of Sciences, P. O. Box 49, H-1525 Budapest, Hungary

- réseau d'amitiés du personnel d'une université britannique
- 81 nœuds (personnes)
- 817 liens dirigés et pondérés
- attributs des nœuds : affiliation (école), 4 catégories

```
library(igraphdata) #pour le jeu de données
library(igraph) #pour les fonctions sur les graphes

###obtention des données et génération de trois variables quantitatives aléatoires###
data(UKfaculty)
set.seed(10)
V(UKfaculty)$var1<-runif(81)
V(UKfaculty)$var2<-rnorm(81,V(UKfaculty)$Group,1)
V(UKfaculty)$var3<-rnorm(81,degree(UKfaculty),1)


###visualiser le graphe###
group1<-which(V(UKfaculty)$Group==1)
group2<-which(V(UKfaculty)$Group==2)
group3<-which(V(UKfaculty)$Group==3)
group4<-which(V(UKfaculty)$Group==4)
plot(UKfaculty, mark.groups =
list(group1,group2,group3,group4),layout =
layout_with_mds)
```



```
###créer la matrice d'adjacence###
adjacency<-t(as_adj(UKfaculty,sparse=F))

#la version pondérée
weighted.adj<-t(as.matrix(UKfaculty[,]))

binarize<-function(x) ifelse(x==0,0,1)
sum(adjacency-apply(weighted.adj,c(1,2),binarize))
[1] 0

###modèle de configuration###
sample.config<-lapply(1:100,function(x)
sample_degseq(deg.out, deg.in, method =
"simple.no.multiple"))
```

Les statistiques résumées

Idée générale : comprendre les caractéristiques d’un réseau à partir de quelques métriques bien choisies

Quelles métriques ?

choisies sur la base :

- d’un lien **théorique** avec une propriété mathématique d’un modèle dépendant du réseau
- d’un “comportement” **empirique** en adéquation avec la propriété verbale que l’on cherche à qualifier

Peuvent concerner le graphe, les liens ou les nœuds

Statistiques résumées : pour quoi faire ?

Option #1 : stats résumées pour « colorier » les nœuds

Option #2 : stats résumées pour tester une hypothèse

Option #3 : stats résumées régressées sur des variables externes

Degrés

Degré = nombre de connexions d'un nœud

$$d_i = \sum_j a_{ij}$$

Réseaux symétriques => OK

Réseaux dirigés…?

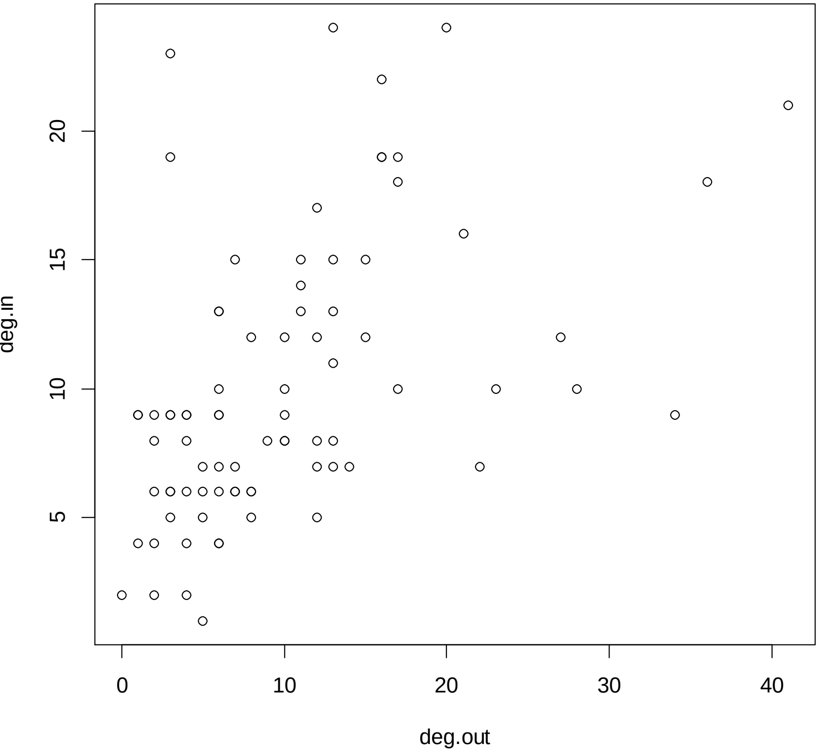
- degré entrant = nombre de liens entrants

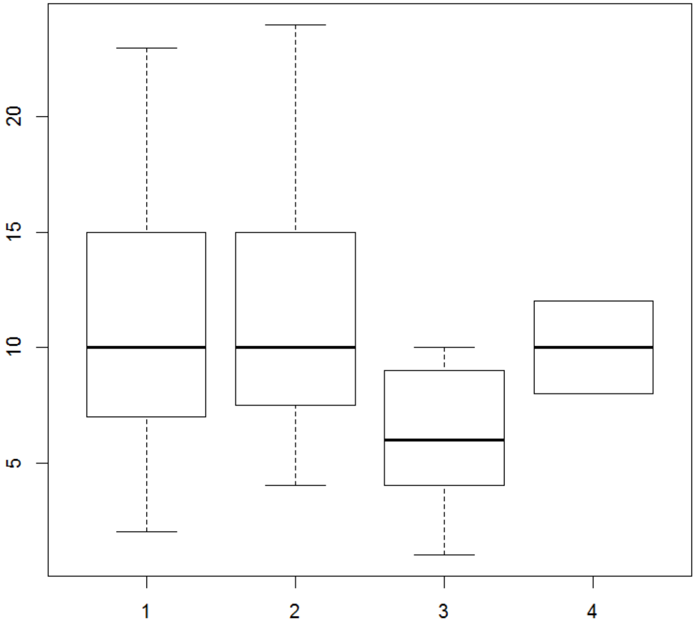
$$d_i^- = \sum_j a_{ij}$$

- degré sortant = nombre de liens sortants

$$d_i^+ = \sum_j a_{ji}$$

```
deg<-degree(UKfaculty)
deg.in<-degree(UKfaculty,mode="in")
deg.out<-degree(UKfaculty,mode="out")

plot(deg.in~deg.out)
```






```
boxplot(deg.in~V(UKfaculty)$Group)

###exercice de régression###
glm(deg.out~V(UKfaculty)$var1+V(UKfaculty)$var2+V(UKfacult
y)$var3,family=poisson())
```

Degrés (suite)

Réseaux pondérés…?

- idem précédemment, mais avec a_{ij} non binaire
- métrique d' (divergence KL, Blüthgen et al. 2006)

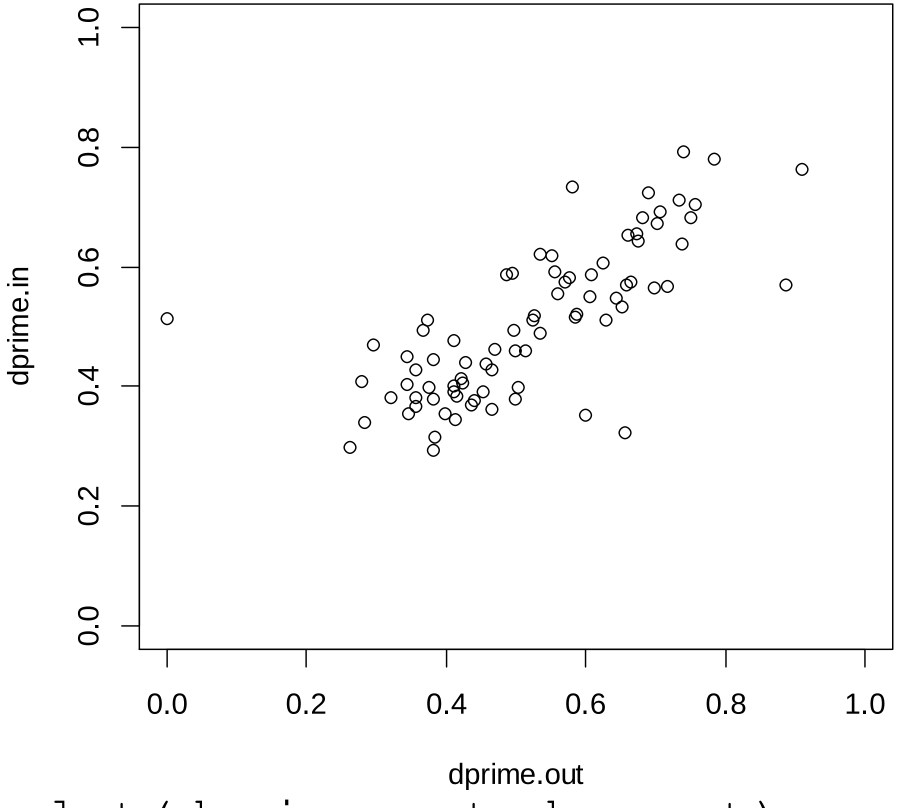
$$d_i = \sum_j \frac{a_{ij}}{A_i} \ln\left[\frac{a_{ij} A}{A_i A_j}\right]$$

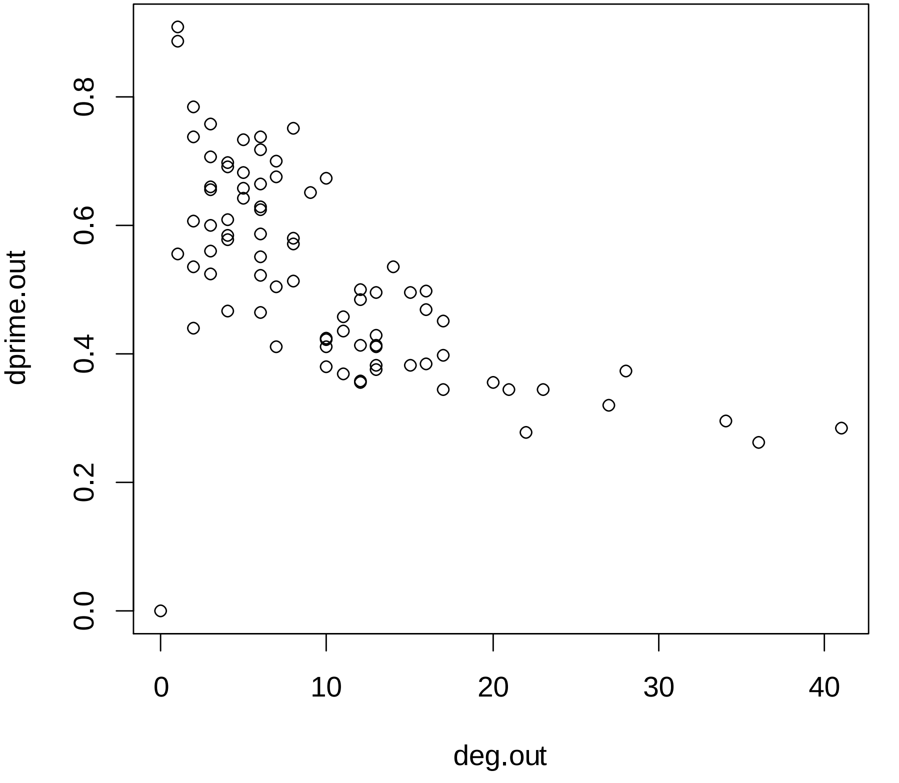
$$d_i' = \frac{d_i - d_{min}}{d_{max} - d_{min}}$$

en théorie, $d_{min} = 0$, et $d_{max} = \ln(A/A_i)$

```
###fonctions utiles pour le calcul de dprime###
pseudolog<-function(x) ifelse(x==0,0,log(x))
notzero<-function(x) ifelse(x==0,1,x)

dfun<-function(mat) {
	n<-dim(mat)[1]
	a<-sum(mat)
	ai<-notzero(apply(mat,1,sum))
	aj<-notzero(apply(mat,2,sum))
	ai_mat<-matrix(rep(ai,n),nrow=n,ncol=n,byrow=F)
	aj_mat<-matrix(rep(aj,n),nrow=n,ncol=n,byrow=T)
	elem<-(mat/ai_mat)*pseudolog(a*mat/(ai_mat*aj_mat))
	d<-apply(elem,1,sum)
	dmax<-pseudolog(a/ai)
	d/dmax
}
```

```
###dprime de Bluethgen###
dprime.in<-dfun(weighted.adj)
dprime.out<-dfun(t(weighted.adj))

plot(dprime.in~dprime.out,xlim=c(0,1),ylim=c(0,1))
```






```
plot(dprime.out~deg.out)
###exercice de régression###
glm(dprime.in~V(UKfaculty)$var1+V(UKfaculty)$var2+V(UKfacu
lty)$var3,family=gaussian())
```

Centralité

Définition classique (eigen-centralité) :

$$c_i = \frac{1}{\lambda} \sum_j a_{ij} c_j$$

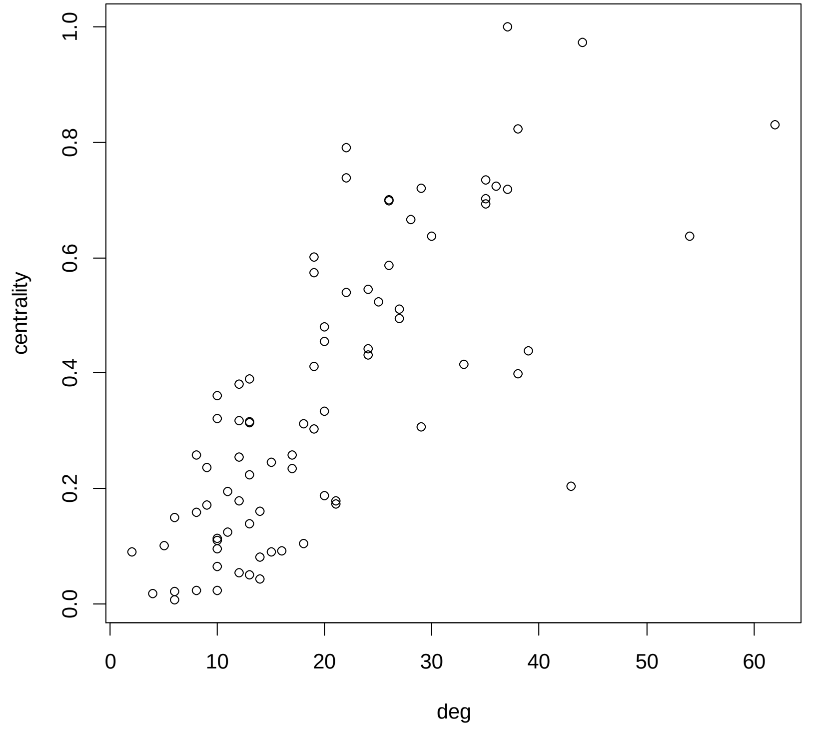
avec λ la valeur propre associée à un vecteur propre positif de A (cf. Perron-Frobenius)

Définition Katz-Bonacich (graphe dirigé) :

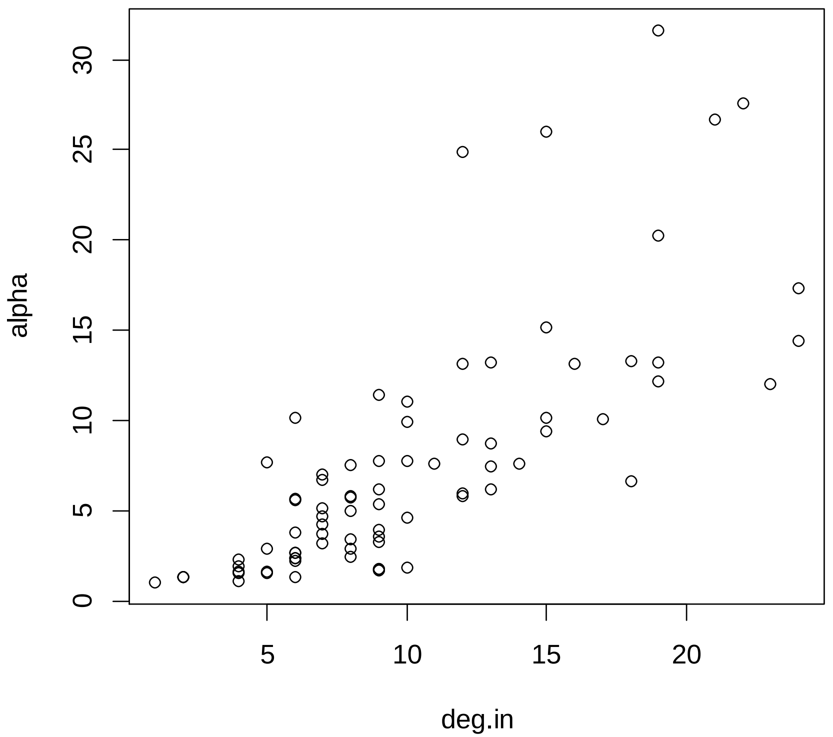
$$c_i = \alpha \sum_j a_{ij} c_j + \varepsilon$$

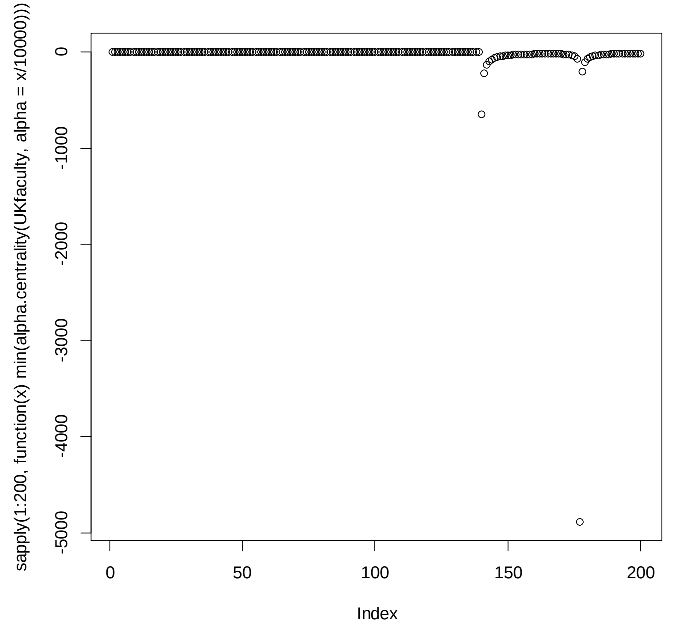
Problème : valeur maximale admissible de α

```
###calcul de centralités###
centrality<-centr_eigen(UKfaculty, directed = TRUE)$vector

plot(centrality~deg)
```



```
alpha<-alpha.centrality(UKfaculty,alpha=0.013)

plot(sapply(1:200, function(x)
min(alpha.centrality(UKfaculty,alpha=x/10000))))
```






```
plot(alpha~deg.in)
glm(centrality~V(UKfaculty)$var1+V(UKfaculty)$var2+V(UKfac
ulty)$var3,family=gaussian())
```

Modules et modularité

Modularité

$$Q = \frac{1}{A}\sum_{i,j}\left[a_{ij} - \frac{d_i d_j}{A}\right]\delta_{ij}$$

Principe : comparer a_{ij} à son « espérance » au vu des degrés, et ne prendre que les éléments de la somme qui correspondent à des paires ne nœuds d’un même groupe

Modules = groupes qui permettent d’obtenir la plus grande valeur de Q

Modules et modularité

Fonctionne pour des graphes non dirigés

Plusieurs algorithmes (edge-betweenness, leading eigenvector, fast greedy…)

Non adapté aux graphes dirigés

- « symétriser » le réseau
- utiliser une autre définition de la recherche de modularité

Modules en réseau dirigé

Méthode `infomap` proposée par Rosvall & Bergstrom (2008)

Maps of random walks on complex networks reveal community structure

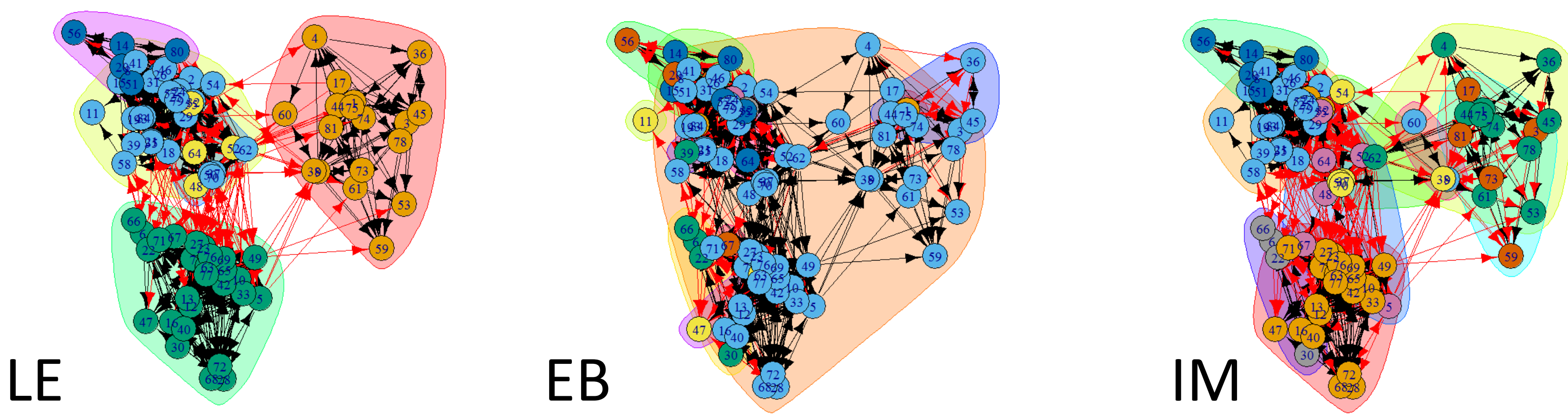
Martin Rosvall*† and Carl T. Bergstrom*‡

Principe : simplifier le codage d'un mouvement Brownien sur le graphe

Un module = un préfixe permettant de simplifier l'information « la particule est dans le module X »

Critère d'optimisation = minimiser le nombre de bits nécessaires pour coder une trajectoire

```
###Modules et modularité###
EB.mod<-cluster_edge_betweenness(UKfaculty) #algorithme edge-betweenness
LE.mod<-cluster_leading_eigen(UKfaculty) #algorithme leading eigenvector
IM.mod<-cluster_infomap(UKfaculty) #algorithme infomap (Rosvall & Bergstrom)
OP.mod<-cluster_optimal(UKfaculty) #algorithme tout-calculer... attention c'est long

plot(LE.mod,UKfaculty,layout = layout_with_mds)
plot(EB.mod,UKfaculty,layout = layout_with_mds)
plot(IM.mod,UKfaculty,layout = layout_with_mds)
```



Comparaison de classifications

Groupement des nœuds en modules = classification

Critères de congruence de classifications (Danon et al. 2005)

Information mutuelle normalisée (NMI) entre classification A (c_A groupes) et classification B (c_B) :

	1	…	c_B	Σ
1	N_{11}		$N_{1,cb}$	$N_{1.}$
…		…		
c_A	$N_{ca,1}$		$N_{ca,cb}$	$N_{ca.}$
Σ	$N_{.1}$		$N_{.cb}$	

$$I(A,B) = \frac{-2\sum_{1\le i\le c_A}\sum_{1\le j\le c_B} N_{ij}\log\left(\frac{N_{ij}N}{N_{i\bullet}N_{\bullet j}}\right)}{\sum_{1\le i\le c_A} N_{i\bullet}\log\left(\frac{N_{i\bullet}}{N}\right)+\sum_{1\le j\le c_B} N_{\bullet j}\log\left(\frac{N_{\bullet j}}{N}\right)}$$

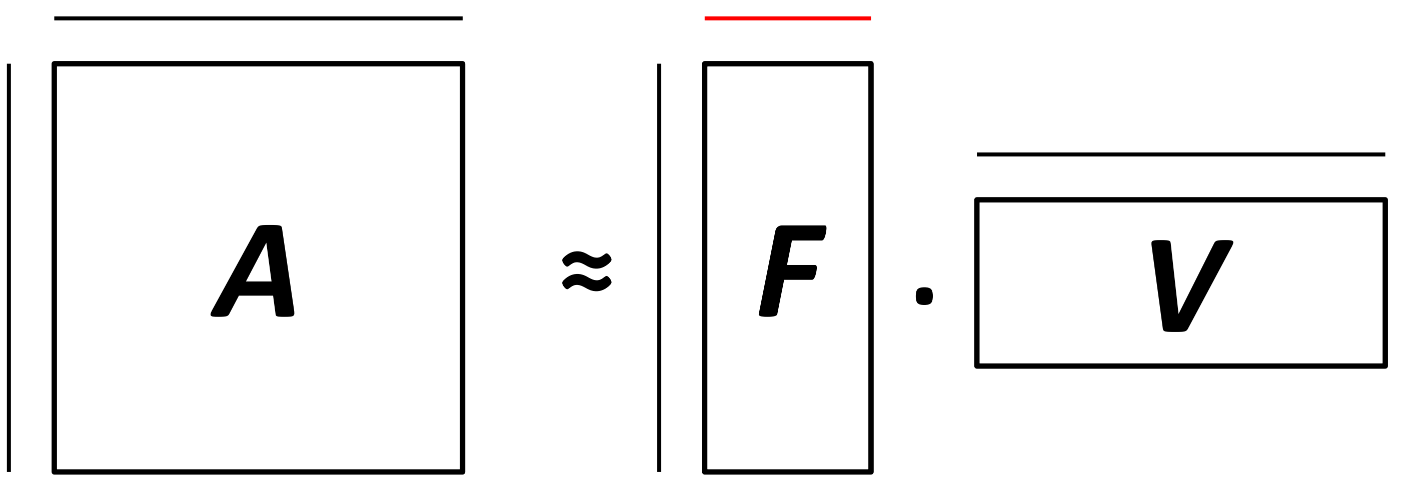
$$0 \le I \le 1$$

indépendance — même classification

```
###Comparaisons de classifications###
compare(IM.mod,LE.mod,method="nmi")
[1] 0.7508369
compare(V(UKfaculty)$Group,LE.mod$mem,method="nmi")
[1] 0.7977774
compare(V(UKfaculty)$Group,OP.mod$mem,method="nmi")
[1] 0.7880018
compare(V(UKfaculty)$Group,IM.mod$mem,method="nmi")
[1] 0.6600338



LE.nmi.config<-sapply(1:100, function(x)
compare(V(UKfaculty)$Group,cluster_leading_eigen(sample.co
nfig[[x]])$mem,method="nmi"))

quantile(LE.nmi.config, probs = c(0.025, 0.975))
 2.5%       97.5%
0.01491554 0.09849240

compare(V(UKfaculty)$Group,cluster_leading_eigen(UKfaculty
,weights=NA),method="nmi")
[1] 0.8386368
```

Approche RDPG : décomposition SVD

Idée : s’affranchir de la dépendance statistique entre liens du réseau en approximant la matrice d’adjacence par un produit scalaire de matrices « de traits »



Philosophie = ACP (diminution du nb de dimensions)

Approche RDPG : décomposition SVD

Décomposition en valeurs singulières

$$\boxed{\boldsymbol{A}} = \boxed{\boldsymbol{L}} \,.\, \boxed{\boldsymbol{S}} \,.\, \boxed{\boldsymbol{R}}$$

avec ***L*** et ***R*** réelles et orthogonales, ***S*** diagonale à valeurs positives ou nulles, ordonnées

Approximation : ne garder que les valeurs de ***S*** suffisamment grandes ; réduire ***L*** et ***R*** de la même façon

$$\boxed{\boldsymbol{A}} \approx \boxed{\acute{\boldsymbol{L}}} \,.\, \boxed{\acute{\boldsymbol{S}}} \,.\, \boxed{\acute{\boldsymbol{R}}}$$

Approche RDPG : décomposition SVD

Racine carrée de ***Ś***

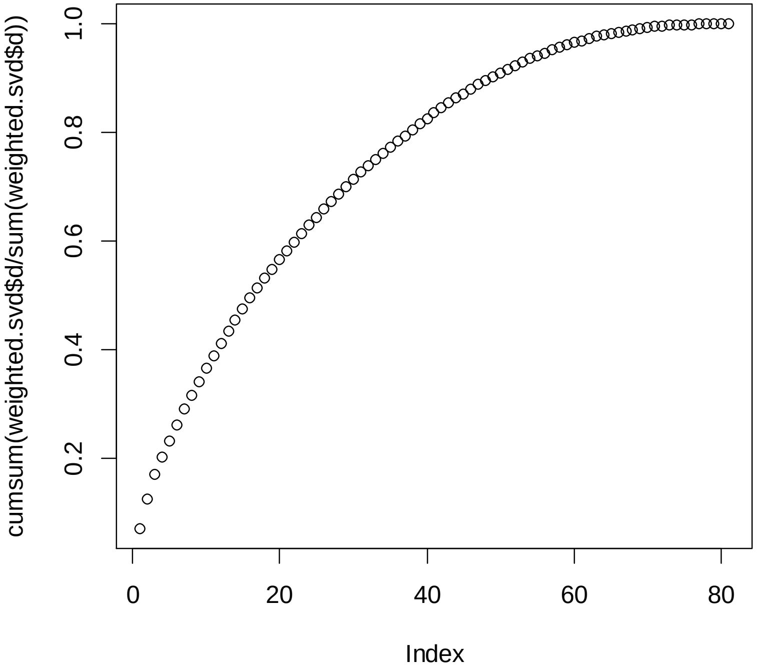
$$\boxed{\acute{S}} = \boxed{\acute{S}}^{1/2} \cdot \boxed{\acute{S}}^{1/2}$$

Et on obtient ***F*** et ***V*** :

$$\boxed{A} \approx \underbrace{\boxed{\acute{L}} \cdot \boxed{\acute{S}}^{1/2}} \cdot \underbrace{\boxed{\acute{S}}^{1/2} \cdot \boxed{\acute{R}}}$$

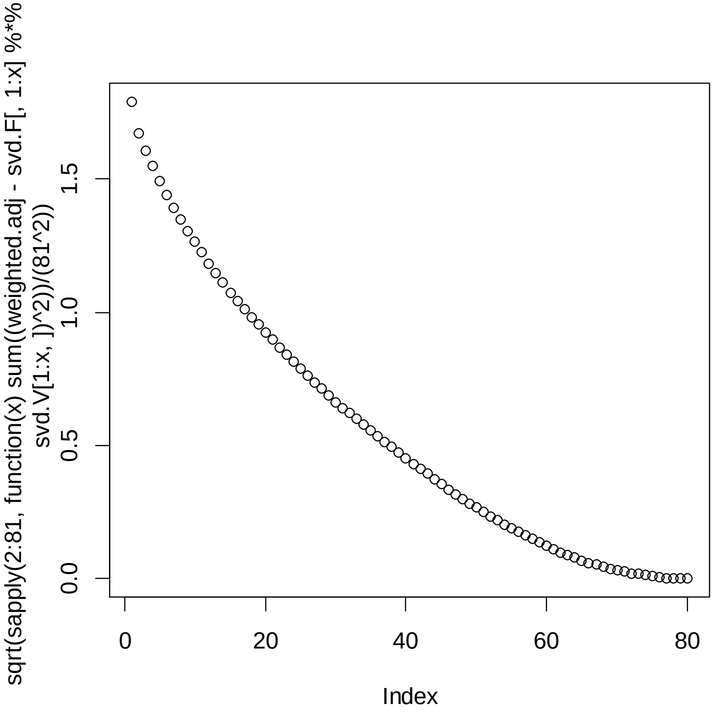
$$\boxed{A} \approx \boxed{F} \cdot \boxed{V}$$

```
###Méthode RDPG###
weighted.svd<-svd(weighted.adj)
svd.L <- weighted.svd$u
svd.R <- weighted.svd$v
svd.S <- diag(weighted.svd$d)
svd.Ssqrt <- structure(vapply(svd.S, sqrt,
numeric(1)),dim=dim(svd.S))

plot(cumsum(weighted.svd$d/sum(weighted.svd$d)))
```




```
cumsum(weighted.svd$d/sum(weighted.svd$d))[40]
[1] 0.8255881
```

```
svd.F <- svd.L %*% svd.Ssqrt
svd.V <- svd.Ssqrt %*% svd.R
plot(sqrt(sapply(2:81,function(x) sum((weighted.adj-
svd.F[,1:x]%*%svd.V[1:x,])^2))/(81^2)))
```

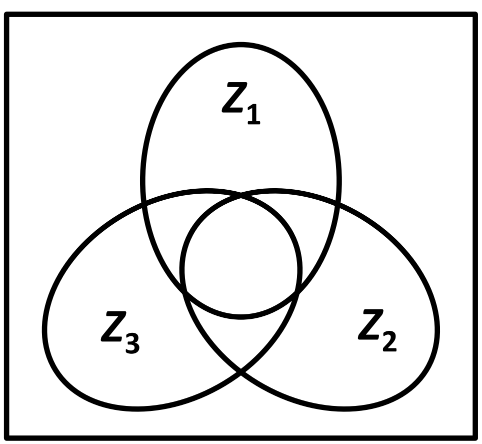



```
sqrt(sum((weighted.adj-
svd.F[,1:60]%*%svd.V[1:60,])^2/(81^2)))
[1] 0.1325902
svd.F <-svd.F[,1:60]
svd.V <-t(svd.V[1:60,])
```

Approche RDPG : analyse de redondance

Analyse de redondance (RDA) = décomposer un tableau via des projections sur les espaces vectoriels d'autres tableaux (facteurs explicatifs)

$$\boxed{\boldsymbol{X}} \sim \boxed{\boldsymbol{Z_1}} + \boxed{\quad\boldsymbol{Z_2}\quad} + \boxed{\boldsymbol{Z_3}}$$

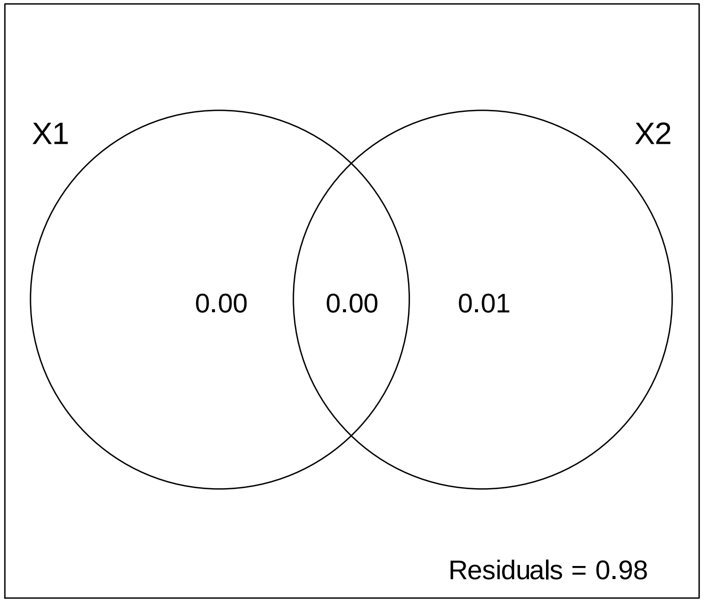


R^2 expliqués par les différentes fractions (ex. $Z_1 \mid Z_2 + Z_3$)

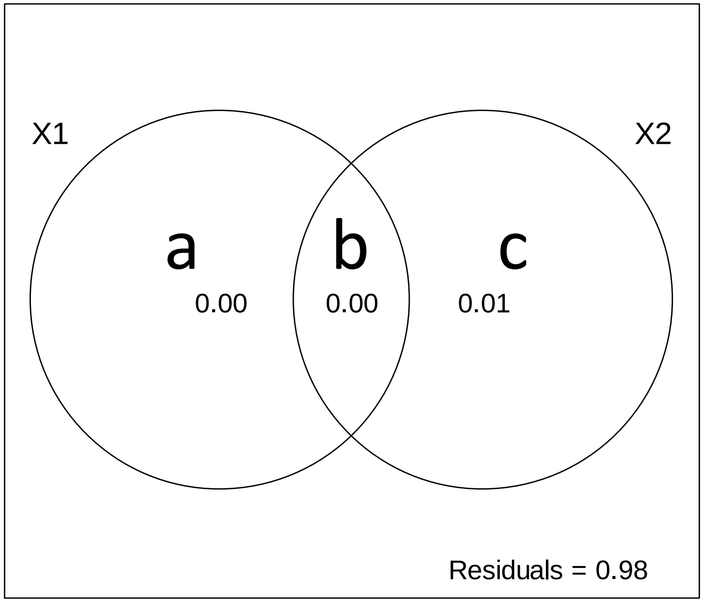
Testables de deux manières :

- permutations des lignes
- reconfiguration du réseau

```
###Analyse de redondance###
library(vegan) #pour les analyses de redondance

dummy<-
as.matrix(cbind(V(UKfaculty)$var1,V(UKfaculty)$var2))
notsodummy<-as.matrix(V(UKfaculty)$var3)

rda.F.all<-rda(svd.F ~ dummy + notsodummy)
vap.F<-varpart(svd.F,dummy, notsodummy)
plot(vap.F)
```



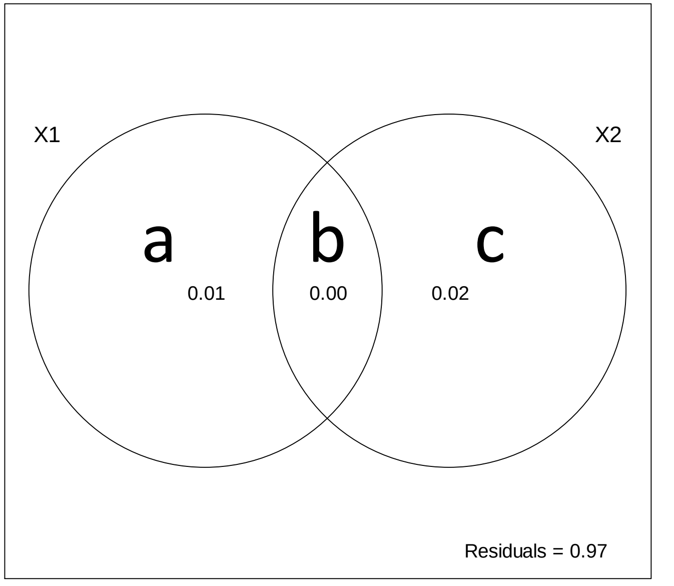
```
anova.F.a<-anova(rda(svd.F ~ dummy
+Condition(notsodummy)),permutations=how(nperm=9999))
anova.F.c<-anova(rda(svd.F ~ notsodummy
+Condition(dummy)),permutations=how(nperm=9999))
anova.F.aplusb<-anova(rda(svd.F ~
dummy),permutations=how(nperm=9999))
anova.F.bplusc<-anova(rda(svd.F ~
notsodummy),permutations=how(nperm=9999))
anova.F.aplusbplusc<-anova(rda(svd.F ~ dummy +
notsodummy),permutations=how(nperm=9999))

anova.F.a$Pr
[1] 0.1077     NA
anova.F.c$Pr
[1] 2e-04    NA
anova.F.aplusb$Pr
[1] 0.0362     NA
anova.F.bplusc$Pr
[1] 1e-04    NA
anova.F.aplusbplusc$Pr
[1] 1e-04    NA
```



```
rda.V.all<-rda(svd.V ~ dummy + notsodummy)
vap.V<-varpart(svd.V,dummy, notsodummy)
plot(vap.V)
```


```
anova.V.a<-anova(rda(svd.V ~ dummy
+Condition(notsodummy)),permutations=how(nperm=9999))
anova.V.c<-anova(rda(svd.V ~ notsodummy
+Condition(dummy)),permutations=how(nperm=9999))
anova.V.aplusb<-anova(rda(svd.V ~
dummy),permutations=how(nperm=9999))
anova.V.bplusc<-anova(rda(svd.V ~
notsodummy),permutations=how(nperm=9999))
anova.V.aplusbplusc<-anova(rda(svd.V ~ dummy +
notsodummy),permutations=how(nperm=9999))
```

```
anova.V.a$Pr
[1] 0.019    NA
anova.V.c$Pr
[1] 1e-04    NA
anova.V.aplusb$Pr
[1] 0.0071     NA
anova.V.bplusc$Pr
[1] 1e-04    NA
anova.V.aplusbplusc$Pr
[1] 1e-04    NA
```



```
analysis.function<-function(gr,nb,var1,var2){
adj<-t(as.matrix(gr[,]))
gr.svd<-svd(adj)
svd.L <- gr.svd$u
svd.R <- t(gr.svd$v)
svd.S <- diag(gr.svd$d)
svd.Ssqrt <- structure(vapply(svd.S, sqrt,
numeric(1)),dim=dim(svd.S))
svd.F <- svd.L %*% svd.Ssqrt
svd.V <- svd.Ssqrt %*% svd.R
svd.F <-svd.F[,1:nb]
svd.V <-t(svd.V[1:nb,])
c(varpart(svd.F,var1,var2)$part$indfract[["Adj.R.squared"]],varpa
rt(svd.V,var1,var2)$part$indfract[["Adj.R.squared"]])
}

analysis.function(UKfaculty,60,dummy,notsodummy)
[1] 0.003243487 0.001695240 0.012543866 0.982517407 0.005806808
[6] 0.001485421 0.022470097 0.970237675
```

```
varpart.config<-sapply(1:100,function(x)
analysis.function(sample.config[[x]],60,dummy,notsodummy))

quantiles.fraction<-apply(varpart.config,1,function(x)
quantile(x,c(0.025,0.5,0.975)))

quantiles.fraction
              [,1]          [,2]        [,3]      [,4]          [,5]
2.5%  -0.005687494 -0.0002258130 0.006044904 0.9877393 -0.0033750331
50%   -0.001850077  0.0007672433 0.009045832 0.9920442  0.0007527783
97.5%  0.001530951  0.0017993537 0.011709092 0.9961728  0.0037415060
              [,6]       [,7]      [,8]
2.5%  -0.0002847333 0.02030196 0.9727478
50%    0.0007518207 0.02206201 0.9766004
97.5%  0.0016056255 0.02380491 0.9815032

analysis.function(UKfaculty,60,dummy,notsodummy)
[1] 0.003243487 0.001695240 0.012543866 0.982517407 0.005806808
[6] 0.001485421 0.022470097 0.970237675
```

Références

Blüthgen, N., Menzel, F. & Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC ecology,* **6,** 9.

Dalla Riva, G. V. & Stouffer, D. B. (2016) Exploring the evolutionary signature of food webs' backbones using functional traits. *Oikos,* **125**, 446-456.

Danon, L., Díaz-Guilera, A., Duch, J. & Arenas, A. (2005) Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment,* **2005**, P09008.

Kolaczyk, E. D. & Csárdi, G. (2014) *Statistical analysis of network data with R,* Springer.

Nepusz, T., Petróczi, A., Négyessy, L. & Bazsó, F. (2008) Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E,* **77,** 016107.

Newman, M. E. J. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences,* **103**, 8577-8582.

Rosvall, M. & Bergstrom, C. T. (2008) Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences,* **105**, 1118-1123.