

Advanced metrics on networks & null models

François Massol

Outline

1. Advanced metrics

1. degrees
2. trophic levels
3. modularity & clustering
4. generalism, network specialization, nestedness
5. robustness to species removal
6. β diversity of networks

2. Models

1. de novo generation of networks
2. null models

Preamble

Check installation of the following packages:

- bipartite
- igraph
- Fwebs
- vegan
- cheddar
- matlib
- sbm
- alluvial
- econetwork
- poweRlaw

Open the script
`script_metricsnull.R`
and set the `setwd` line to your liking

```
#check Rtools installation if
issues appear)

devtools::install_github("FMest
re1/fw_package")
```

Preamble

Load a bucketful of functions and libraries

Load the example food web data from Cruz-Escalona, V. H., Arreguín-Sánchez, F. & Zetina-Rejón, M. (2007) Analysis of the ecosystem structure of Laguna Alvarado, western Gulf of Mexico, by means of a mass balance model. *Estuarine, Coastal and Shelf Science*, 72, 155-167.

Load the example plant-poll data from Barrett, S. C. H. & Hellenurum, K. (1987) The reproductive biology of boreal forest herbs. I. Breeding systems and pollination. *Canadian Journal of Botany*, 65, 2036-2046.

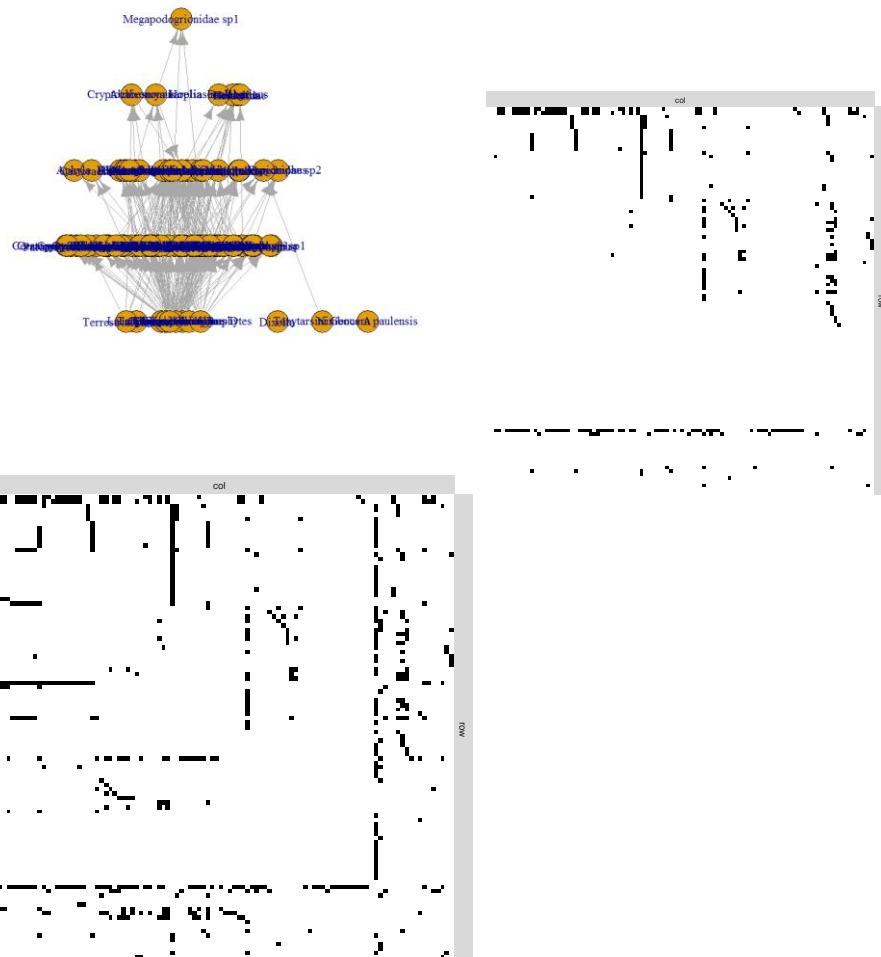
```
source('functions.R')

mat_foodweb<-
t(as.matrix(mg1[[1]][[223]]))
rownames(mat_foodweb)<-
names(mg1[[1]][[223]])
colnames(mat_foodweb)<-
names(mg1[[1]][[223]])

mat_plantpol<-barrett1987
mat_plantpol_bin<-mat_plantpol
mat_plantpol_bin[mat_plantpol>0]<-1
```

o. Building graphs

Create an igraph object (directed graph)



```
foodweb<-
graph_from_adjacency_matrix(mat
 _foodweb)
```

```
plot(foodweb, layout=
layout_as_food_web3(foodweb))
```

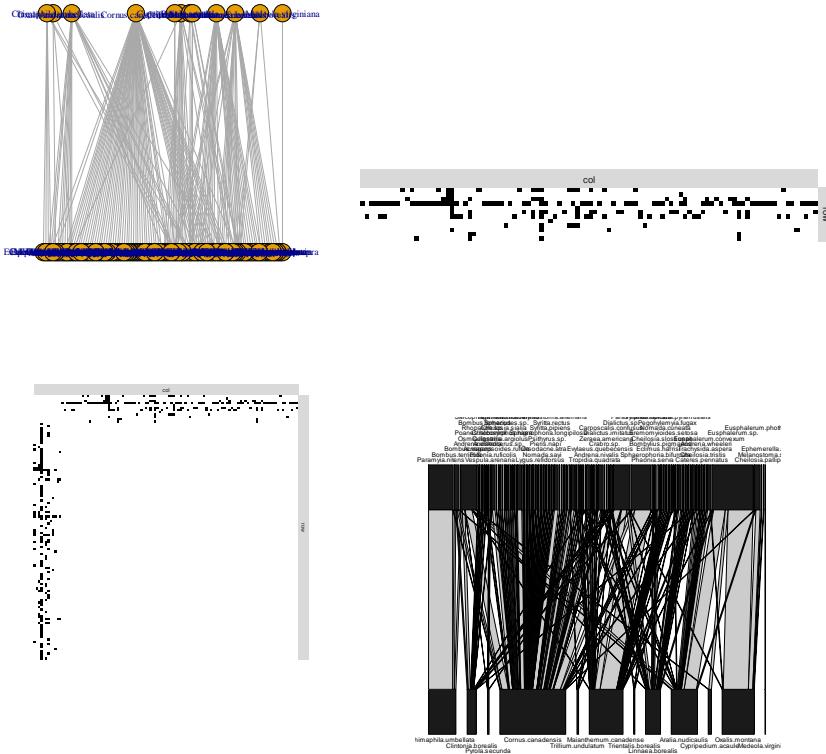
```
plotMyMatrix(as_adj (foodweb, spa
rse=FALSE))
```

```
undirected_foodweb<-
as.undirected(foodweb)
```

```
plotMyMatrix(as_adj (undirected_
foodweb, sparse=FALSE))
```

o. Building graphs

Create (weighted) bipartite graph



Create binary bipartite graph

```
pollination<-
graph_from_bipartite_matrix(m
at_plantpol)
```

```
plot(pollination, layout=
layout_as_bipartite)
```

```
plotMyMatrix(as_adj(pollination
,sparse=FALSE))
```

```
plotMyMatrix(as_bipartite_matrix(pollination,sparse=FALSE))
```

```
pollination_bin<-
graph_from_bipartite_matrix(m
at_plantpol_bin)
```

```
plotweb(mat_plantpol)
```

ADVANCED METRICS

Advanced metrics

Look at the built-in automatic outputs

- from Fwebs

```
> networklevel(mat_plantpol,
> fw.metrics(list(list(mat_foodweb)))
Computing network metrics for food web 1 of 1...
$number_nodes
[1] 106

$number_links
[1] 300

$link_density
[1] 2.830189

$connectance
[1] 0.02695418

$compartmentalization
[1] NaN

$maximum_trophic_level
[1] 3.902716
```

- from bipartite

```
> networklevel(mat_plantpol)
   connectance          web asymmetry      links per species      number of compartments
   0.13643791        0.78947368       1.46491228        2.00000000
compartment diversity    cluster coefficient     modularity.Q      nestedness
   1.09233738        0.08333333       0.56822149       10.26919212
                           NODF           weighted.nestedness  weighted.NODF interaction.strength.asymmetry
   30.78467181        0.66621615       13.49547377        0.23565934
specialisation asymmetry linkage density      weighted.connectance Fisher.alpha
   -0.20755966        9.29902902       0.08157043        81.60982542
Shannon diversity      interaction evenness Alatalo.interaction.evenness H2
   4.11901613        0.57933698       0.38703061        0.57807328
number.of.species.HL  number.of.species.LL mean.number.of.shared.partners.HL mean.number.of.shared.partners.LL
   102.00000000        12.00000000       0.56105611        2.07575758
cluster.coefficient.HL cluster.coefficient.LL weighted.cluster.coefficient.HL weighted.cluster.coefficient.LL
   0.29803030        0.28659537       0.83564121        0.17336473
niche.overlap.HL      niche.overlap.LL      togetherness.HL      togetherness.LL
   0.34762949        0.13855758       0.14499265        0.05157815
C.score.HL            C.score.LL          V.ratio.HL          V.ratio.LL
   0.50388420        0.59112597       1.54171552        27.81509917
discrepancy.HL       discrepancy.LL      extinction.slope.HL extinction.slope.LL
   71.00000000        63.00000000       5.25810274        1.42139458
robustness.HL         robustness.LL      functional.complementarity.HL functional.complementarity.LL
   0.81381191        0.50677215       382.39912254        338.58919107
partner.diversity.HL partner.diversity.LL generality.HL      vulnerability.LL
   0.68675599        2.17835524       2.31225322        16.28580482
```

```
fw.metrics(list(list(mat_foodweb)))
```

```
networklevel(mat_plantpol)
```

Degrees

For binary adjacency matrices A ($=1$ if there is an edge, 0 otherwise), one can compute:

- node i **degree** = total number of connections in an undirected network

$$d_i = \sum_j a_{ij} \quad \text{degree(foodweb)}$$

- node i **in-degree** and **out-degree** = number of connections coming in or going out of node i

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

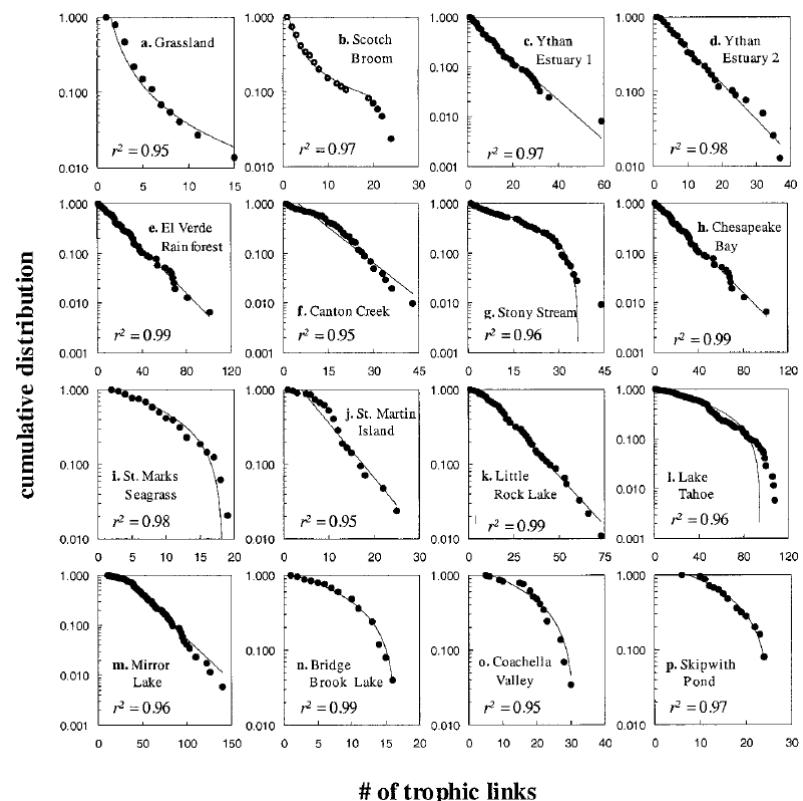
`degree(foodweb, mode="in")`

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

`degree(foodweb, mode="out")`

Degree distributions

1. Can be computed empirically (easily)
2. Networks can be generated from degree sequence (harder)
3. Can be compared to benchmark distributions (power law, Poisson, Erdős-Rényi...)



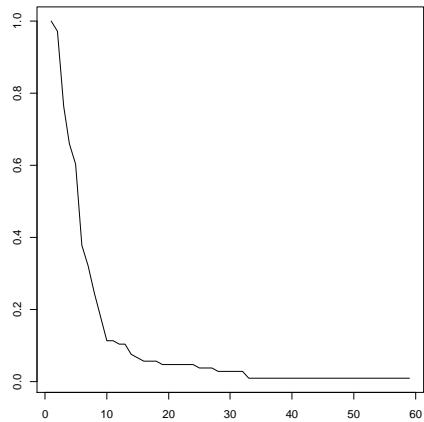
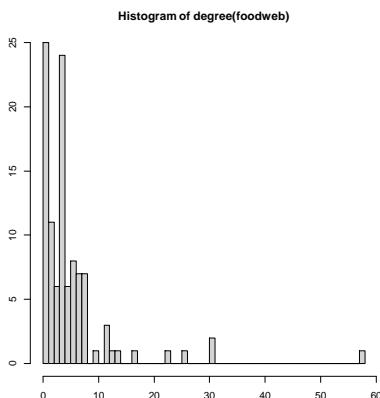
Dunne et al. 2002 PNAS

Degree distributions

The function degree counts both incoming and outgoing links

You can specify if you want just one or the other

There are different ways of plotting the degree distribution



```
degree(foodweb)
```

```
degree(foodweb, mode="in")
```

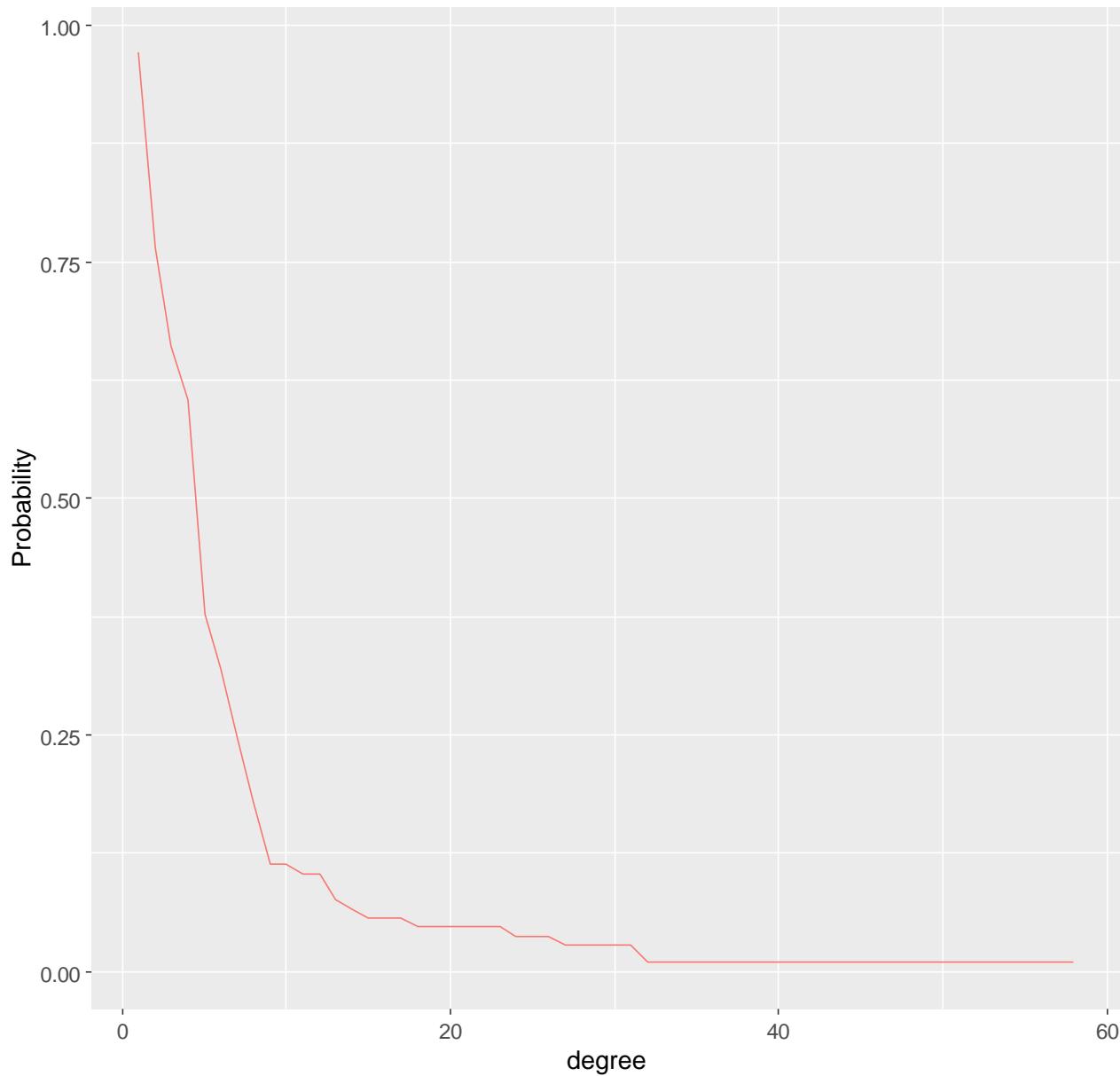
```
degree(foodweb, mode="out")
```

```
hist(degree(foodweb), breaks=0:m  
ax(degree(foodweb)))
```

```
plot(degree_distribution(foodwe  
b, cumulative = TRUE), type="l")
```

```
dd.fw(list(as.data.frame(mat_fo  
odweb)), log = FALSE,  
cumulative = TRUE)
```

Degree distribution



Comparison of degree distributions

Is the degree distribution binomial?

```
ks.test(degree(foodweb), "pbinom"  
       , size=length(V(foodweb)), prob=  
       mean(degree(foodweb))/length(V(  
       foodweb)))
```

Is the degree distribution Poisson?

```
ks.test(degree(foodweb), "ppois"  
       , lambda=mean(degree(foodweb)))
```

Is the degree distribution a negative binomial?

```
ks.test(degree(foodweb), "pnbinom"  
       , mu = mean(degree(foodweb)),  
       size =  
       mean(degree(foodweb))^2/(var(de  
       gree(foodweb))-  
       mean(degree(foodweb))))
```

Degree distributions

Usefulness of power laws in ecology?
few nodes in ecology => impossible to ascertain?

SIAM REVIEW
Vol. 51, No. 4, pp. 661–703

© 2009 Society for Industrial and Applied Mathematics

Power-Law Distributions in Empirical Data*

Aaron Clauset[†]
Cosma Rohilla Shalizi[‡]
M. E. J. Newman[§]

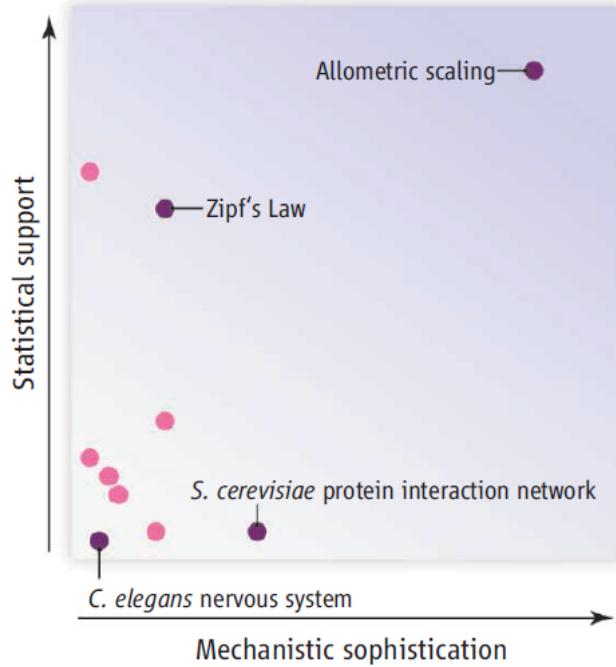
MATHEMATICS

Critical Truths About Power Laws

Michael P. H. Stumpf¹ and Mason A. Porter²

Most reported power laws lack statistical support and mechanistic backing.

Degree distributions



As a rule of thumb, a candidate power law should exhibit an approximately linear relationship on a log-log plot over at least two orders of magnitude in both the x and y axes. This criterion rules out many data sets, including just about all biological networks.

How good is your power law? The chart reflects the level of statistical support—as measured in (16, 21)—and our opinion about the mechanistic sophistication underlying hypothetical generative models for various reported power laws. Some relationships are identified by name; the others reflect the general characteristics of a wide range of reported power laws. Allometric scaling stands out from the other power laws reported for complex systems.

Connectance

Connectance = nb of realized interactions/nb of potential interactions

Several ways of computing the connectance (more or less corrected for impossible links)

```
mean(mat_foodweb)
```

```
DirectedConnectance(as_Community(foodweb))
```

```
mean(as_adj(undirected_foodweb,  
sparse=FALSE))/2
```

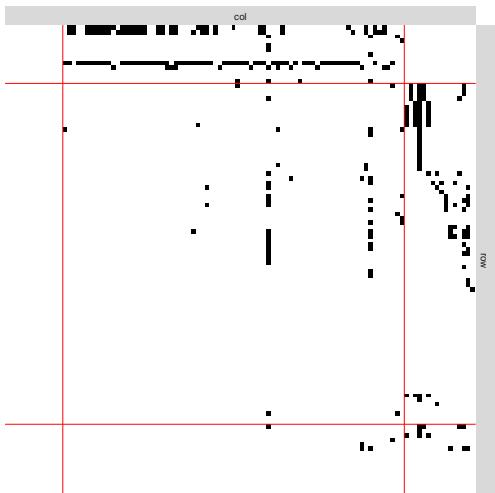
```
mean(as_adj(foodweb,sparse=FALS  
E))
```

```
fw.metrics(list(list(mat_foodwe  
b)))$connectance
```

Trophic levels

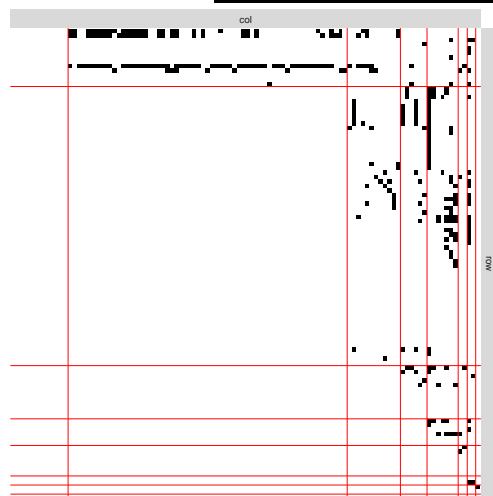
All the classic measures (bottom-up)

Plot the “matrix” by levels



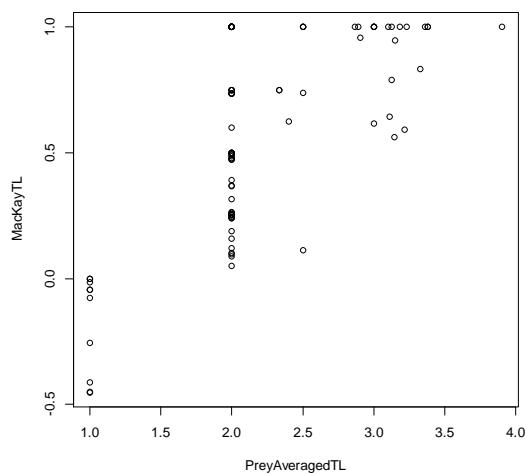
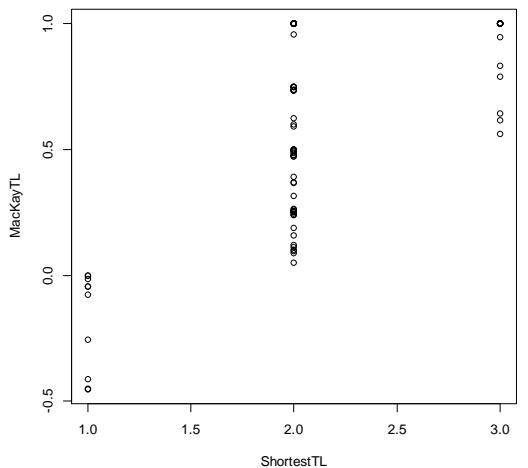
```
foodweb_TL<-
TrophicLevels(as_Community(food
web))
```

```
plotMyMatrix(mat_foodweb,cluste
ring=list("row"=foodweb_TL[,1],
"col"=foodweb_TL[,1]))
plotMyMatrix(mat_foodweb,cluste
ring=list("row"=foodweb_TL[,3],
"col"=foodweb_TL[,3]))
```



Trophic levels

The top-down measure of MacKay et al.
(issue : more than one component...)



```
count_components(foodweb)
```

```
tl.1<-  
trophic_levels(largest_componen  
t(foodweb))
```

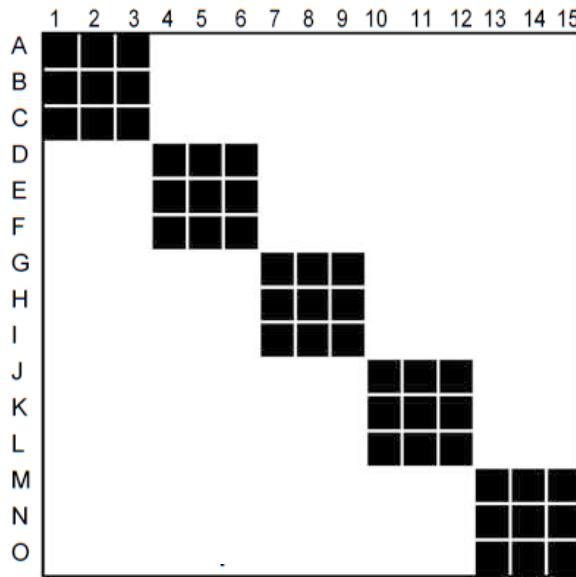
```
plot(TrophicLevels(as_Community  
(largest_component(foodweb), ".  
)[,1], tl.1[,1], xlab="ShortestT  
L", ylab="MacKayTL")
```

```
plot(TrophicLevels(as_Community  
(largest_component(foodweb), ".  
)[,6], tl.1[,1], xlab="PreyAvera  
gedTL", ylab="MacKayTL")
```

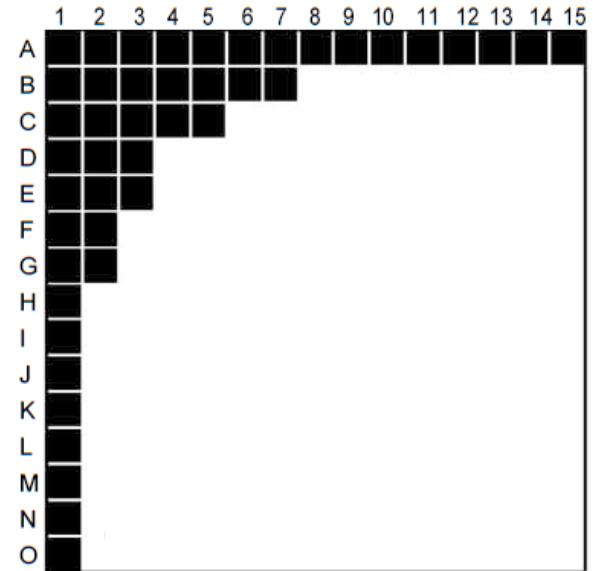
Network structure

Ecological interaction networks might have non-random structure

modular



nested



Modularity

Modularity

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

where δ_{ij} equals 1 if nodes i and j belong to the same module

Principle: compare a_{ij} to its expectation given the degrees d_i , taking only edges within modules (= groups)

Modules = groups that maximize the value of Q

Difficulty: complex problem, several algorithms

Modularity

Works for undirected networks

Not adapted to directed networks

- make the network symmetric or
- use another definition of modularity for directed networks

Three classic algorithms for computing modularity

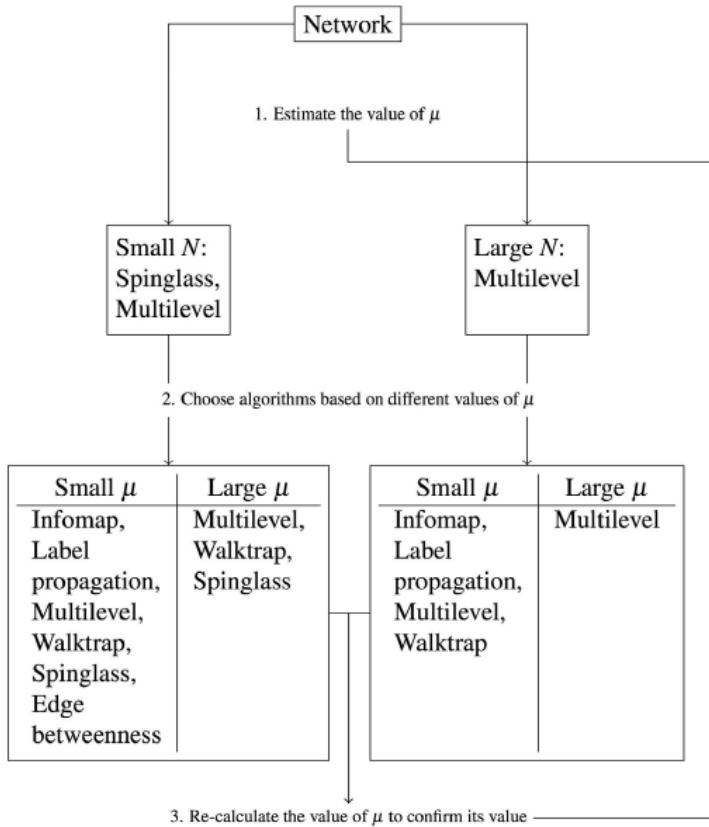
```
foodweb_EB.mod<-
cluster_edge_betweenness(undirected_foodweb)
```

```
foodweb_LE.mod<-
cluster_leading_eigen(undirected_foodweb)
```

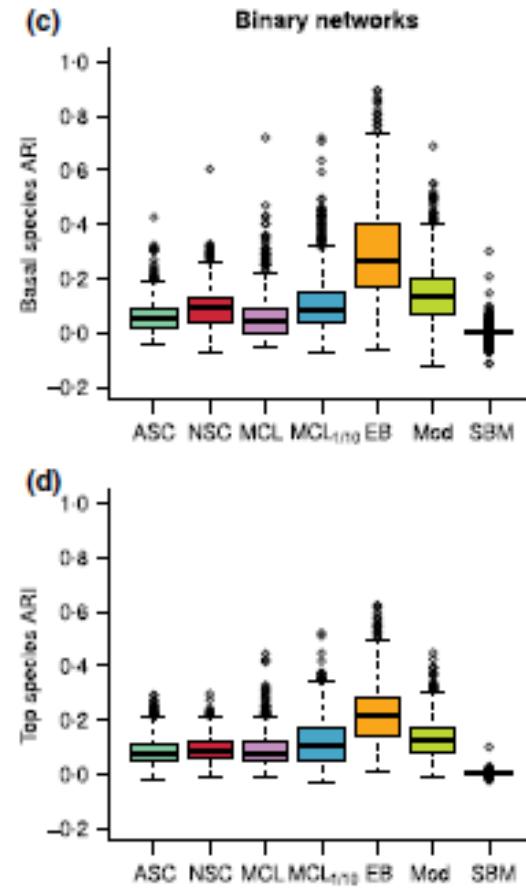
```
foodweb_DL.mod<-
cluster_louvain(undirected_foodweb)
```

Which algorithm?

Unipartite networks

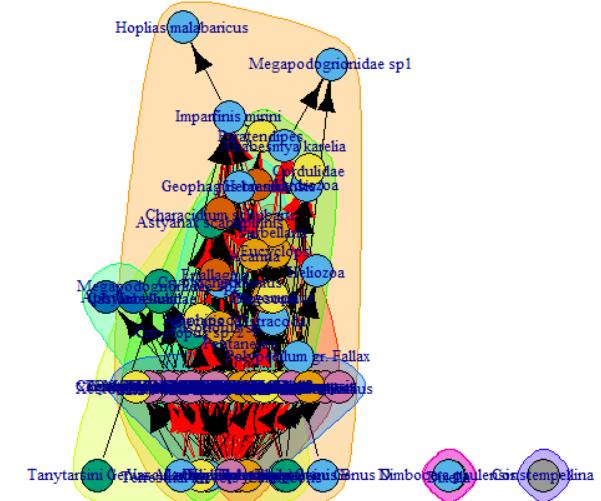
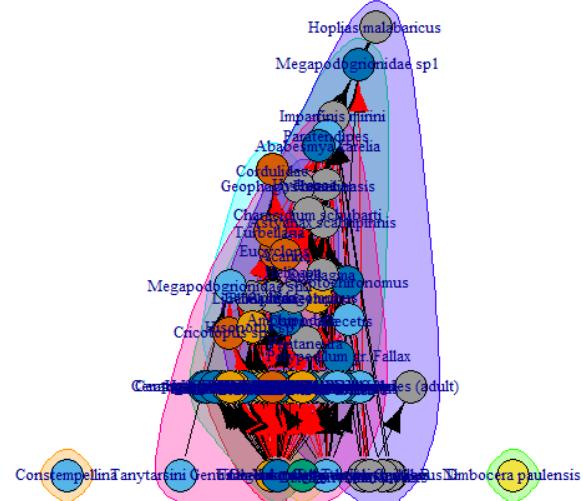
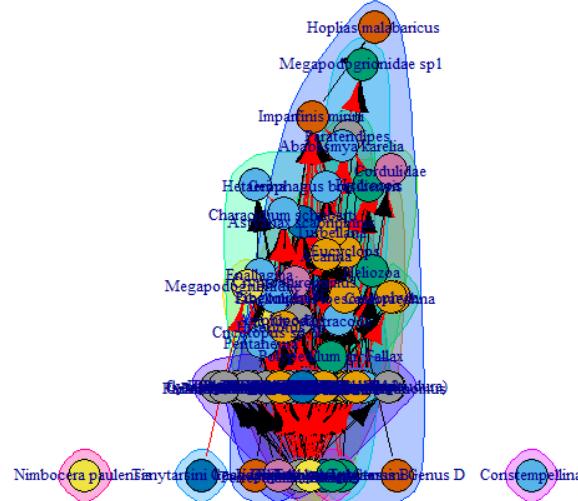


Bipartite networks



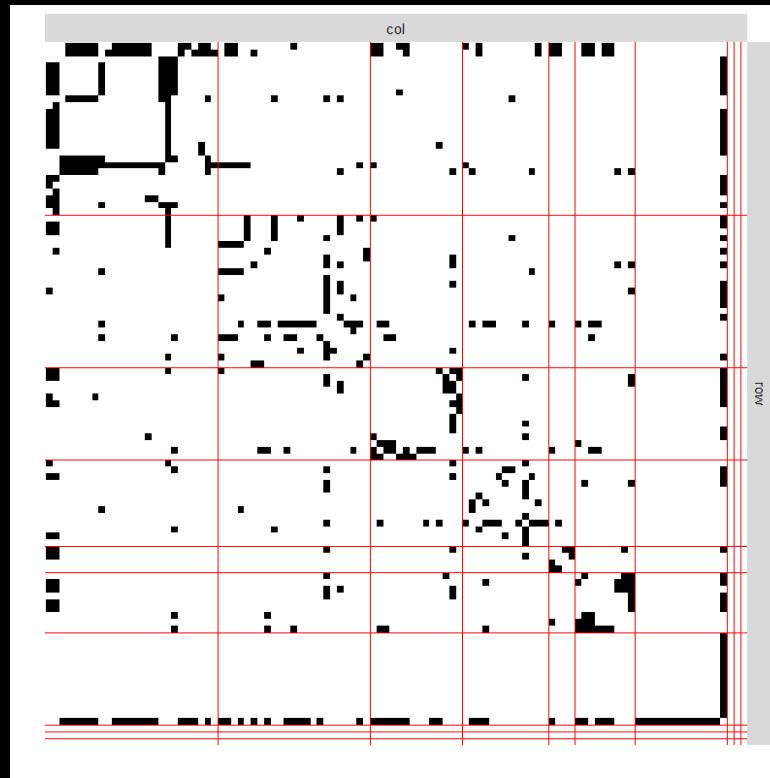
Plotting modularity

```
par(mfrow=c(1,3))  
plot(foodweb_EB.mod, foodweb, layout =  
layout_as_food_web(foodweb))  
plot(foodweb_LE.mod, foodweb, layout =  
layout_as_food_web(foodweb))  
plot(foodweb_ML.mod, foodweb, layout =  
layout_as_food_web(foodweb))
```



Plotting modularity

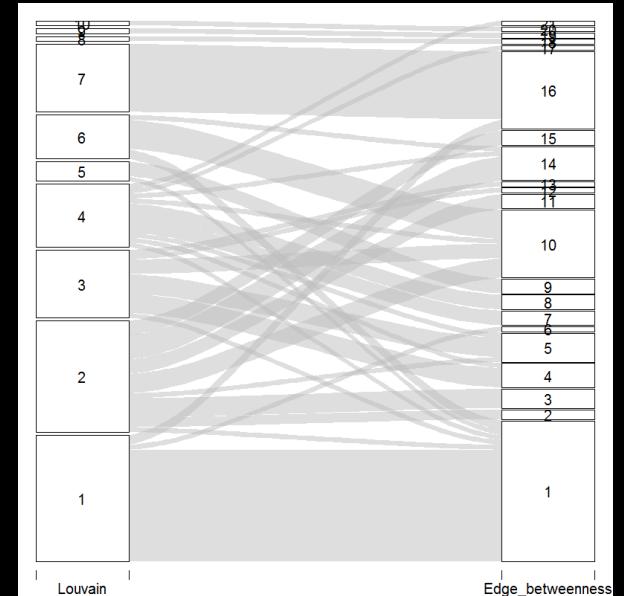
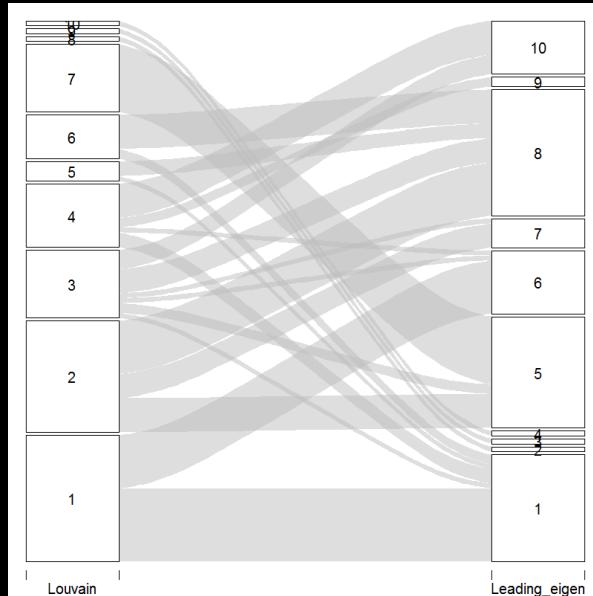
```
plotMyMatrix(as_adj(undirected_foodweb,sparse=F  
ALSE),clustering=list("row"=foodweb_ML.mod$mem  
bership,"col"=foodweb_ML.mod$membership))
```



Comparing modules from the different algorithms

```
make_alluvial_2(foodweb_ML.mod$membership, foodweb_LE.mod$membership, "Louvain", "Leading_eigen")
```

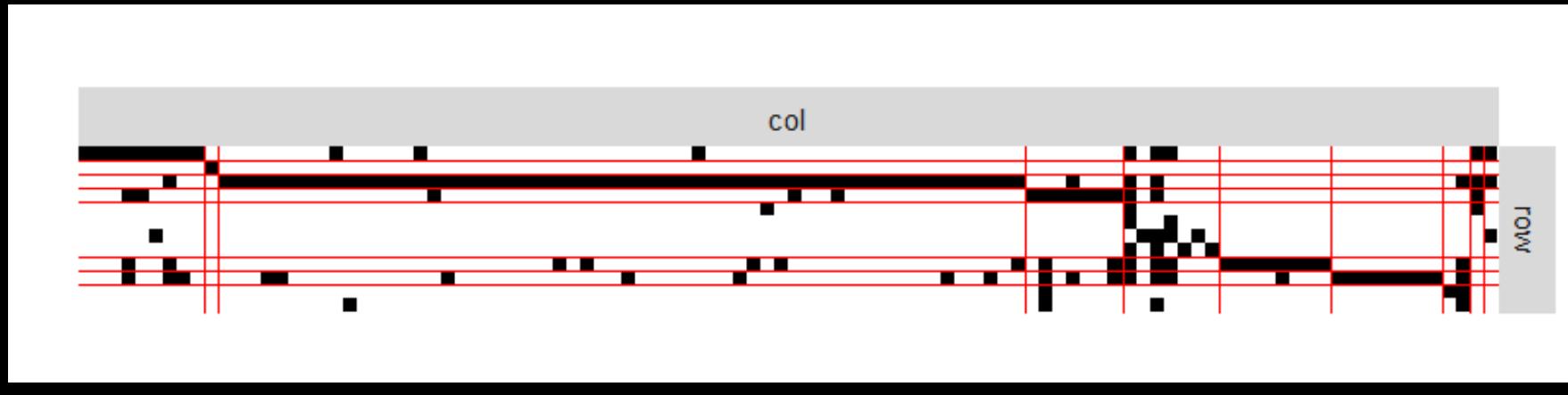
```
make_alluvial_2(foodweb_ML.mod$membership, foodweb_EB.mod$membership, "Louvain", "Edge_betweenness")
```



Modularity on bipartite network

```
pollination_LE.mod<-  
cluster_leading_eigen(pollination_bin)
```

```
plotMyMatrix(mat_plantpol_bin,clustering=list("row"  
=pollination_LE.mod$membership[!V(pollination_bin)$type], "col"  
=pollination_LE.mod$membership[V(pollination_bin)$type]))
```



Spectral clustering

Idea: the Laplacian matrix of the graph, given by

$$L = D - A$$

where D is the diagonal matrix with node degrees and A is the adjacency matrix,

is able to find components of the graph (= parts that are disconnected from one another) as the eigenvectors of eigenvalue 0

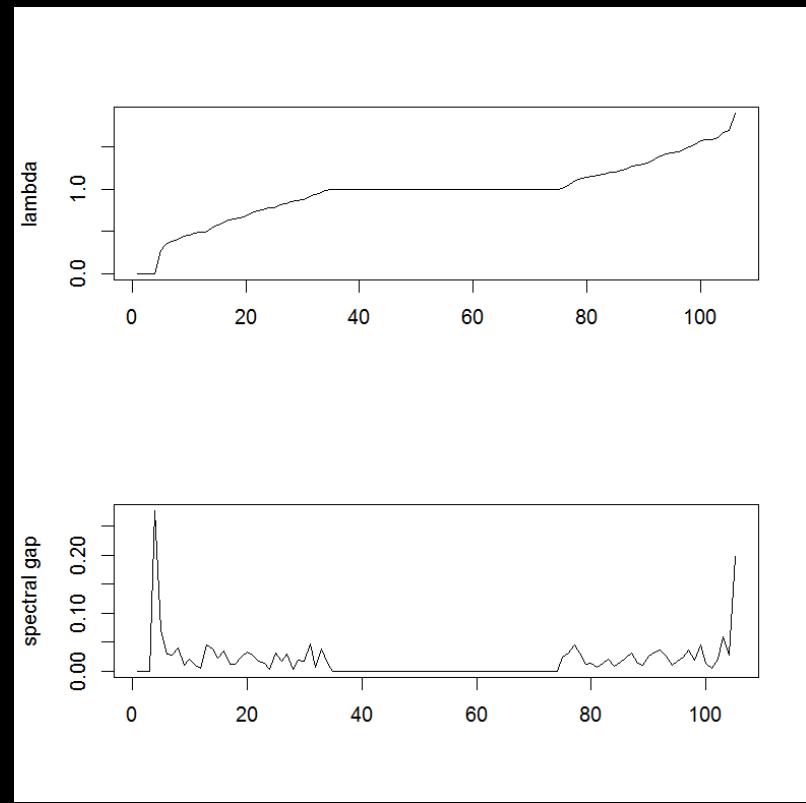
Heuristics: the eigenvectors associated with the low eigenvalues of L characterize the membership of nodes to “clusters” (through K-means algo)

Spectral clustering

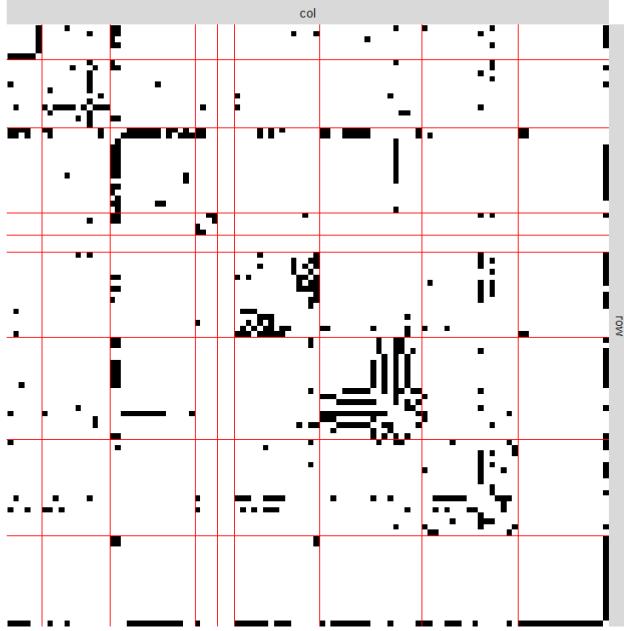
```
foodweb_SG<-  
laplacian_spectral_gap(undirected_foodweb)
```

```
foodweb_SG$optim_n
```

Looks like we need
5 clusters



Spectral clustering



The modularity of the spectral clustering is not so bad either (0.2638) and it's even not the target

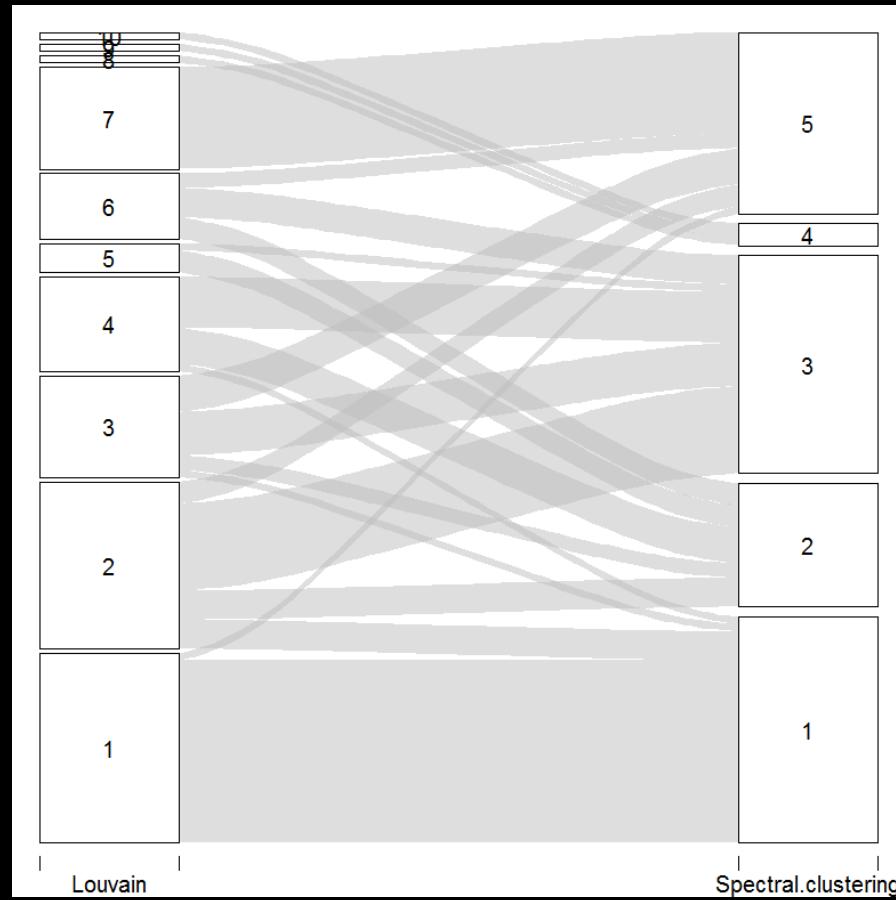
```
foodweb_SC<-
spectral_clustering(undirected_
foodweb,5)

plotMyMatrix(as_adj(undirected_
foodweb,sparse=FALSE),clusterin
g=list("row"=foodweb_SC,"col"=f
oodweb_SC))

modularity(undirected_foodweb,f
oodweb_SC)
```

Comparison between modules and spectral clusters

```
make_alluvial_2(foodweb_ML.mod$membership, foodw  
eb_SC, "Louvain", "Spectral clustering")
```



Node specialization

Metric proposed by Blüthgen et al. (2006) to measure effective specialization in weighted networks (edges have a weight)

$$d_i = \sum_j \frac{w_{ij}}{W_i} \ln \left[\frac{w_{ij} W}{W_i W_j} \right]$$

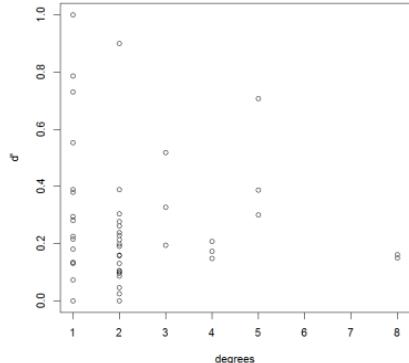
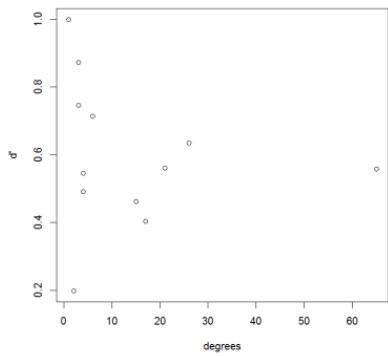
$$d_i' = \frac{d_i - d_{min}}{d_{max} - d_{min}}$$

where w are the edge weights, W are the sums by rows, columns or overall

Node specialization

The dfun function in bipartite does it
(but beware the d_min computation is
not always accurate)

You need to transpose the matrix to
obtain the d' for the other level



```
dfun(mat_plantpol)
```

```
plot(degree(pollination)[!V(pollination_bin)$type],dfun(mat_plantpol)$dprime)
```

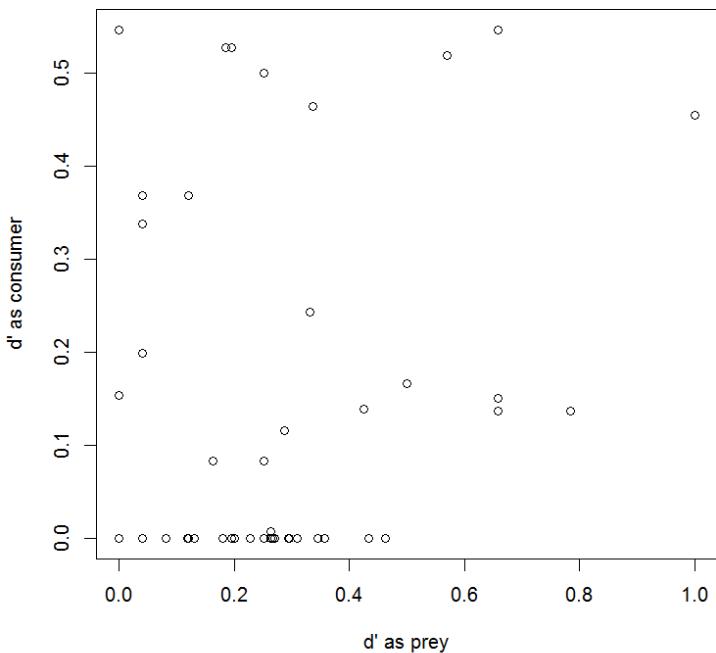
```
dfun(t(mat_plantpol))
```

```
plot(degree(pollination)[V(pollination_bin)$type],dfun(t(mat_plantpol))$dprime)
```

Node specialization

also usable for food webs

Here, let's check the relation between specialization as consumer and specialization as prey



```
foodweb_spe.res<-
dfun(mat_foodweb)$dprime #as
resource
foodweb_spe.con<-
dfun(t(mat_foodweb))$dprime #as
consumer
interm_species_list<-
intersect(names(foodweb_spe.res),
names(foodweb_spe.con))

plot(foodweb_spe.res[which(names(
foodweb_spe.res)%in%interm_sp
ecies_list)], foodweb_spe.con[wh
ich(names(foodweb_spe.con)%in%i
nterm_species_list)])
```

Network specialization

also from Blüthgen et al. (2006)

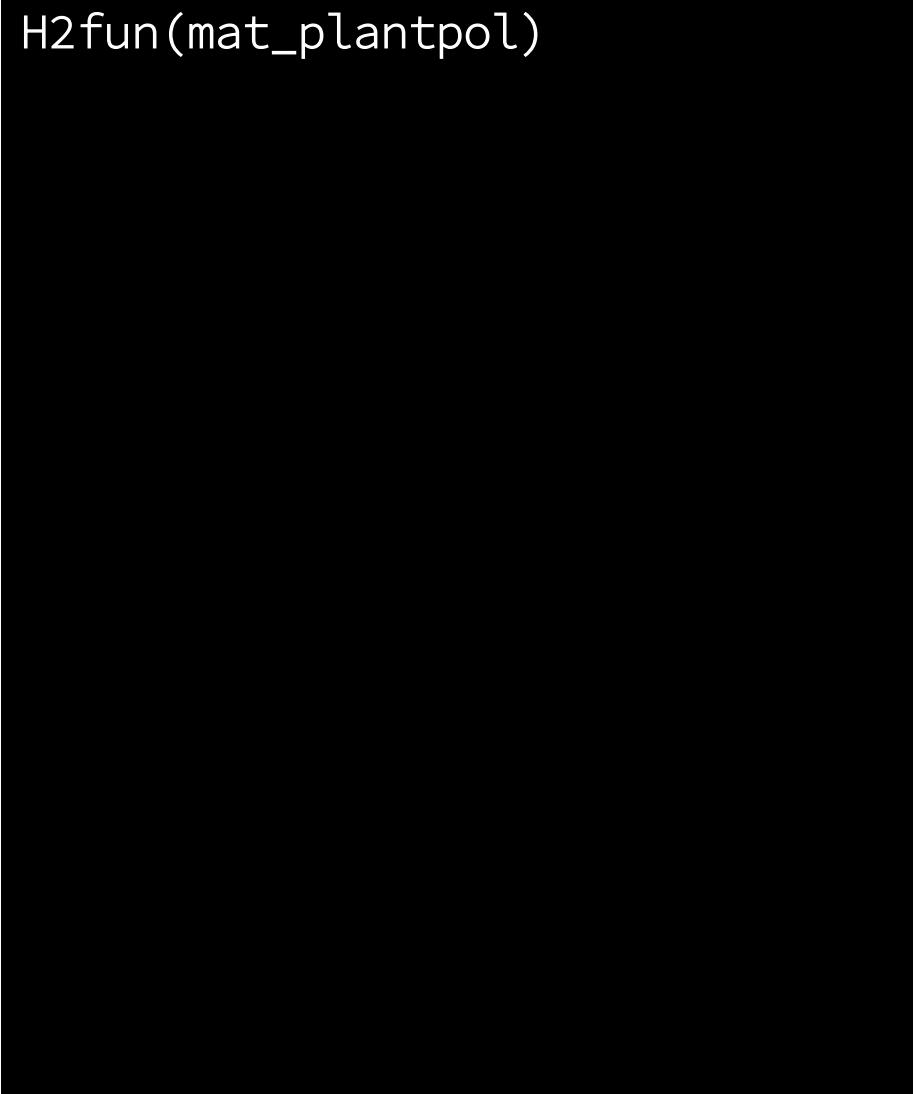
$$H_2' = \frac{H_{2\max} - H_2}{H_{2\max} - H_{2\min}}$$

where $H_2 = -\sum \frac{w_{ij}}{W} \log\left(\frac{w_{ij}}{W}\right)$

H_2' measures the overall level of specialization in the network

can be compared among webs

H2fun(mat_plantpol)



Nestedness

- Idea: measure the tendency for specialists to only interact with a subsample of interactors of generalists
- Several indices

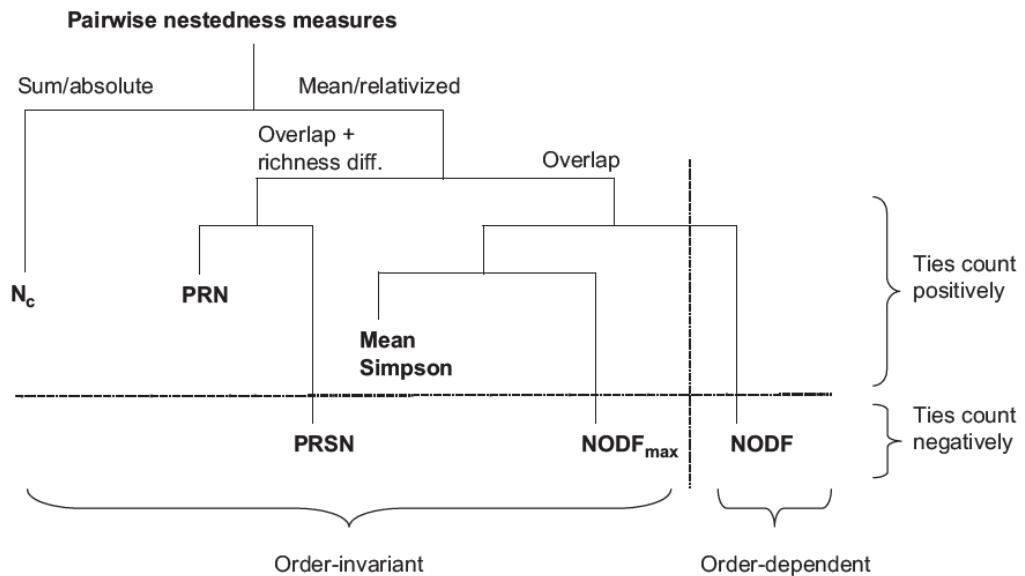


Figure 4. A tree-like summary of conceptual and technical issues to be considered when selecting pairwise measures of nestedness.

Nestedness

Computes nestedness indices using the bipartite package (borrows from vegan)

```
pollination_nestdness<-  
nested(mat_plantpol, method =  
"ALL", rescale=TRUE,  
normalised=TRUE)
```

```
> pollination_nestdness<-nested(mat_plantpol, method = "ALL", rescale=TRUE, normalised=TRUE)  
> pollination_nestdness  
binmatnest.temperature      discrepancy2      discrepancy      C score      checker  
89.656269                  60.000000      71.000000      50.388420      1.006407  
      NODF2                      NODF  
30.784672                  15.656957      weighted NODF      wine  
> |
```

using vegan directly

binmatnest.temperature	discrepancy2	discrepancy	C score	checker
89.656269	60.000000	71.000000	50.388420	1.006407
NODF2	NODF	weighted NODF	wine	WNODF
30.784672	15.656957	13.495474	68.579573	22.582853

```
nestednodf(mat_plantpol)  
nestednodf(mat_plantpol,weighted=  
TRUE)  
nestednodf(mat_plantpol,weighted=  
TRUE,wbinary=T)  
nestednodf(mat_plantpol_bin)  
nestednodf(mat_plantpol_bin,weigh  
ted=TRUE)  
nestednodf(mat_plantpol_bin,weigh  
ted=TRUE,wbinary=T)
```

Nestedness

- Not necessarily a fruitful concept because...
 - theoretically questioned (James et al. 2012 = connectance)
 - choosing the good index is hard (Podani & Schmera 2012)
 - different indices give different results (Thomas et al. 2015)
 - for power law degree distribution, nestedness is completely determined by the exponent (Astegiano et al. 2015)
 - nestedness of common datasets is natural when sampling from the right matrix ensemble (Payrató-Borràs et al. 2019)

Robustness to secondary extinctions

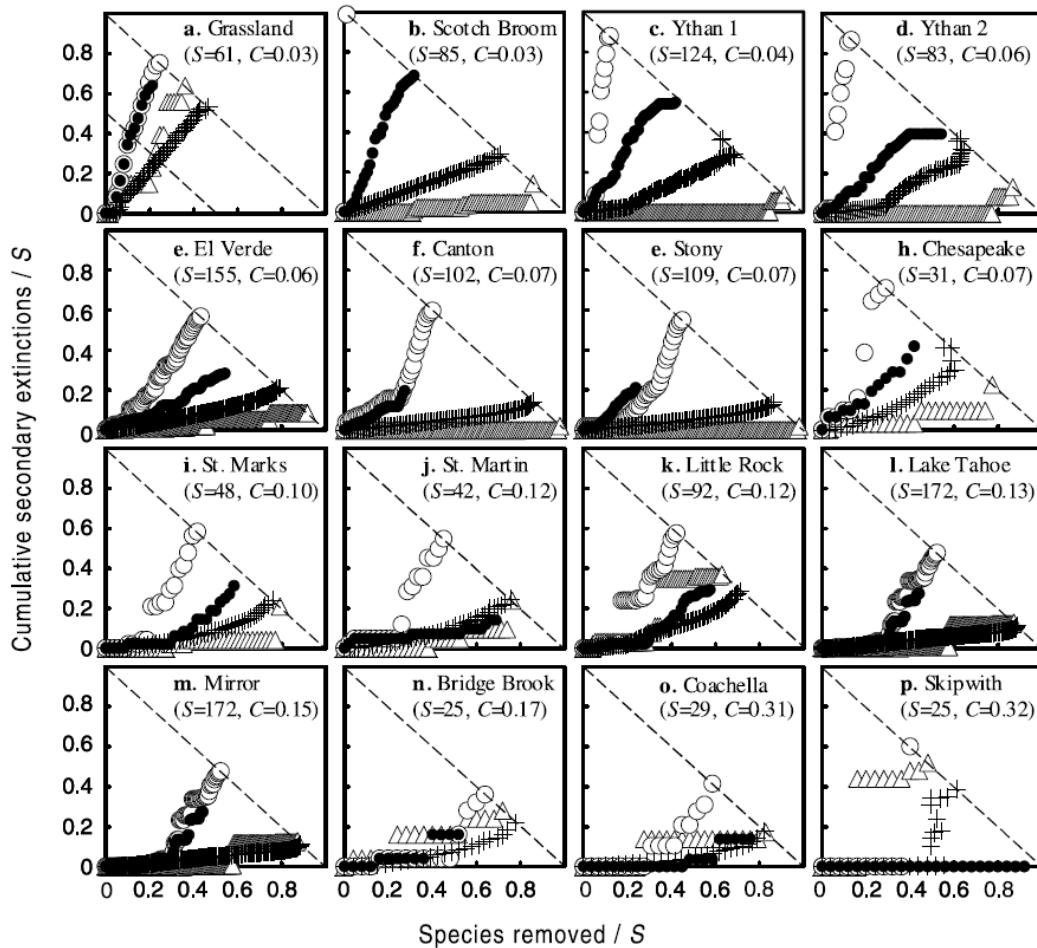
General idea: if one species is removed, what happens to the remaining ones?

Under certain rules, secondary extinction

Principle: how many species are lost when R species are removed?

R_{50} = how many species need to be removed before 50% of the initial species are extinct?

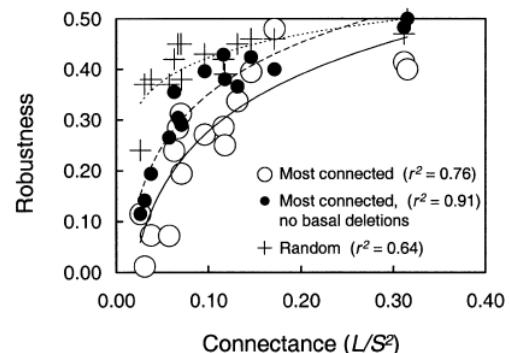
Robustness to secondary extinctions



Examples based on a variety of datasets

Secondary extinction when a species loses all its prey items (except itself if cannibalistic)

Different scenarios based on how removed species are chosen



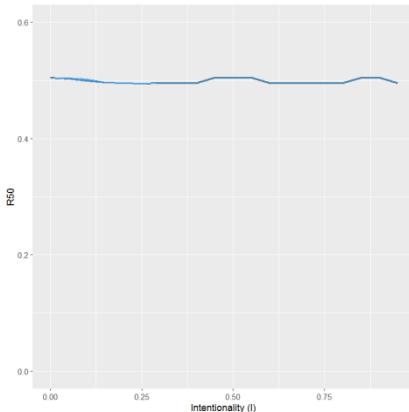
Robustness analysis

On a virtual dataset, using Fwebs

We create a food web using the niche model (will explain later)

Sequence i_index: defines the importance of hubs (in terms of primary attack)

alpha1 = % of secondary extinctions after primary one (among species connected to the removed sp)

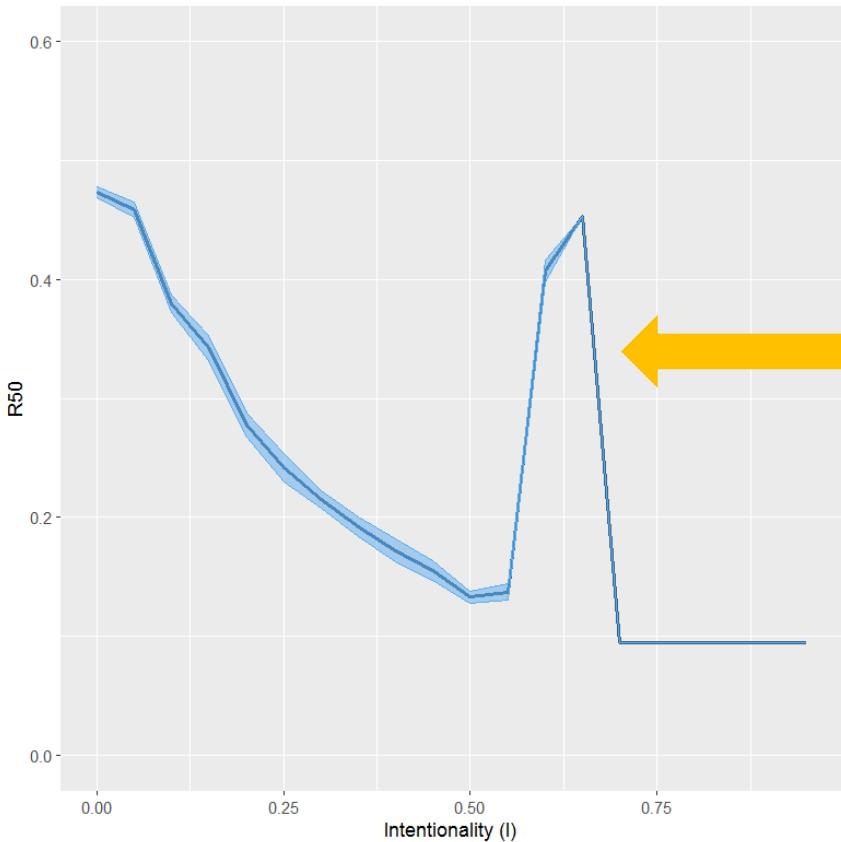


```
niche<-niche_matrix(0.2,200)
m<-niche$matrix
net<-
graph_from_adjacency_matrix(m,m
ode="directed")

i_index <- seq(from = 0, to =
0.95, by =0.05)
prob_exp<-exponent.remove(net,
i_index)
V(net)$name<-1:200
iterate(fw_to_attack=net,
prob_exp, alpha1=50, iter=10,
i_index, plot = TRUE)
```

Robustness analysis

This time on true data



```
prob_exp<-  
exponent.removal(foodweb,  
i_index)  
iterate(fw_to_attack=foodweb,  
prob_exp, alpha1=50, iter=20,  
i_index, plot = TRUE)
```

Targeting hubs can result in disconnected components, which are quite robust against further secondary extinctions

Beta diversity of networks

Idea (Ohlmann et al.): compare networks based on the beta diversity of nodes and links

- For nodes: classic metrics of community ecology
- For links: based on links shared or not between webs

$$L_{ij} = \pi_{ij} p_i p_j$$

L_{ij} : relative weight of the link between sp i and j

p_i : relative abundance of species i

π_{ij} : probability of interaction between species i and j

In practice: beta diversities of L , p and π

Beta diversity of networks

```
for(i in 1:4){  
  graphLocal <- erdos.renyi.game(60, type =  
    'gnp', p.or.m=0.1, directed=TRUE)  
  V(graphLocal)$name <- as.character(1:60)  
  gList <- c(gList, list(graphLocal))  
}  
names(gList) <- c("A","B","C","D")  
  
disPairwise(gList, type='P')  
disPairwise(gList, type='L')  
disPairwise(gList, type='Pi')
```

NETWORK MODELS

Network models

Statistics on networks are hard to test because...
everything is dependent on everything else

Two classes:

- null models: obey some constraints
input = a network, output = a network
- generative models: follow a building process
input = parameters, output = a network

Among generative models, some are probabilistic
(i.e. can be fitted) => statistical tests, etc.

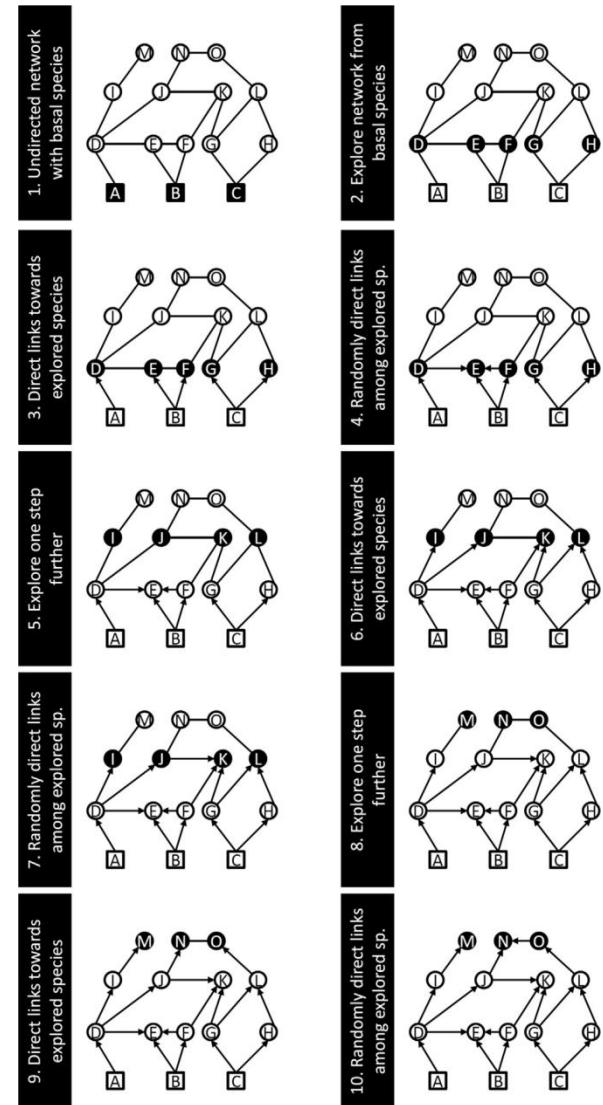
How to make a food web from an undirected unipartite graph?

The `make_food_web_from_undirected` function does that based on

- an undirected graph, and
- a number of basal species

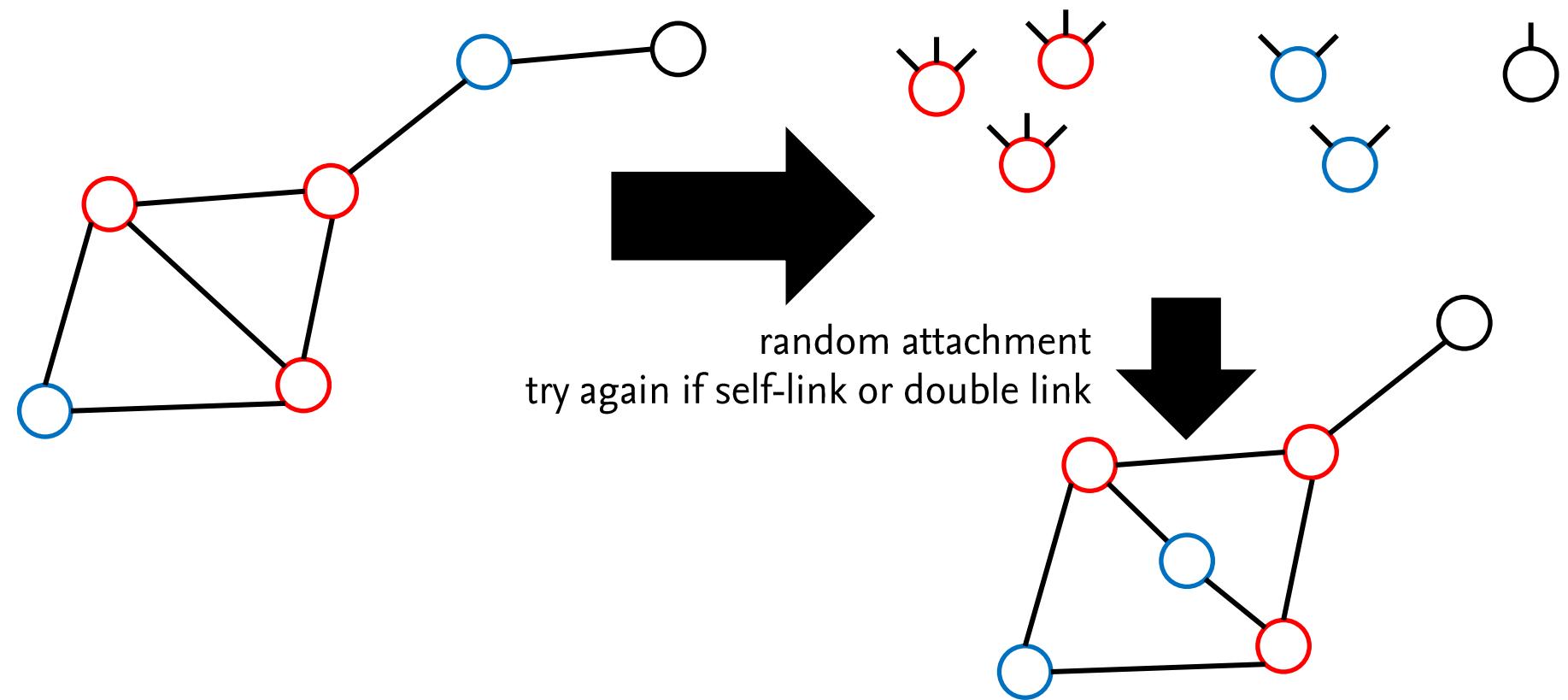
Two steps:

1. coloring of the graph (= sets of nodes that never interact), to find one color that has at least as many members as the basal set, then assign the basals
2. direct the graph using a breadth-first search



The configuration model

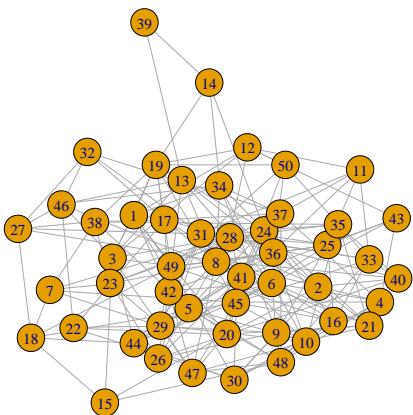
Null model corresponding to randomization of edges with fixed degrees



Generate configurations for an undirected unipartite network

We use Viger-Latapy's algorithm

The function `config_VL` does that automatically from the input network or from its adjacency matrix (automatic detection of input)

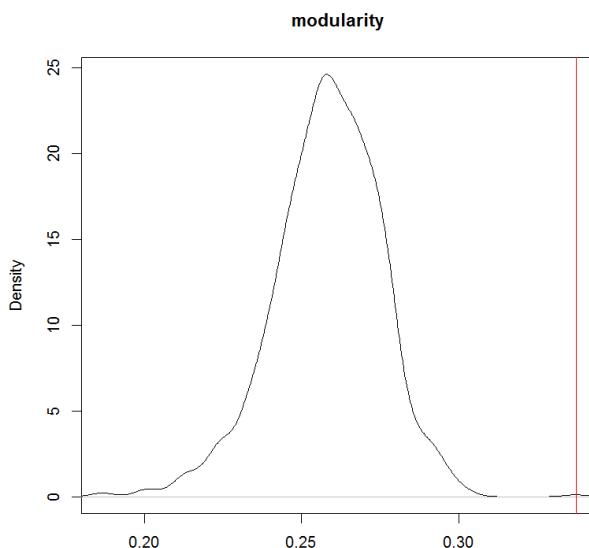


```
net<-sample_gnp(50,0.2,  
directed=FALSE)  
sample.config.undirected<-  
lapply(1:100,function(x)  
sample_degseq(degree(net),  
method = "vl"))  
length(sample.config.undirected  
)  
  
alt.config<-config_VL(net,100)  
plot(alt.config[[42]])
```

Example of use of unipartite configuration model

We want to assess whether the **modularity** of the food web data (largest component only) is extraordinary or not...

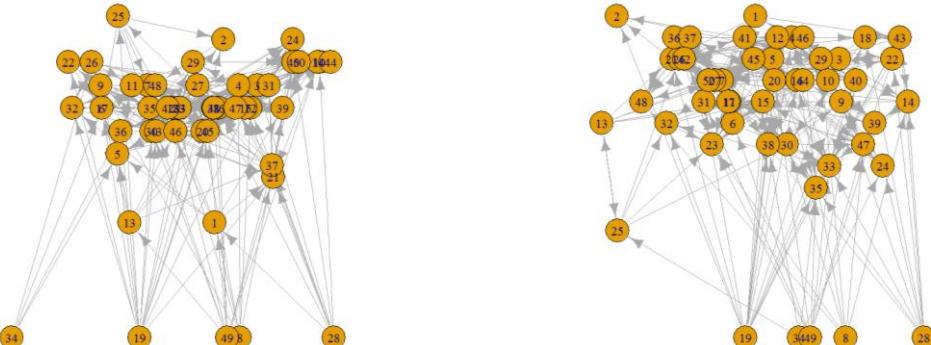
The function `p.val` computes the p-value based on a collection of null model-generated networks and their modularities



```
one_comp_foodweb<-  
as.undirected(largest_component  
(foodweb))  
one_comp_foodweb_LE.mod<-  
cluster_leading_eigen(one_comp_  
foodweb)  
one_comp_foodweb_LE.mod$mod  
  
foodweb_configs<-  
config_VL(one_comp_foodweb,1000  
)  
mods_configs<-sapply(1:1000,  
function(x)  
cluster_leading_eigen(foodweb_c  
onfigs[[x]])$mod)  
p.val(one_comp_foodweb_LE.mod$m  
od,mods_configs,method="larger"  
,label="modularity")
```

Randomization of directed networks

Also obtained from `sample_degseq` in `igraph`, but not assured of sampling uniformly among all possible directed networks with the same degree sequences



```
net<-sample_gnp(50,0.2,  
directed =FALSE)  
net_directed<-  
make_food_web_from_undirected(n  
et,5)  
  
sample.config.directed<-  
lapply(1:100,function(x)  
sample_degseq(degree(net_direct  
ed,mode="out"),  
degree(net_directed,mode="in"),  
method = "simple.no.multiple"))  
  
par(mfrow=c(1,2))  
plot(net_directed,layout=layout  
_as_food_web3(net_directed))  
plot(sample.config.directed[[1]  
],layout=layout_as_food_web3(sa  
mple.config.directed[[1]]))
```

Configuration model for bipartite networks

We use Strona's curveball algorithm

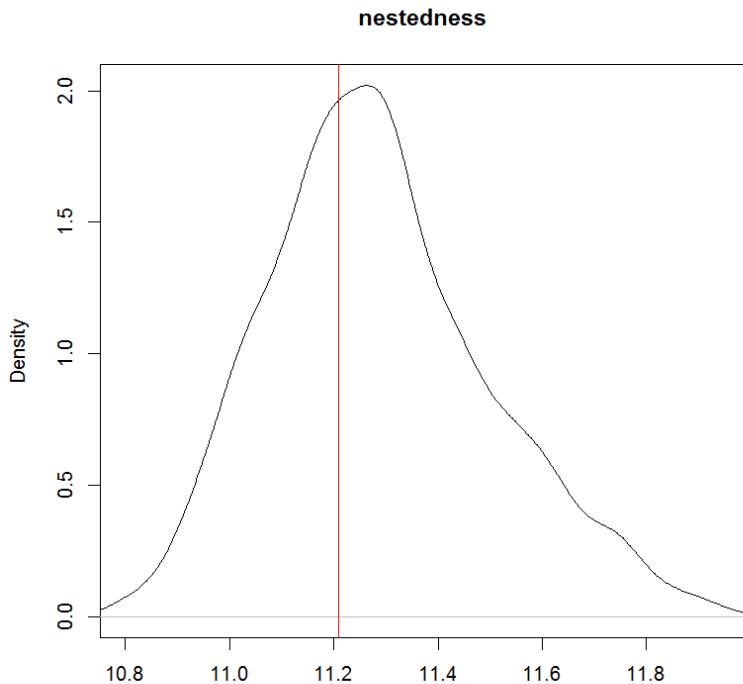
Also implemented as the choice by default in `config_VL` when the input is recognized as bipartite

Don't forget the `vegan::` before calling `nullmodel` – package bipartite also has its own `nullmodel` function...

```
net<-  
sample_bipartite(40,60,"gnp",0.  
1)  
net<-largest_component(net)  
net_mat<-  
as.matrix(as_biadjacency_matrix  
(net))  
sample.bip.config<-  
simulate(vegan::nullmodel(net_m  
at,"curveball"),nsim=1000)  
dim(sample.bip.config)  
  
alt.config<-config_VL(net,1000)
```

Example of use of bipartite configuration model

We want to ascertain how extraordinary the nestedness of our generated network is



```
net_nestedness<-nested(net_mat,  
method = "NODF2")  
nestedness_configs<-  
  sapply(1:1000, function(x)  
nested(sample.bip.config[, ,x],  
method = "NODF2"))
```

```
p.val(net_nestedness,nestedness  
_configs,method="two-  
sided",label="nestedness")
```

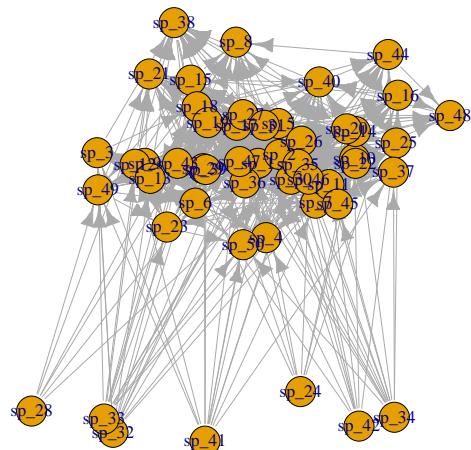
Generative models

A sample of existing models

- *Erdos-Rényi*: all links are equally likely
- *Preferential attachment*: nodes added sequentially, linked more likely with high-degree nodes
- “*Models with invariants*”: models that try to preserve mean degree or connectance
- *Models with prescribed degree sequence*
- *Expected degree distribution* models

Food web generation using ER

Erdős-Rényi algorithm: all edges have the same probability of occurring
Plot using MacKay's TL as y-axis



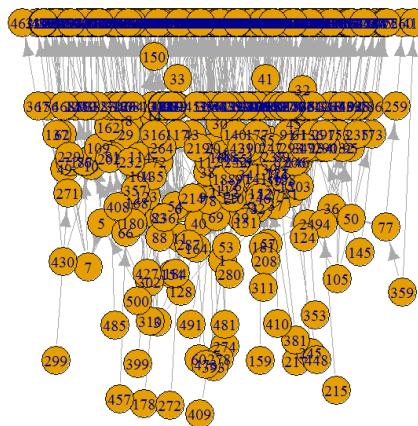
Kolmogorov-Smirnov test: degree distribution is not different from binomial

```
net<-  
make_food_web_from_undirected(s  
ample_gnp(50,0.2,directed  
=TRUE),basal = 10)  
  
V(net)$name<-  
sapply(1:50,function(x)  
paste0("sp_",x))  
  
plot(net,layout=layout_as_food_  
web2(net))  
  
ks.test(degree(net),"pbinary",  
length(V(net))-  
1,mean(degree(net))/(length(V(n  
et))-1))
```

“Food web” generation using preferential attachment

Generate the undirected graph using
pref. attachment

Make it a food web



NB: can be detected as significantly
different from a power law... even when
power law parameters are re-estimated
beforehand

```
net<-
make_food_web_from_undirected(sam-
ple_pa(500, power = 1, directed =
FALSE, out.pref=TRUE),basal = 50)
```

```
plot(net,layout=layout_as_food_we-
b2(net))
```

```
net_dist<-displ$new(degree(net))
net_dist$setXmin(estimate_xmin(di-
spl$new(degree(net)))$xmin)
net_pars<-estimate_pars(net_dist)
```

```
ks.test(degree(as.undirected(net))
),"ppldis",xmin =
estimate_xmin(displ$new(degree(ne-
t)))$xmin, alpha = net_pars$pars)
```

Food web models with invariants

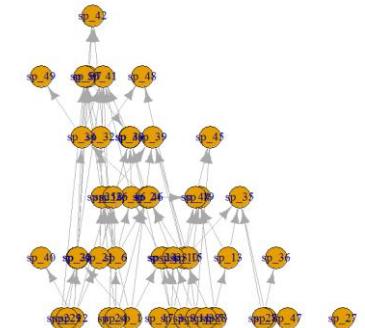
Two models from the 80-90's:

1. the **cascade model** (Cohen-Briand): predators eat species that are smaller than them with probability c/S
⇒ connectance decreases as $1/S$

```
net<-cascade_matrix(3,50)
sum(net)/(dim(net)[1]*(dim(net)[1]-1))
```

```
graph_cascade<-graph_from_adjacency_matrix(net)
V(graph_cascade)$name<-sapply(1:50,function(x) paste0("sp_",x))
plot(graph_cascade,layout=layout_as_food_web3(graph_cascade))
```

```
net<-cascade_matrix(3,200)
sum(net)/(dim(net)[1]*(dim(net)[1]-1))
```



Food web models with invariants

Two models from the 80-90's:

2. the **niche model** (Williams-Martinez): predators eat all species that are within a size interval

⇒ connectance stays constant

```
net<-niche_matrix(0.1,50)
sum(net$matrix)/(dim(net$matrix)[1]*(dim(net$matrix)[1]-1))

graph_niche<-graph_from_adjacency_matrix(net$matrix)
V(graph_niche)$name<-sapply(1:50,function(x) paste0("sp_",x))
plot(graph_niche,layout=layout_as_food_web3(graph_niche))

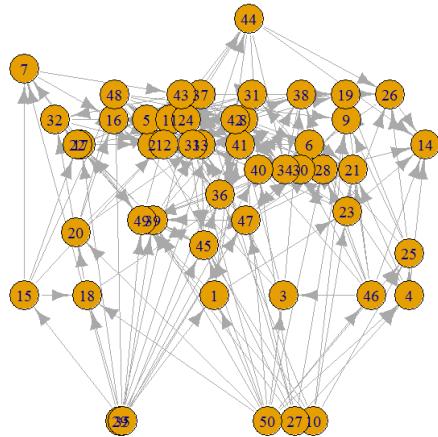
net<-niche_matrix(0.1,200)
sum(net$matrix)/(dim(net$matrix)[1]*(dim(net$matrix)[1]-1))
```

Food webs with prescribed degree sequence

Idea:

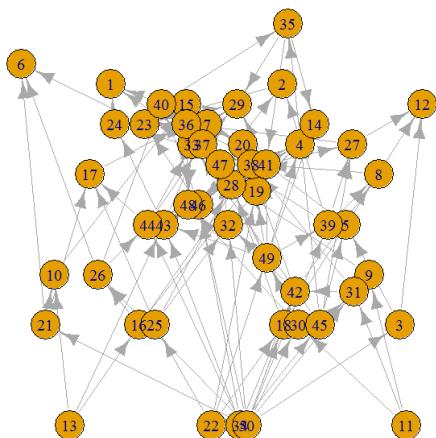
1. generate a degree sequence that is typical of a given degree distribution (using quantiles)
2. generate an undirected graph from the degree sequence (using Viger-Latapy's algorithm)
3. make it a food web using the algorithm described before

Food webs with prescribed degree sequence



```
net<-
  generate_DM_model(50,0.1,quant_
  fun="qpois", lambda=5)

plot(net,layout=layout_as_food_
web3(net))
```

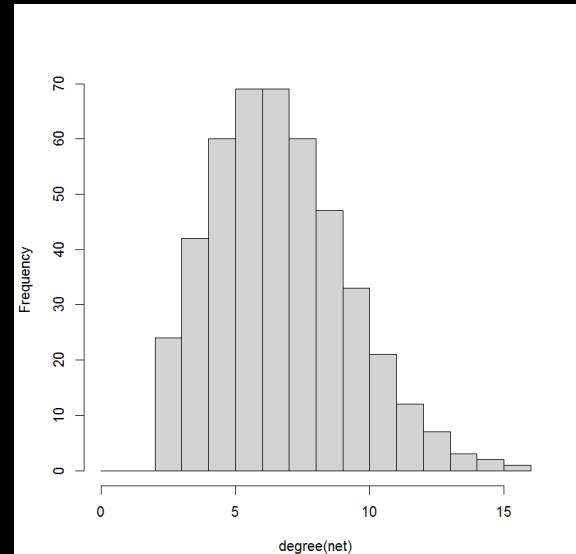


```
net<-
  generate_DM_model(50,0.1,quant_
  fun="qnbinom", mu = 4, size=50)

plot(net,layout=layout_as_food_
web3(net))
```

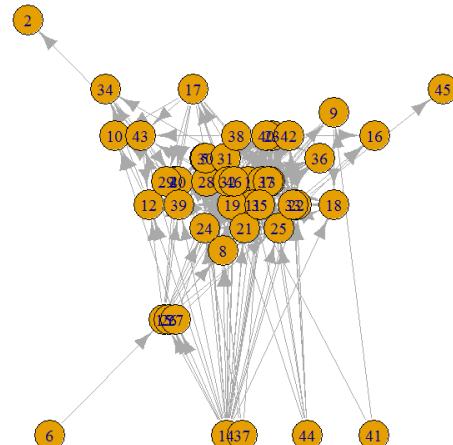
Food webs with prescribed degree sequence

```
net<-
generate_DM_model(450,0.1,quant_fun="qpois",
lambda=7)
hist(degree(net),breaks=0:max(degree(net)),main
="")
ks.test(degree(net),"ppois",lambda =
mean(degree(net)))
```



Food webs following the EDD model

Generate connections according to a latent layer, with more or less heterogeneity in expected degrees



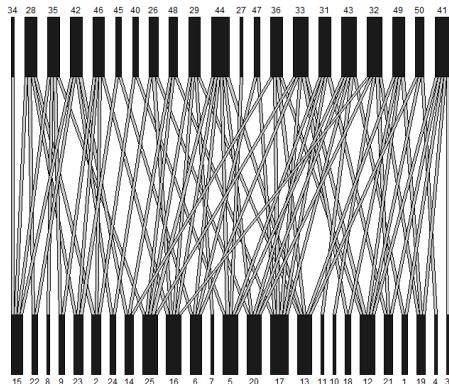
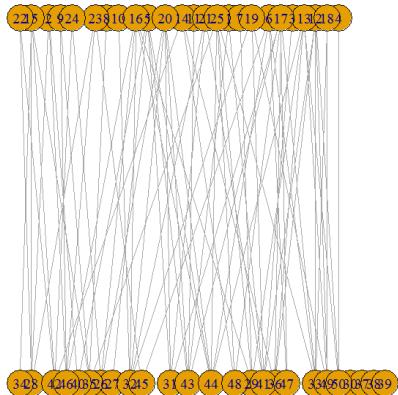
```
net_EDD<-Simul_EDD(50,0.2,2)
graph_EDD<-
graph_from_adjacency_matrix(net
_EDD$net,mode="undirected")
graph_EDD<-
largest_component(graph_EDD)

plotMyMatrix(net_EDD$net)

foodweb_EDD<-
make_food_web_from_undirected(g
raph_EDD,5)
plot(foodweb_EDD,layout=layout_
as_food_web3(foodweb_EDD))
```

Bipartite network generation using ER

The probability of interaction only applies between nodes from different levels



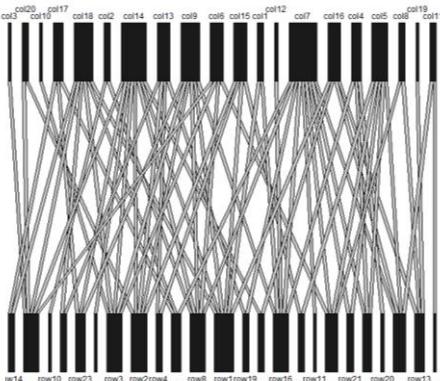
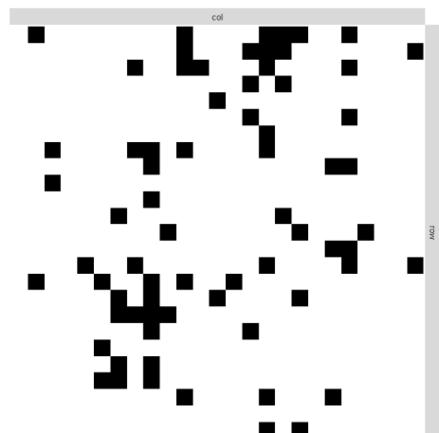
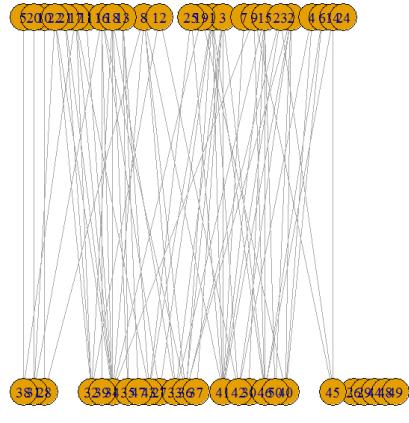
```
net<-
sample_bipartite(25,25,"gnp",0.
1)

plot(net,layout=layout_as_bipar
tite)

plotweb(as_biadjacency_matrix(n
et))
```

Bipartite networks using the BEDD model

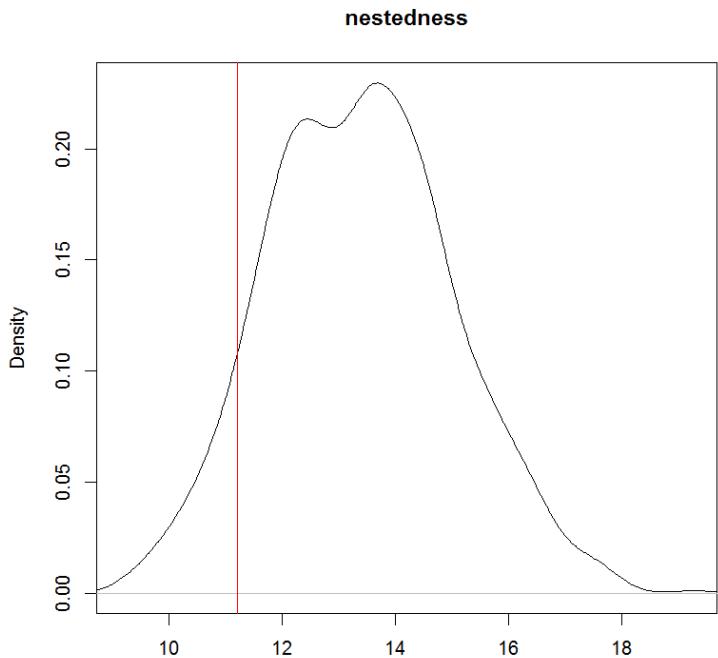
Same as EDD, but bipartite



```
net_bedd<-SimulB_EDD(25, 25,  
0.1, 2, 2)  
graph_bedd<-  
graph_from_bipartite_matrix(n  
et_bedd$net)  
  
plot(graph_bedd, layout=layout_a  
s_bipartite)  
  
plotweb(net_bedd$net)  
  
plotMyMatrix(net_bedd$net)
```

Randomization through the BEDD model

Example of use with nestedness



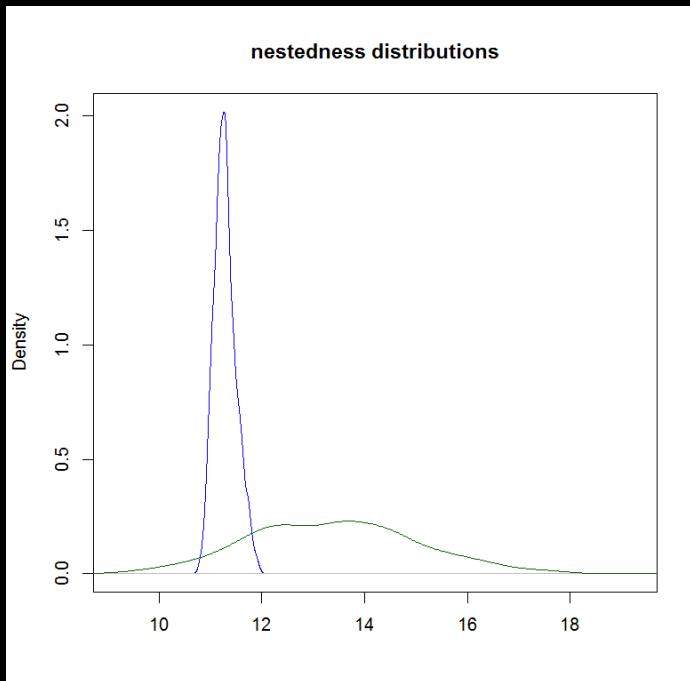
```
sample.bip.EDD<-lapply(1:1000,  
function(x)  
randomize.BEDD(net_mat))
```

```
nestedness_EDD<-sapply(1:1000,  
function(x)  
nested(sample.bip.EDD[[x]]),  
method = "NODF2"))
```

```
p.val(net_nestedness,nestedness  
_EDD,method="two-  
sided",label="nestedness")
```

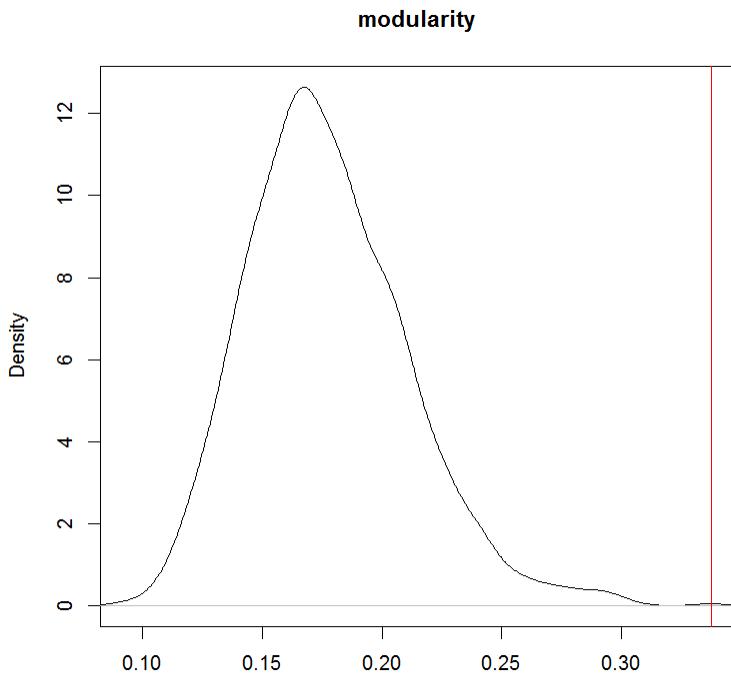
Comparison of null models

```
plot(density(nestedness_configs), xlim=c(min(nestedness_configs, nestedness_EDD), max(nestedness_configs, nestedness_EDD)), xlab="", main="nestedness distributions", col="blue")
lines(density(nestedness_EDD), col="darkgreen")
```



Randomization using the EDD model

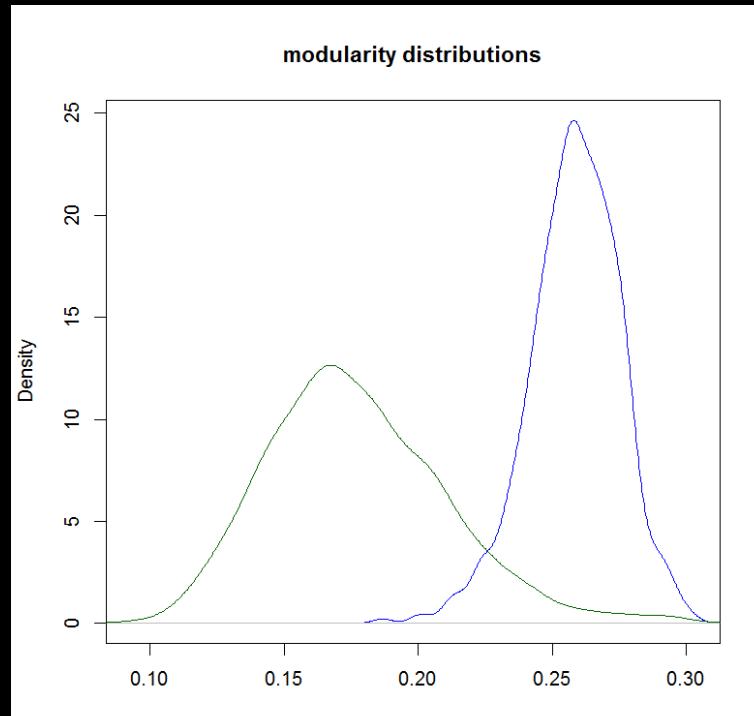
Example of use with modularity in food webs



```
foodweb_EDD<-lapply(1:1000,  
function(x)  
randomize.EDD(one_comp_foodweb)  
)  
mods_EDD<-sapply(1:1000,  
function(x)  
cluster_leading_eigen(as.undirected(foodweb_EDD[[x]]))$mod)  
p.val(one_comp_foodweb_LE.mod$mod,mods_EDD,method="larger",label="modularity")
```

Comparison of null models

```
plot(density(mods_configs),xlim=c(min(mods_configs,mods_EDD),max(mods_configs,mods_EDD)),xlab="",main="modularity distributions",col="blue")
lines(density(mods_EDD),col="darkgreen")
```



References

- Barrett, S. C. & Hellenurm, K. (1987) The reproductive biology of boreal forest herbs. I. Breeding systems and pollination. *Canadian Journal of Botany*, **65**, 2036-2046.
- Blüthgen, N., Menzel, F. & Blüthgen, N. (2006) Measuring specialization in species interaction networks. *BMC ecology*, **6**, 9.
- Clauset, A., Shalizi, C. R. & Newman, M. E. J. (2009) Power-law distributions in empirical data. *SIAM Review*, **51**, 661-703.
- Cruz-Escalona, V. H., Arreguín-Sánchez, F. & Zetina-Rejón, M. (2007) Analysis of the ecosystem structure of Laguna Alvarado, western Gulf of Mexico, by means of a mass balance model. *Estuarine, Coastal and Shelf Science*, **72**, 155-167.
- Dunne, J. A., Williams, R. J. & Martinez, N. D. (2002) Food-web structure and network theory: The role of connectance and size. *Proceedings of the National Academy of Sciences*, **99**, 12917-12922.
- Dunne, J. A., Williams, R. J. & Martinez, N. D. (2002) Network structure and biodiversity loss in food webs: robustness increases with connectance. *Ecology Letters*, **5**, 558-567.
- James, A., Pitchford, J. W. & Plank, M. J. (2012) Disentangling nestedness from models of ecological complexity. *Nature*, **487**, 227-230.
- Leger, J.-B., Daudin, J.-J. & Vacher, C. (2015) Clustering methods differ in their ability to detect patterns in ecological networks. *Methods in Ecology and Evolution*, **6**, 474-481.
- Lewinsohn, T. M., Prado, P. I., Jordano, P., Bascompte, J. & Olesen, J. M. (2006) Structure in plant-animal interaction assemblages. *Oikos*, **113**, 174-184.
- MacKay, R. S., Johnson, S. & Sansom, B. (2020) How directed is a directed network? *Royal Society Open Science*, **7**, 201138.
- Massol, F., Dubart, M., Calcagno, V., Cazelles, K., Jacquet, C., Kéfi, S. & Gravel, D. (2017) Island biogeography of food webs. *Advances in Ecological Research* vol. 56 - *Networks of Invasion: A Synthesis of Concepts* (eds D. A. Bohan, A. J. Dumbrell & F. Massol), pp. 183-262. Academic Press.
- Newman, M. E. J. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, **103**, 8577-8582.
- Pocock, M. J. O., Evans, D. M. & Memmott, J. (2012) The robustness and restoration of a network of ecological networks. *Science*, **335**, 973-977.
- Payrató-Borràs, C., Hernández, L. & Moreno, Y. (2019) Breaking the Spell of Nestedness: The Entropic Origin of Nestedness in Mutualistic Systems. *Physical Review X*, **9**, 031024.
- Podani, J. & Schmera, D. (2012) A comparative evaluation of pairwise nestedness measures. *Ecography*, **35**, 889-900.
- Strona, G., Nappo, D., Boccacci, F., Fattorini, S. & San-Miguel-Ayanz, J. (2014) A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, **5**.
- Stumpf, M. P. H. & Porter, M. A. (2012) Critical truths about power laws. *Science*, **335**, 665-666.
- von Luxburg, U. (2007) A tutorial on spectral clustering. *Statistics and Computing*, **17**, 395-416.
- Williams, R. J. & Martinez, N. D. (2000) Simple rules yield complex food webs. *Nature*, **404**, 180-183.
- Yang, Z., Algesheimer, R. & Tessone, C. J. (2016) A Comparative Analysis of Community Detection Algorithms on Artificial Networks. *Scientific Reports*, **6**, 30750.