

# Species association networks unveil scale effects in community assemblage mechanisms.

AUTHOR  
Jeanne Tous, Julien Chiquet

## Useful libraries & functions

PLNmodels version should be 1.2.1 or higher.

```
library(dplyr)
```

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

```
filter, lag
```

The following objects are masked from 'package:base':

```
intersect, setdiff, setequal, union
```

```
library(PLNmodels)
```

Warning: package 'PLNmodels' was built under R version 4.5.0

This is package 'PLNmodels' version 1.2.1

Use future::plan(multicore/multisession) to speed up PLNP/PLNmixture/stability\_selectio

```
library(igraph)
```

Attaching package: 'igraph'

The following objects are masked from 'package:dplyr':

```
as_data_frame, groups, union
```

The following objects are masked from 'package:stats':

```
decompose, spectrum
```

The following object is masked from 'package:base':

```
union
```

```
plot_network = function(net,
  type,
  edge.color = c("#F8766D", "#00BFC4"),
  remove.isolated = FALSE,
  node.labels = NULL,
  layout = layout_in_circle) {
  G <- graph_from_adjacency_matrix(net, mode = "undirected", weighted = TRUE, diag = FA
  if (!is.null(node.labels)) {
    igrph::V(G)$label <- node.labels
  } else {
    igrph::V(G)$label <- colnames(net)
  }
  ## Nice nodes
  V.deg <- degree(G)/sum(degree(G))
  igrph::V(G)$label.cex <- V.deg / max(V.deg) + .5
  igrph::V(G)$size <- V.deg * 100
  igrph::V(G)$label.color <- rgb(0, 0, .2, .8)
  igrph::V(G)$frame.color <- NA
  ## Nice edges
  igrph::E(G)$color <- ifelse(igrph::E(G)$weight > 0, edge.color[1], edge.color[2])
  if (type == "support")
    igrph::E(G)$width <- abs(igrph::E(G)$weight)
  else
    igrph::E(G)$width <- 15*abs(igrph::E(G)$weight)
  if (remove.isolated) {
    G <- delete.vertices(G, which(degree(G) == 0))
  }
  plot(G, layout = layout)
  invisible(G)
}
```

## Loading & formatting data

### Regional dataset

```
load("REGIONAL_DATASET_COUNTS.Rdata")
load("REGIONAL_DATASET_COVARIATES.Rdata")
```

```
regional_data <- prepare_data(counts = REGIONAL_DATASET_COUNTS,
  covariates = REGIONAL_DATASET_COVARIATES)
```

Warning in prepare\_data(counts = REGIONAL\_DATASET\_COUNTS, covariates = REGIONAL\_DATASET\_C  
i Function will proceed assuming:  
i - samples are in the same order;  
i - samples are rows of 'counts'.

### Northern Range dataset

```
load("NORTHERN_RANGE_DATASET_COUNTS.Rdata")
load("NORTHERN_RANGE_DATASET_COVARIATES.Rdata")
```

```
northern_range_data <- prepare_data(counts = NORTHERN_RANGE_DATASET_COUNTS,
  covariates = NORTHERN_RANGE_DATASET_COVARIATES)
```

## Running the ZIPLN-network model

We fix the seed for reproducibility.

```
set.seed(1)
```

### Regional dataset

```
regional_model <- ZIPLNnetwork(Abundance ~ 1 + LONGLAT, data = regional_data, control =
```

Initialization...  
Adjusting 30 ZI-PLN with sparse inverse covariance estimation and col specific parameter  
sparsifying penalty = 1.211703  
sparsifying penalty = 0.9548783  
sparsifying penalty = 0.7524883  
sparsifying penalty = 0.5929956  
sparsifying penalty = 0.467308  
sparsifying penalty = 0.3682603  
sparsifying penalty = 0.2902062  
sparsifying penalty = 0.2286959  
sparsifying penalty = 0.1802229  
sparsifying penalty = 0.142024  
sparsifying penalty = 0.1119215  
sparsifying penalty = 0.08819932  
sparsifying penalty = 0.06950515  
sparsifying penalty = 0.05477327  
sparsifying penalty = 0.04316388  
sparsifying penalty = 0.03401513  
sparsifying penalty = 0.0268055  
sparsifying penalty = 0.02112397  
sparsifying penalty = 0.01664667  
sparsifying penalty = 0.01311835  
sparsifying penalty = 0.01033786  
sparsifying penalty = 0.008146715  
sparsifying penalty = 0.006419988  
sparsifying penalty = 0.005059248  
sparsifying penalty = 0.003986921  
sparsifying penalty = 0.003141879  
sparsifying penalty = 0.002475946  
sparsifying penalty = 0.00195116  
sparsifying penalty = 0.001537604  
sparsifying penalty = 0.001211703  
DONE!

We select a penalty using stability selection process StARS. As it stability selection requires a long time to run, we memorize the corresponding penalty and later use it directly to access the model selected with StARS.

```
# regional_model_StARS <- regional_model$getBestModel("StARS")
regional_model_StARS <- regional_model$getModel(0.06950515)
```

Warning in regional\_model\$getModel(0.06950515): No such a model in the collection. Accept  
franks (for PCA)  
sclusters (for mixture models)  
spenalties (for network)  
Returning model with closest value. Requested: 0.06950515 , returned: 0.0695051495418744

### Northern Range dataset

```
northern_range_model <- ZIPLNnetwork(Abundance ~ 1 + LONGLAT, data = northern_range_data
```

Initialization...  
Adjusting 30 ZI-PLN with sparse inverse covariance estimation and col specific parameter  
sparsifying penalty = 1.187862  
sparsifying penalty = 0.9360903  
sparsifying penalty = 0.7376825  
sparsifying penalty = 0.5813279  
sparsifying penalty = 0.4581133  
sparsifying penalty = 0.3610145  
sparsifying penalty = 0.2844961  
sparsifying penalty = 0.2241961  
sparsifying penalty = 0.1766769  
sparsifying penalty = 0.1392296  
sparsifying penalty = 0.1097194  
sparsifying penalty = 0.08646393  
sparsifying penalty = 0.06813758  
sparsifying penalty = 0.05369557  
sparsifying penalty = 0.04231459  
sparsifying penalty = 0.0334586  
sparsifying penalty = 0.02627808  
sparsifying penalty = 0.02070834  
sparsifying penalty = 0.01631913  
sparsifying penalty = 0.01286023  
sparsifying penalty = 0.01013446  
sparsifying penalty = 0.007986421  
sparsifying penalty = 0.00629367  
sparsifying penalty = 0.004959703  
sparsifying penalty = 0.003908475  
sparsifying penalty = 0.00308006  
sparsifying penalty = 0.002427229  
sparsifying penalty = 0.001912769  
sparsifying penalty = 0.001507351  
sparsifying penalty = 0.001187862  
DONE!

We select a penalty using stability selection process StARS. As it stability selection requires a long time to run, we memorize the corresponding penalty and later use it directly to access the model selected with StARS.

```
# northern_range_model_StARS <- northern_range_model$getBestModel("StARS")
northern_range_model_StARS <- northern_range_model$getModel(0.08646393)
```

Warning in northern\_range\_model\$getModel(0.08646393): No such a model in the collection.  
franks (for PCA)  
sclusters (for mixture models)  
spenalties (for network)  
Returning model with closest value. Requested: 0.08646393 , returned: 0.0864639296277481

## Resulting networks

We want to place species the datasets share in the same order for readability.

```
regional_species <- colnames(regional_data$Abundance)
northern_range_species <- colnames(northern_range_data$Abundance)
intersecting_species <- intersect(northern_range_species, regional_species)
```

### Regional network

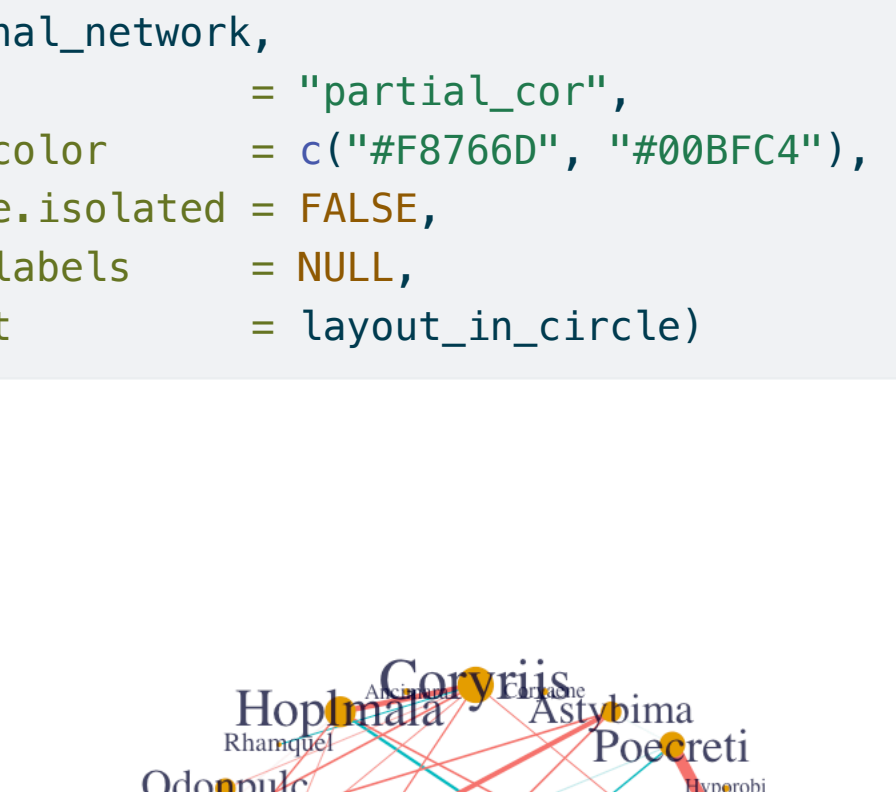
Sorting species names.

```
regional_sorted_species <- c(intersecting_species, regional_species[! regional_species %
```

Extracting the association network from the model.

```
regional_network <- as.matrix(regional_model_StARS$latent_network("partial_cor"))
regional_sorted_indices <- match(regional_sorted_species, colnames(regional_network))
regional_network <- regional_network[regional_sorted_indices, regional_sorted_ind
```

```
plot_network(regional_network,
  type = "partial_cor",
  edge.color = c("#F8766D", "#00BFC4"),
  remove.isolated = FALSE,
  node.labels = NULL,
  layout = layout_in_circle)
```



### Northern Range network

Sorting species names.

```
northern_range_sorted_species <- c(intersecting_species, northern_range_species[! northe
```

```
northern_range_network <- as.matrix(northern_range_model_StARS$latent_network("pa
northern_range_sorted_indices <- match(northern_range_sorted_species, colnames(northern
northern_range_network <- northern_range_network[northern_range_sorted_indices, n
```

```
plot_network(northern_range_network,
  type = "partial_cor",
  edge.color = c("#F8766D", "#00BFC4"),
  remove.isolated = FALSE,
  node.labels = NULL,
  layout = layout_in_circle)
```

