

Supplementary Table

Table S1. R-script used for this study. To run an R-script, download R from <https://cran.r-project.org/bin/>, and (optionally) RStudio from <https://www.rstudio.com/products/rstudio/download/>. Instructions below are based on the RStudio interface.

```
#####

#To run a command line, select it and press Ctrl+enter or click on the "Run" button in the top right corner.
#Before running the analysis, make sure that the package(s) below are activated ('library'). If you use this R script on another PC, you might have to install the
packages first.
#To install packages, run 'install.packages("package name")' here or below in the Console window (if you want to run commands in the console window, you just have
to click the enter key)
#Packages have to be installed only once on each PC. The 'library' command below has to be run every time you restart RStudio.

install.packages("igraph") #only mandatory for the first time you use the igraph package on your PC
library(igraph) #this package is needed to perform network analyses

#####

#####
#Import data
#####

#this imports an excel file, where the first row and first column are assigned to be names, so only the cells with 0 and 1 are considered for the analysis
#make sure, that your excel table is saved as a CSV file that is separated by semicolons, otherwise it will not work!
#make also sure, that your CSV files are in the same folder as your R script/R project,

sheet.capto <- read.csv("network_sheet.csv", header=T, row.names=1, sep=";")

#This transforms the imported CSV file into an adjacency matrix. This is needed to get the igraph object below

mx.capto <- data.matrix(sheet.capto, rownames.force=T)

#This transforms the matrix into an igraph object. igraph objects are needed to perform a network analysis with the "igraph" package.

net.capto <- graph_from_adjacency_matrix(mx.capto, mode="undirected")

plot(net.capto) #not mandatory. Just use it to take a preliminary look at your network

#####
# First parameters
#####

#You can already determine some parameters that are independent of potential modules
#gives you D, number of observed links relative to the possible number of links

edge_density(net.capto, loops=F)

#gives you C, average clustering coefficient

transitivity(net.capto, type="average")

#gives you K, number of links

sum(degree(net.capto, mode="in"))

#gives you L, average shortest path length

mean_distance(net.capto)

#####
#Modules
#####

#There are several algorithms to calculate modules. We're using here a heuristic walk-trap algorithm.
#Use the following line to find modules. "Steps" is the length of random walks to perform.

mod.capto <- cluster_walktrap(net.capto, steps=3)

#Use these two lines together to get the dendrogram. You can change the values in 'mar' (i.e. the margin) to adjust the format of the plot

par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.capto), horiz=T)

#To get a table with the module membership for each bone/muscle, run the following sets of lines

m.capto <- as.data.frame(sort(membership(mod.capto)))
names(m.capto)[names(m.capto) == "sort(membership(mod.capto))"] <- "Membership"
member.capto <- tibble::rownames_to_column(m.capto, "Bones")
View(member.capto)
```

```
#####
#Q parameter
#####

#Run the "mod.captio" object below
#In the first line of the output, you will find the number of modules ("groups"), and the modularity/Q parameter ("mod")

mod.captio

#Alternatively, to get only the modularity, you can also use the following version

modularity(mod.captio)

#####
#Skull models
#####

#Repeat the procedure above for the respective skull models
#infracenestral 1
sheet.LTF <- read.csv("network_sheet_LTF.csv", header=T, row.names=1, sep=";")
mx.LTF <- data.matrix(sheet.LTF, rownames.force=T)
net.LTF <- graph_from_adjacency_matrix(mx.LTF, mode="undirected")

plot(net.LTF)

edge_density(net.LTF, loops=F)
transitivity(net.LTF, type="average")
sum(degree(net.LTF, mode="in"))
mean_distance(net.LTF)

mod.LTF <- cluster_walktrap(net.LTF, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.LTF), horiz=T)

mLTF <- as.data.frame(sort(membership(mod.LTF)))
names(mLTF)[names(mLTF) == "sort(membership(mod.LTF))"] <- "Membership"
member.LTF <- tibble::rownames_to_column(mLTF, "Bones")
View(member.LTF)

mod.LTF
modularity(mod.LTF)

#infracenestral 2
sheet.thera <- read.csv("network_sheet_therapsid.csv", header=T, row.names=1, sep=";")
mx.thera <- data.matrix(sheet.thera, rownames.force=T)
net.thera <- graph_from_adjacency_matrix(mx.thera, mode="undirected")

plot(net.thera)

edge_density(net.thera, loops=F)
transitivity(net.thera, type="average")
sum(degree(net.thera, mode="in"))
mean_distance(net.thera)

mod.thera <- cluster_walktrap(net.thera, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.thera), horiz=T)

m.thera <- as.data.frame(sort(membership(mod.thera)))
names(m.thera)[names(m.thera) == "sort(membership(mod.thera))"] <- "Membership"
member.thera <- tibble::rownames_to_column(m.thera, "Bones")
View(member.thera)

mod.thera
modularity(mod.thera)

#infracfossal
sheet.infra <- read.csv("network_sheet_infracfossal.csv", header=T, row.names=1, sep=";")
mx.infra <- data.matrix(sheet.infra, rownames.force=T)
net.infra <- graph_from_adjacency_matrix(mx.infra, mode="undirected")

plot(net.infra)

edge_density(net.infra, loops=F)
transitivity(net.infra, type="average")
sum(degree(net.infra, mode="in"))
mean_distance(net.infra)

mod.infra <- cluster_walktrap(net.infra, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.infra), horiz=T)

m.infra <- as.data.frame(sort(membership(mod.infra)))
names(m.infra)[names(m.infra) == "sort(membership(mod.infra))"] <- "Membership"
member.infra <- tibble::rownames_to_column(m.infra, "Bones")
View(member.infra)
```

```

mod.infra
modularity(mod.infra)

#suprafenestral
sheet.UTF <- read.csv("network_sheet_UTF.csv", header=T, row.names=1, sep=";")
mx.UTF <- data.matrix(sheet.UTF, rownames.force=T)
net.UTF <- graph_from_adjacency_matrix(mx.UTF, mode="undirected")

plot(net.UTF)

edge_density(net.UTF, loops=F)
transitivity(net.UTF, type="average")
sum(degree(net.UTF, mode="in"))
mean_distance(net.UTF)

mod.UTF <- cluster_walktrap(net.UTF, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.UTF), horiz=T)

m.UTF <- as.data.frame(sort(membership(mod.UTF)))
names(m.UTF)[names(m.UTF) == "sort(membership(mod.UTF))"] <- "Membership"
member.UTF <- tibble::rownames_to_column(m.UTF, "Bones")
View(member.UTF)

mod.UTF
modularity(mod.UTF)

#suprafossal
sheet.supra <- read.csv("network_sheet_suprafossal.csv", header=T, row.names=1, sep=";")
mx.supra <- data.matrix(sheet.supra, rownames.force=T)
net.supra <- graph_from_adjacency_matrix(mx.supra, mode="undirected")

plot(net.supra)

edge_density(net.supra, loops=F)
transitivity(net.supra, type="average")
sum(degree(net.supra, mode="in"))
mean_distance(net.supra)

mod.supra <- cluster_walktrap(net.supra, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.supra), horiz=T)

m.supra <- as.data.frame(sort(membership(mod.supra)))
names(m.supra)[names(m.supra) == "sort(membership(mod.supra))"] <- "Membership"
member.supra <- tibble::rownames_to_column(m.supra, "Bones")
View(member.supra)

mod.supra
modularity(mod.supra)

#bifenestral 1
sheet.dia <- read.csv("network_sheet_diapsid.csv", header=T, row.names=1, sep=";")
mx.dia <- data.matrix(sheet.dia, rownames.force=T)
net.dia <- graph_from_adjacency_matrix(mx.dia, mode="undirected")

plot(net.dia)

edge_density(net.dia, loops=F)
transitivity(net.dia, type="average")
sum(degree(net.dia, mode="in"))
mean_distance(net.dia)

mod.dia <- cluster_walktrap(net.dia, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.dia), horiz=T)

m.dia <- as.data.frame(sort(membership(mod.dia)))
names(m.dia)[names(m.dia) == "sort(membership(mod.dia))"] <- "Membership"
member.dia <- tibble::rownames_to_column(m.dia, "Bones")
View(member.dia)

mod.dia
modularity(mod.dia)

#bifenestral 2
sheet.neo <- read.csv("network_sheet_neodiapsidal.csv", header=T, row.names=1, sep=";")
mx.neo <- data.matrix(sheet.neo, rownames.force=T)
net.neo <- graph_from_adjacency_matrix(mx.neo, mode="undirected")

plot(net.neo)

edge_density(net.neo, loops=F)
transitivity(net.neo, type="average")
sum(degree(net.neo, mode="in"))
mean_distance(net.neo)

```

```

mod.neo <- cluster_walktrap(net.neo, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.neo), horiz=T)

m.neo <- as.data.frame(sort(membership(mod.neo)))
names(m.neo)[names(m.neo) == "sort(membership(mod.neo))"] <- "Membership"
member.neo <- tibble::rownames_to_column(m.neo, "Bones")
View(member.neo)

mod.neo
modularity(mod.neo)

#bifossil
sheet.bi <- read.csv("network_sheet_bifossil.csv", header=T, row.names=1, sep=";")
mx.bi <- data.matrix(sheet.bi, rownames.force=T)
net.bi <- graph_from_adjacency_matrix(mx.bi, mode="undirected")

plot(net.bi)

edge_density(net.bi, loops=F)
transitivity(net.bi, type="average")
sum(degree(net.bi, mode="in"))
mean_distance(net.bi)

mod.bi <- cluster_walktrap(net.bi, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.bi), horiz=T)

m.bi <- as.data.frame(sort(membership(mod.bi)))
names(m.bi)[names(m.bi) == "sort(membership(mod.bi))"] <- "Membership"
member.bi <- tibble::rownames_to_column(m.bi, "Bones")
View(member.bi)

mod.bi
modularity(mod.bi)

#fossafenestral 1
sheet.fo <- read.csv("network_sheet_neodiapsida2.csv", header=T, row.names=1, sep=";")
mx.fo <- data.matrix(sheet.fo, rownames.force=T)
net.fo <- graph_from_adjacency_matrix(mx.fo, mode="undirected")

plot(net.fo)

edge_density(net.fo, loops=F)
transitivity(net.fo, type="average")
sum(degree(net.fo, mode="in"))
mean_distance(net.fo)

mod.fo <- cluster_walktrap(net.fo, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.fo), horiz=T)

m.fo <- as.data.frame(sort(membership(mod.fo)))
names(m.fo)[names(m.fo) == "sort(membership(mod.fo))"] <- "Membership"
member.fo <- tibble::rownames_to_column(m.fo, "Bones")
View(member.fo)

mod.fo
modularity(mod.fo)

#fossafenestral 2
sheet.kata <- read.csv("network_sheet_kata.csv", header=T, row.names=1, sep=";")
mx.kata <- data.matrix(sheet.kata, rownames.force=T)
net.kata <- graph_from_adjacency_matrix(mx.kata, mode="undirected")

plot(net.kata)

edge_density(net.kata, loops=F)
transitivity(net.kata, type="average")
sum(degree(net.kata, mode="in"))
mean_distance(net.kata)

mod.kata <- cluster_walktrap(net.kata, steps=3)
par(mar=c(2,7,0.1,7))
plot(as.dendrogram(mod.kata), horiz=T)

m.kata <- as.data.frame(sort(membership(mod.kata)))
names(m.kata)[names(m.kata) == "sort(membership(mod.kata))"] <- "Membership"
member.kata <- tibble::rownames_to_column(m.kata, "Bones")
View(member.kata)

mod.kata
modularity(mod.kata)

```