

# Example\_NetGAM\_Network

## Installation

---

We recommend installing NetGAM using the `install_github` function in the `devtools` package.

```
library(devtools)
devtools::install_github("sgleich/NetGAM")

library(NetGAM)
#
install.packages(c("mgcv", "dplyr", "compositions", "stats", "psych", "pulsar", "batchtools", "hug

library(mgcv)
#> Loading required package: nlme
#> This is mgcv 1.8-38. For overview type 'help("mgcv-package")'.
library(pulsar)
library(batchtools)
#>
#> Attaching package: 'batchtools'
#> The following objects are masked from 'package:pulsar':
#>
#>     findConfFile, findTemplateFile
library(dplyr)
#>
#> Attaching package: 'dplyr'
#> The following object is masked from 'package:nlme':
#>
#>     collapse
#> The following objects are masked from 'package:stats':
#>
#>     filter, lag
#> The following objects are masked from 'package:base':
#>
#>     intersect, setdiff, setequal, union
library(psych)
library(huge)
library(stats)
```

## NetGAM::netGAM.df

---

The first function available in the NetGAM package will take a species abundance dataframe (samples as rows and species as columns) and will GAM-transform the species abundance data. These GAM-

transformed species abundance values can be thought of as species abundance values with a reduced influence of time. The GAM-transformed dataframe can be used as input for a downstream network analysis of your choice.

```
# Load in species abundance dataframe. Here, the samples are rows (1-200) and the columns
are species (V1-V20). The species abundance values here are relative abundance values.
data <- system.file("extdata", "output_asvs.csv", package = "NetGAM")
df <- read.csv(data, header=TRUE, row.names=1)

# The netGAM.df function requires two vectors as input: the month of year vector and the
day of time-series vector. We will set up our vectors for input in the function.
moy <- rep(1:12, length=200)
dayOfTs <- 1:200

# Now, we will run the netGAM.df function. The parameter clrt is being set to TRUE here
(default) because we want to CLR-transform our data to account for compositionality.
df.out <- NetGAM::netGAM.df(df, moy, dayOfTs, clrt=TRUE)

# df.out is the GAM-transformed dataframe that can be used in downstream association
network analyses.
```

## NetGAM::netGAM.network (Glasso)

---

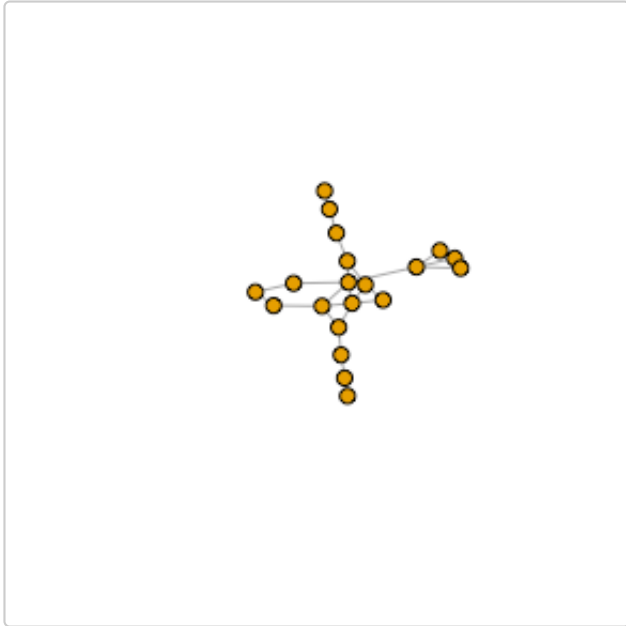
The second function available in the NetGAM package will use GAM-transformed data (i.e. the output of the netGAM.df function) and will use those GAM-transformed data as input for a graphical lasso (glasso), pairwise spearman correlation (scc), or pairwise pearson correlation (pcc) network analysis. The result of the netGAM.network function will be a species-species adjacency matrix in which 1s imply an association between two species and 0s imply no association between two species. This adjacency matrix can then be used to graph the network in the igraph package.

```
# Run the netGAM.network function using the output of the netGAM.df function (i.e.
df.out) as input. We will first run a glasso network (default) and then we can try
running a scc network after.
glasso.net <- NetGAM::netGAM.network(df.out, method="glasso")
#> Conducting the nonparanormal (npr) transformation via shrunkun ECDF....done.
#> Sourcing configuration file
'/Users/samanthagleich/Library/R/3.6/library/pulsar/config/batchtools.conf.R' ...
#> Created registry in
'/var/folders/5n/08h315tn3n12xmlntsd5w2s80000gn/T/RtmpciBN6k/registry17e1c2d0a908' using
cluster functions 'Interactive'
#> Adding 51 jobs ...
#> Submitting 51 jobs in 51 chunks using cluster functions 'Interactive' ...
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
glasso.graph <- graph_from_adjacency_matrix(glasso.net,mode="undirected")
plot(glasso.graph,vertex.label=NA)

```



## NetGAM::netGAM.network (SCC) We can also run the netGAM.network function to build a GAM-transformed SCC network.

```

# Run the netGAM.network function using the output of the netGAM.df function (i.e.
df.out) as input. Here, will will run a scc network.
scc.net <- NetGAM::netGAM.network(df.out,method="scc",pvalue=0.05)
#> Conducting the nonparanormal (npn) transformation via shrunkun ECDF....done.
library(igraph)
scc.graph <- graph_from_adjacency_matrix(scc.net,mode="undirected")
plot(scc.graph,vertex.label=NA)

```

