

Genetic Effects of a Foundation Tree on Networks of Leaf Modifying Arthropods

Libraries

```
libs <- c("ComGenR", "coNet")
if (!(all(libs %in% installed.packages()[, 1]))){
  devtools::install_github("ECGen/ComGenR")
  devtools::install_github("ECGen/coNet", ref = "master")
}
sapply(libs, require, character.only = TRUE)
```

```
## ComGenR    coNet
##      TRUE      TRUE
```

Species/Functional Groups

- PB = P. betae
- pb.pred = parasitic fly
- pb.abort =
- edge.fold = sawfly
- pinch =
- mid.miner = lep
- edge.miner = lep
- tip.miner = lep
- fish.ey = ??
- tier = lep
- thrips = Thysanura
- chew.holes = ??
- chomp = large herbivore
- scrape = radula
- chew.edge = lep?

Pre-processing Data

Load datasets

```
pit <- read.csv('../data/acn/arth_cooc_PIT_Lau.csv')
pit[is.na(pit)] <- 0
```

Should any genotypes be removed?

```
rm.1007 <- TRUE
```

Restrict main analysis to live leaves only

```
### separating out "fungal" necrosis
### necrosis <- pit$fungal
pit <- pit[,colnames(pit) != 'fungal']
```

Remove 1007

```
if (rm.1007){pit <- pit[pit[, "geno"] != "1007", ]}
```

Remove mite

```
pit <- pit[, colnames(pit) != "mite"]
```

combine pb

```
pb <- pit$pb.upper + pit$pb.lower + pit$pb.woody
pb.pred <- pit$pb.pred + pit$pb.woody.pred + pit$pb.hole
pb.abort <- pit$pb.abort
pit <- pit[,!(grepl('pb',colnames(pit)))]
pit <- data.frame(pit,pb.abort,pb.pred,pb)
pit <- data.frame(pit[,1:6],pit[,ncol(pit):7])
tree.info <- paste(pit[, "tree"], pit[, "geno"], pit[, "leaf.type"])
tree.arth <- pit[, 7:ncol(pit)]
tree.arth <- split(tree.arth, tree.info)
```

community matrix

```
com.acn <- do.call(rbind, lapply(tree.arth, function(x) apply(x, 2, sum)))
```

tree level networks for arthropods

```
cn.acn <- lapply(tree.arth, coNet, ci.p = 95, cond = TRUE)
```

tree net distances

```
d.cn.acn <- netDist(cn.acn)
```

network stats

```

l.cn.acn <- do.call(rbind, lapply(cn.acn, enaR:::structure.statistics))[, "L"]
cen.cn.acn <- unlist(lapply(cn.acn,
                           function(x)
                             sna::centralization(x, FUN = sna::degree, normalize = FALSE)),
                      )
nm.cn.acn <- data.frame(L = l.cn.acn, C = cen.cn.acn)

```

tree info

```

acn.dat <- data.frame(do.call(rbind, strsplit(names(cn.acn), split = " ")))
names(acn.dat) <- c("tree", "geno", "leaf.type")

```

Limit senscent trees with greater than 20 leaves

```

n.sen <- unlist(lapply(tree.arth[acn.dat[, "leaf.type"] == "sen"], nrow))
n.live <- unlist(lapply(tree.arth[acn.dat[, "leaf.type"] == "live"], nrow))
sen.trees <- tree.arth[names(tree.arth) %in% names(n.sen)[n.sen >= 20]]
sen.dat <- data.frame(do.call(rbind, strsplit(names(sen.trees), split = " ")))
colnames(sen.dat) <- c("tree", "geno", "leaf.type")
com.sen <- do.call(rbind, lapply(sen.trees, function(x) apply(x, 2, sum)))

```

Network Ordination

```

if (file.exists("../results/nms_cn_acn.rda")){
  nms.cn.acn <- dget("../results/nms_cn_acn.rda")
}else {
  set.seed(1234)
  ## nms.cn.acn <- nmds(d.cn.acn, 2, 2)
  nms.cn.acn <- nmds(netDist(cn.acn[grepl("live", names(cn.acn))]), 2, 2)
  dput(nms.cn.acn, "../results/nms_cn_acn.rda")
}
ord.cn.acn <- nmds.min(nms.cn.acn)
vec.com.acn <- envfit(ord.cn.acn,
                     com.acn[grepl("live", rownames(com.acn)), apply(com.acn, 2, sum) > 10])
vec.nm.acn <- envfit(ord.cn.acn, nm.cn.acn[grepl("live", rownames(nm.cn.acn)), ])

cn.sen <- lapply(sen.trees, coNet, ci.p = 95, cond = TRUE)

```

Main Results

Modularity of bipartite networks

```

computeModules(com.acn[grepl("live", rownames(com.acn)), ])
computeModules(com.acn[grepl("sen", rownames(com.acn)), ])

```