main_jupyter

February 5, 2021

1 Data aggregation in food webs: effects on key positions

The following R code uses the igraph package to plot the original plankton food web of the Gulf of Naples, as well as its aggregated versions.

1.1 Load packages and import data

Let's start from calling the packages we will need.

[4]:
```
library(igraph)
library(NetIndices)
```

```
Attaching package: ‘igraph’


The following objects are masked from ‘package:stats’:

    decompose, spectrum


The following object is masked from ‘package:base’:

    union


Warning message:
"package ‘NetIndices’ was built under R version 4.0.2"
Loading required package: MASS
```

Import the adjacency matrices of the original food web and of the clustered food webs.

[ ]:
```
edge_list <- read.delim('../data/edge_list_for_R.txt', header = FALSE)
edge_list_jaccard <- read.delim('../variables/edge_list_jaccard.txt', header =␣
 ↪FALSE)
edge_list_rege <- read.delim('../variables/edge_list_rege.txt', header = FALSE)
edge_list_prey_modularity <- read.delim('../variables/edge_list_prey_modularity.
 ↪txt', header = FALSE)
```

```
edge_list_density_modularity <- read.delim('../variables/
↪edge_list_density_modularity.txt', header = FALSE)
edge_list_groups <- read.delim('../variables/edge_list_groups.txt', header =␣
↪FALSE)
```

Import the matrix with the membership of the nodes to different clusters of different food webs.

```
[ ]: membership <- read.csv('../variables/membership.txt',header = FALSE)
```

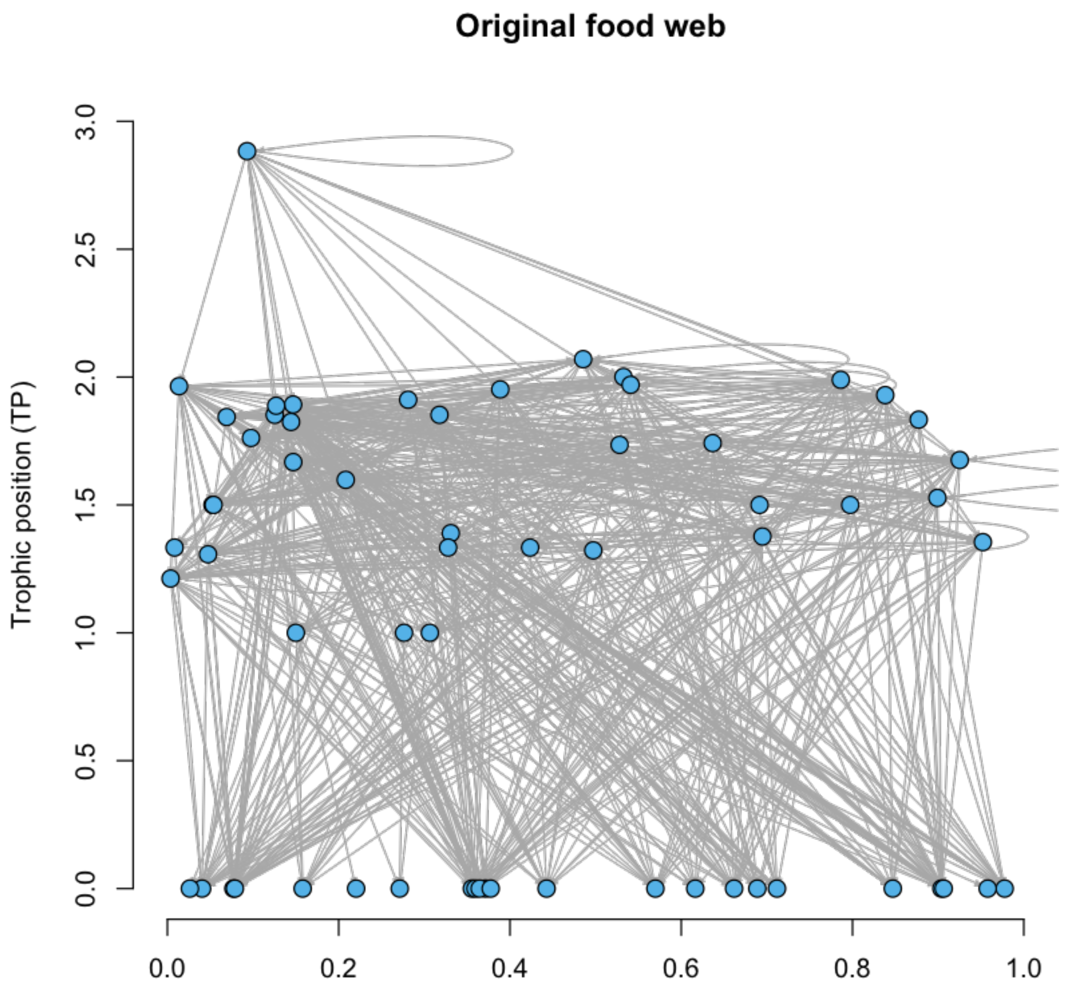
Import the trophic position of the different nodes in different food webs.

```
[ ]: TP <- as.matrix(read.csv('../variables/TP.txt', header = FALSE))
     TP_jaccard <- as.matrix(read.csv('../variables/TP_jaccard_clusters.txt', header␣
     ↪= FALSE))
```

1.2 Original food web

1.2.1 Create

```
[6]: edges <- as.matrix(edge_list[,c("V2","V1")]) #You need to invert i and j
     G <- graph_from_edgelist(edges)
     G$weight <- edge_list[,"V3"]
     V(G)$TP <- TP[,1]
     layout.matrix<-matrix( nrow=length(V(G)),ncol=2)
     layout.matrix[,1]<-runif(length(V(G)))
     layout.matrix[,2] <- TP[,1]
     V(G)$color <- "666"
     A<-get.adjacency(G,sparse=F) #i and j are inverted
     plot.igraph(G,
                 main= "Original food web ",
                 vertex.label=NA,
                 vertex.size=2,
                 edge.arrow.size=.25,
                 layout=layout.matrix,
                 axes=TRUE,
                 xlim = c(0,1),
                 ylim=c(0,3),
                 ylab="Trophic position (TP)",
                 rescale=F,
                 asp=0)
```



1.2.2 Global metrics

Let's take a look at some basic information about our food web.

[19]:
```
indices<-data.frame(GenInd(A))
indices
```

```
                         N      T..     TST     Lint    Ltot    LD         C           Tijbar   TSTbar     Cl
A data.frame: 1 × 10     <int>  <dbl>   <dbl>   <int>   <int>   <dbl>      <dbl>       <dbl>    <dbl>      <
                         62     652     1077    652     652     10.51613   0.1723956   1        17.37097   0.
```

where N = number of compartments, excluding the externals; T.. = total system throughput; TST = total system throughflow; Lint = number of Internal links; Ltot = total number of links; LD = link density; C = connectance (internal); Tijbar = average link weight; TSTbar = average

Compartment Throughflow; Cbar = compartmentalization,[0,1], the degree of connectedness of subsystems within a network.

1.3 Clustered food web

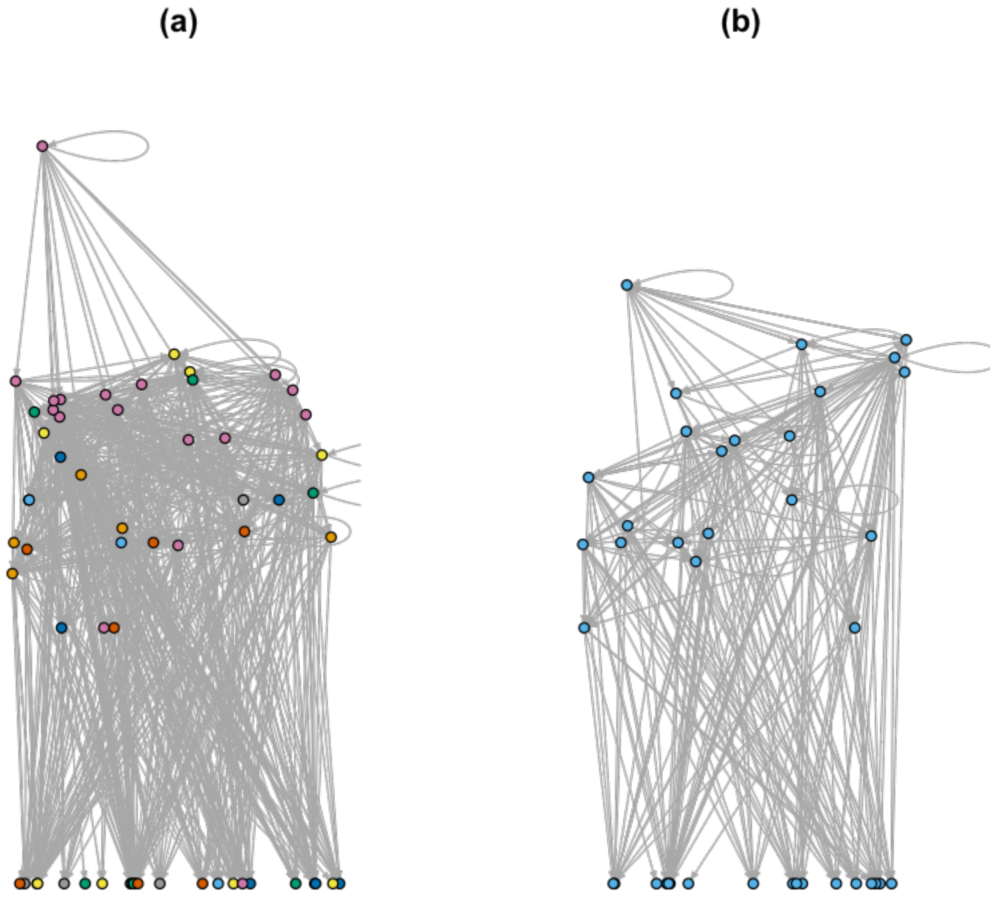
1.3.1 Jaccard index food web

Let's now take a look at the Jaccard food web.

[20]:
```
par(mfrow=c(1,2))

#JACCARD FOOD WEB - colour
V(G)$jaccard <- membership[,1]
V(G)$color <- V(G)$jaccard
plot.igraph(G,
            main = "(a)", #Clustering through the jaccard index
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            rescale=F,
            asp=0,
            #axes=T,
            #ylab="Trophic position (TP)"
)

#JACCARD FOOD WEB - create
edges_jaccard <- as.matrix(edge_list_jaccard[,c("V2","V1")]) #You need to␣
 ↪invert i and j
G_jaccard <- graph_from_edgelist(edges_jaccard)
#G_jaccard$weight <- edge_list_jaccard[,"V3"]
V(G_jaccard)$TP <- TP_jaccard
layout.matrix<-matrix( nrow=length(V(G_jaccard)),ncol=2)
layout.matrix[,1]<-runif(length(V(G_jaccard)))
layout.matrix[,2] <- TP_jaccard
V(G_jaccard)$color <- "666" #V(G_jaccard)
plot.igraph(G_jaccard,
            main= "(b)", #Jaccard food web
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            #axes=TRUE,
            #ylab="Trophic position (TP)",
            rescale=F,
            asp=0)
```



1.3.2 REGE index food web

[23]:
```
#REGE FOOD WEB - colour
V(G)$rege <- membership[,2]
V(G)$color <- V(G)$rege
plot.igraph(G,
            main = "(c)", #Clustering through the rege index
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
```

```r
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            rescale=F,
            asp=0,
            #axes=T,
            #ylab="Trophic position (TP)"
)

#REGE FOOD WEB - create
edges_rege <- edge_list_rege[,c("V2","V1")] #You need to invert i and j
G_rege <- graph_from_edgelist(edges_rege)
weights_rege<-edge_list_rege[,"V3"]
G_rege$weight <- weights_rege
#V(G_rege)$TP <- TP
layout.matrix<-matrix( nrow=length(V(G_rege)),ncol=2)
layout.matrix[,1]<-runif(length(V(G_rege)))
#layout.matrix[,2] <- TP
#V(G_rege)$color <-
plot.igraph(G_rege,
            main= "(b)", #Jaccard food web
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            #axes=TRUE,
            #ylab="Trophic position (TP)",
            rescale=F,
            asp=0)
```

```
Error in layout[loops.v, 1]: subscript out of bounds
Traceback:

1. plot.igraph(G, main = "(c)", vertex.label = NA, vertex.size = 3,
 .     edge.arrow.size = 0.25, layout = layout.matrix, xlim = c(0,
 .         1), ylim = c(0, 3), rescale = F, asp = 0, )
```

(c)

1.3.3 Density-based food web

[26]:
```
V(G)$densitymodularity <- membership[,4]
V(G)$color <- V(G)$densitymodularity
plot.igraph(G,
            #main = "(g)", #Clustering through density modularity maximisation
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            rescale=F,
```

```r
            asp=0,
            #axes=T,
            #ylab="Trophic position (TP)"
)

edges_densitymodularity <- edge_list_densitymodularity[,c("V2","V1")] #You need␣
↪to invert i and j
G_densitymodularity <- graph_from_edgelist(edges_densitymodularity)
weights_densitymodularity<-edge_list_densitymodularity[,"V3"]
G_densitymodularity$weight <- weights_densitymodularity
#V(G_densitymodularity)$TP <- TP_densitymodularity
layout.matrix<-matrix( nrow=length(V(G_densitymodularity)),ncol=2)
layout.matrix[,1]<-runif(length(V(G_densitymodularity)))
#layout.matrix[,2] <- TP
#V(G_densitymodularity)$color <-
plot.igraph(G_densitymodularity,
            main= "(b)", #Jaccard food web
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            #axes=TRUE,
            #ylab="Trophic position (TP)",
            rescale=F,
            asp=0)
```

```
Error in layout[loops.v, 1]: subscript out of bounds
Traceback:

1. plot.igraph(G, vertex.label = NA, vertex.size = 3, edge.arrow.size = 0.25,
 .     layout = layout.matrix, xlim = c(0, 1), ylim = c(0, 3), rescale = F,
 .     asp = 0, )
```

1.3.4 Prey-based food web

[25]:
```
V(G)$preymodularity <- membership[,3]
V(G)$color <- V(G)$preymodularity
plot.igraph(G,
            main = "(e)", #Clustering through prey modularity maximisation
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            rescale=F,
            asp=0,
            #axes=T,
            #ylab="Trophic position (TP)"
)

edges_preymodularity <- edge_list_preymodularity[,c("V2","V1")] #You need to␣
↪invert i and j
G_preymodularity <- graph_from_edgelist(edges_preymodularity)
weights_preymodularity<-edge_list_preymodularity[,"V3"]
G_preymodularity$weight <- weights_preymodularity
#V(G_preymodularity)$TP <- TP
layout.matrix<-matrix( nrow=length(V(G_preymodularity)),ncol=2)
layout.matrix[,1]<-runif(length(V(G_preymodularity)))
#layout.matrix[,2] <- TP
#V(G_preymodularity)$color <-
plot.igraph(G_preymodularity,
            main= "(b)", #Jaccard food web
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            #axes=TRUE,
            #ylab="Trophic position (TP)",
            rescale=F,
            asp=0)
```

```
Error in layout[loops.v, 1]: subscript out of bounds
Traceback:

1. plot.igraph(G, main = "(e)", vertex.label = NA, vertex.size = 3,
 .     edge.arrow.size = 0.25, layout = layout.matrix, xlim = c(0,
 .         1), ylim = c(0, 3), rescale = F, asp = 0, )
```

(e)

1.3.5 Group model food web

[27]:
```r
V(G)$groups <- membership[,5]
V(G)$color <- V(G)$groups
plot.igraph(G,
            #main = "(i)", #Clustering through the signed group model
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            rescale=F,
            asp=0,
            #axes=T,
            #ylab="Trophic position (TP)"
)

edges_groups <- edge_list_groups[,c("V2","V1")] #You need to invert i and j
G_groups <- graph_from_edgelist(edges_groups)
weights_groups<-edge_list_groups[,"V3"]
G_groups$weight <- weights_groups
#V(G_groups)$TP <- TP
layout.matrix<-matrix( nrow=length(V(G_groups)),ncol=2)
layout.matrix[,1]<-runif(length(V(G_groups)))
#layout.matrix[,2] <- TP
#V(G_groups)$color <-
plot.igraph(G_groups,
            main= "(b)", #Jaccard food web
            vertex.label=NA,
            vertex.size=3,
            edge.arrow.size=.25,
            layout=layout.matrix,
            xlim = c(0,1),
            ylim=c(0,3),
            #axes=TRUE,
            #ylab="Trophic position (TP)",
            rescale=F,
            asp=0)
```

```
Error in layout[loops.v, 1]: subscript out of bounds
Traceback:

1. plot.igraph(G, vertex.label = NA, vertex.size = 3, edge.arrow.size = 0.25,
 .     layout = layout.matrix, xlim = c(0, 1), ylim = c(0, 3), rescale = F,
 .     asp = 0, )
```

[ ]:
```

```