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

Species/Functional Groups

1. PB = P. betae
2. pb.pred = parasitic fly
3. pb.abort =
4. edge.fold = sawfly
5. pinch =
6. mid.miner = lep
7. edge.miner = lep
8. tip.miner = lep
9. fish.eye = ??
10. tier = lep
11. thrips = Thysanura
12. chew.holes = ??
13. chomp = large herbivore
14. scrape = radula
15. chew.edge = lep?

Pre-processing Data

Load datasets

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

Limit to live leaves only

```
pit <- pit[pit[, "leaf.type"] == "live", ]
```

Remove necrosis data

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

Remove Genotype 1007

```
pit <- pit[pit[, "geno"] != "1007", ]
```

Remove mite

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

Combine gall variants

```
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])
```

Number species

```
sp.name <- 1:length(((1:ncol(pit))[colnames(pit) == "pb" ]):ncol(pit))
names(sp.name) <- colnames(pit)[((1:ncol(pit))[colnames(pit) == "pb" ]):ncol(pit)]
colnames(pit)[((1:ncol(pit))[colnames(pit) == "pb" ]):ncol(pit)] <-
  1:length(((1:ncol(pit))[colnames(pit) == "pb" ]):ncol(pit))
```

```
tree.info <- paste(pit[, "tree"], pit[, "geno"], pit[, "leaf.type"])
tree.arth <- pit[, 7:ncol(pit)]
tree.arth <- split(tree.arth, tree.info)
```

Create the community matrix

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

Tree level network models

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

Tree network distances

```
d.cn.acn <- netDist(cn.acn, method = "euclidean")
```

Calculate network metrics

```
                                        # Links
l.cn.acn <- do.call(rbind, lapply(cn.acn, enaR:::structure.statistics))[, "L"]
                                        # Centrality
cen.cn.acn <- unlist(lapply(cn.acn, function(x)
    sna::centralization(x, FUN = sna::degree, normalize = FALSE)))
                                        # Modularity
```

```
mod.cn.acn <- numeric(0)
for (i in 1:length(cn.acn)){
  if (sum(abs(sign(cn.acn[[i]]))) <= 3){
    mod.cn.acn[i] <- 0
  }else{
    mod.cn.acn[i] <- slot(computeModules(cn.acn[[i]]), "likelihood")
  }
}
                                        # Wrap into df
nm.cn.acn <- data.frame(Links = l.cn.acn, Centralization = cen.cn.acn, Modularity = mod.cn.acn)
```

Tree info compilation

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

Network ordination

```
if (file.exists("../results/nms_cn_acn.rda")){
    nms.cn.acn <- dget("../results/nms_cn_acn.rda")
}else {
    set.seed(123)
    ##    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.cn.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)
```

Leaf sample size relativization

```
lsr <- unlist(lapply(tree.arth, nrow))
com.acn.lsr <- com.acn / lsr
```

Main Results

Note: genotype 1007 is removed because it only has one rep

Genotype replication

```
table(acn.dat[, "geno"])
```

```
##
## 1000 1008 1017 1023   11  996   T6
##    6    5    5    5    5    5    4
```

Total abundance

```
abund <- apply(com.acn, 1, sum)
reml.abund.acn <- lme4::lmer(I(abund^(1/1)) ~ (1 | geno),
                             data = acn.dat,
                             REML = TRUE)
p.reml.abund.acn <- RLRsim::exactRLRT(reml.abund.acn)
p.reml.abund.acn
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 4.6317, p-value = 0.0106
```

Richness

```
rich <- apply(com.acn, 1, function(x) sum(sign(x)))
reml.rich.acn <- lme4::lmer(I(rich^(1/1)) ~ (1 | geno),
                            data = acn.dat,
                            REML = TRUE)
p.reml.rich.acn <- RLRsim::exactRLRT(reml.rich.acn)
p.reml.rich.acn
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 1.4103, p-value = 0.0961
```

Community Similarity

```
rel.com.acn <- rel(com.acn)
rel.com.acn[is.na(rel.com.acn)] <- 0
                                        # PerMANOVA
                                        # Un-relativized
set.seed(12234)
vegan::adonis2(com.acn ~ geno,
```

```
              data = acn.dat,
              strata = acn.dat[, "tree"],
              perm = 10000, sqrt.dist = TRUE, mrank = TRUE)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = com.acn ~ geno, data = acn.dat, permutations = 10000, sqrt.dist = TRUE, stra
##          Df SumOfSqs     R2      F   Pr(>F)
## geno      6   1.8239 0.2323 1.4121 0.009499 **
## Residual 28   6.0277 0.7677
## Total    34   7.8516 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
                                    # Relativized
set.seed(12234)
vegan::adonis2(rel.com.acn ~ geno,
              data = acn.dat,
              strata = acn.dat[, "tree"],
              perm = 10000, sqrt.dist = TRUE, mrank = TRUE)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = rel.com.acn ~ geno, data = acn.dat, permutations = 10000, sqrt.dist = TRUE,
##          Df SumOfSqs      R2      F Pr(>F)
## geno      6   2.3942 0.23436 1.4285 0.0029 **
## Residual 28   7.8217 0.76564
## Total    34  10.2159 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Network similarity

```
set.seed(12234)
vegan::adonis2(netDist(cn.acn) ~ geno,
              data = acn.dat,
              strata = acn.dat,
              sqrt.dist = FALSE, mrank = TRUE)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## vegan::adonis2(formula = netDist(cn.acn) ~ geno, data = acn.dat, sqrt.dist = FALSE, strata = acn.dat
##          Df SumOfSqs      R2      F Pr(>F)
## geno      6   127.79 0.40888 3.2279  0.006 **
```

```
## Residual 28   184.75 0.59112
## Total    34   312.54 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
                                        # PB as a predictor
pb <- com.acn[, sp.name["pb"]]
set.seed(12234)
vegan::adonis2(netDist(cn.acn) ~ geno + pb,
               data = acn.dat,
               sqrt.dist = FALSE, mrank = TRUE)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 999
##
## vegan::adonis2(formula = netDist(cn.acn) ~ geno + pb, data = acn.dat, sqrt.dist = FALSE, mrank = TRU
##          Df SumOfSqs      R2      F Pr(>F)
## geno      6  127.790 0.40888 4.2273  0.002 **
## pb        1   48.713 0.15586 9.6685  0.002 **
## Residual 27  136.035 0.43526
## Total    34  312.538 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
                                        # Does genotype predict PB?
reml.pb <- lme4::lmer(I(pb^(1/1)) ~ (1 | geno),
                          data = acn.dat,
                          REML = TRUE)
p.reml.pb <- RLRsim::exactRLRT(reml.pb)
p.reml.pb
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 6.7474, p-value = 0.0035
```

Network metrics

```
                                        # number of links
reml.l.acn <- lme4::lmer(I(Links^(1/1)) ~ (1 | geno),
                          data = acn.dat,
                          REML = TRUE)
p.reml.l.acn <- RLRsim::exactRLRT(reml.l.acn)
p.reml.l.acn
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
```

```
## data:
## RLRT = 7.197, p-value = 0.0034
```

```
                                        # centralization
reml.cen.acn <- lme4::lmer(I(Centralization^(1/2)) ~ (1 | geno),
                           data = acn.dat,
                           REML = TRUE)
p.reml.cen.acn <- RLRsim::exactRLRT(reml.cen.acn)
p.reml.cen.acn
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 2.3608, p-value = 0.0511
```

```
                                        # Modularity
reml.mod.acn <- lme4::lmer(I(Modularity^(1/1)) ~ (1 | geno),
                           data = acn.dat,
                           REML = TRUE)
p.reml.mod.acn <- RLRsim::exactRLRT(reml.mod.acn)
p.reml.mod.acn
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 14.702, p-value < 2.2e-16
```

Proportion of PB singles, doubles and triples

```
n.leaf <- unlist(lapply(tree.arth, nrow))
pb1 <- unlist(lapply(tree.arth, function(x) sum(sign(x[, "pb"] == 1))))
pb2 <- unlist(lapply(tree.arth, function(x) sum(sign(x[, "pb"] == 2))))
pb3 <- unlist(lapply(tree.arth, function(x) sum(sign(x[, "pb"] == 3))))
pb4 <- unlist(lapply(tree.arth, function(x) sum(sign(x[, "pb"] == 4))))
pb.1 <- unlist(lapply(tree.arth, function(x) sum(sign(x[, "pb"] == 1))))
pb1.d <- pb1[grepl("live", names(pb1))]/n.leaf[grepl("live", names(n.leaf))] -
    pb1[grepl("sen", names(pb1))]/n.leaf[grepl("sen", names(n.leaf))]
pb2.d <- pb2[grepl("live", names(pb2))]/n.leaf[grepl("live", names(n.leaf))] -
    pb2[grepl("sen", names(pb2))]/n.leaf[grepl("sen", names(n.leaf))]
pb3.d <- pb3[grepl("live", names(pb3))]/n.leaf[grepl("live", names(n.leaf))] -
    pb3[grepl("sen", names(pb3))]/n.leaf[grepl("sen", names(n.leaf))]

                                        # pb1
RLRsim::exactRLRT(
    lme4::lmer(I(pb1.d[acn.dat[, "tree"] %in% sen.dat[, "tree"]]) ~ (1 | geno),
               data = acn.dat[acn.dat[, "tree"] %in% sen.dat[, "tree"], ],
               REML = TRUE)
)
```

```
                                        # pb2
RLRsim::exactRLRT(
    lme4::lmer(I(pb2.d[acn.dat[, "tree"] %in% sen.dat[, "tree"]]) ~ (1 | geno),
               data = acn.dat[acn.dat[, "tree"] %in% sen.dat[, "tree"], ],
               REML = TRUE)
)
                                        # pb3
RLRsim::exactRLRT(
    lme4::lmer(I(pb3.d[acn.dat[, "tree"] %in% sen.dat[, "tree"]]) ~ (1 | geno),
               data = acn.dat[acn.dat[, "tree"] %in% sen.dat[, "tree"], ],
               REML = TRUE)
)
```

No genotype effect, so all trees are tested together

```
t.test(pb1.d)
t.test(pb2.d)
t.test(pb3.d)
```

Leaf type indicator species

```
isa.type <- labdsv::indval(com.acn.lsr, acn.dat[, "leaf.type"])
summary(isa.type)
```

Genotype indicator species

```
isa.geno <- labdsv::indval(com.acn.lsr, acn.dat[, "geno"])
summary(isa.geno)
```

Genotype-species clusters

```
gsc <- computeModules(com.acn.lsr)
slot(gsc, "likelihood")
```

Cluster based on PB similarity

Modularity of bipartite networks

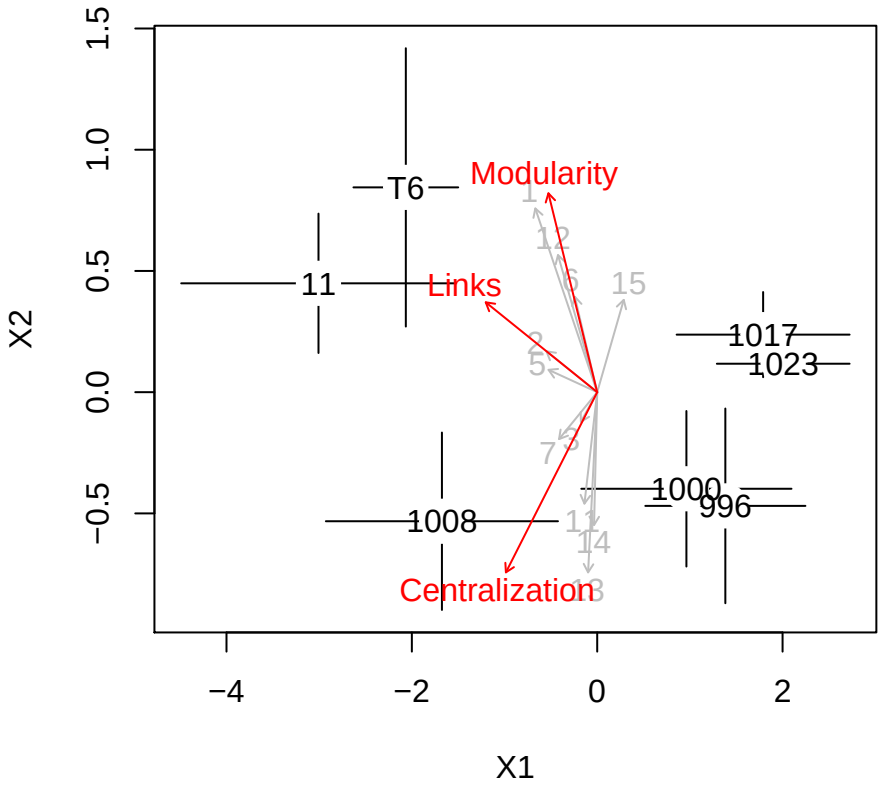
```
bipartite::computeModules(com.acn[grepl("live",
                                        rownames(com.acn)), ])
```

Plots

Main Results

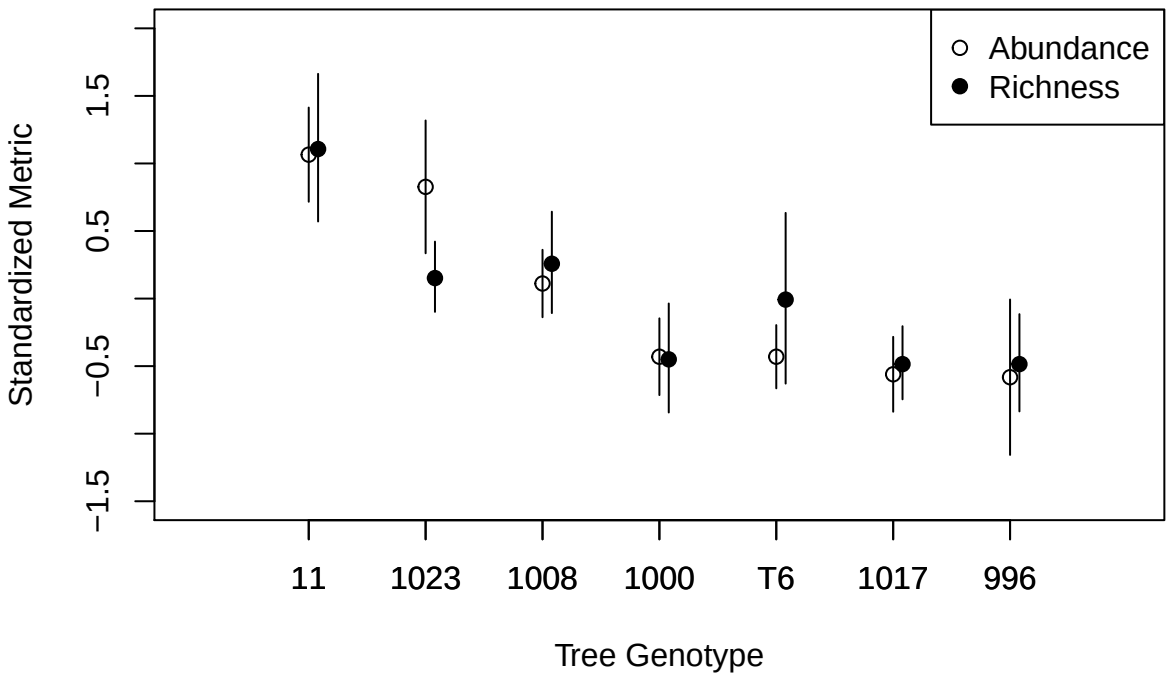
Network Ordination

```
coord <- ch.plot(ord.cn.acn, g = acn.dat[, "geno"],
                 cex = 3, mu.pch = 19, pt.col = "white")
text(coord, labels = rownames(coord))
plot(vec.cn.acn, col = "grey")
plot(vec.nm.acn, col = "red")
```



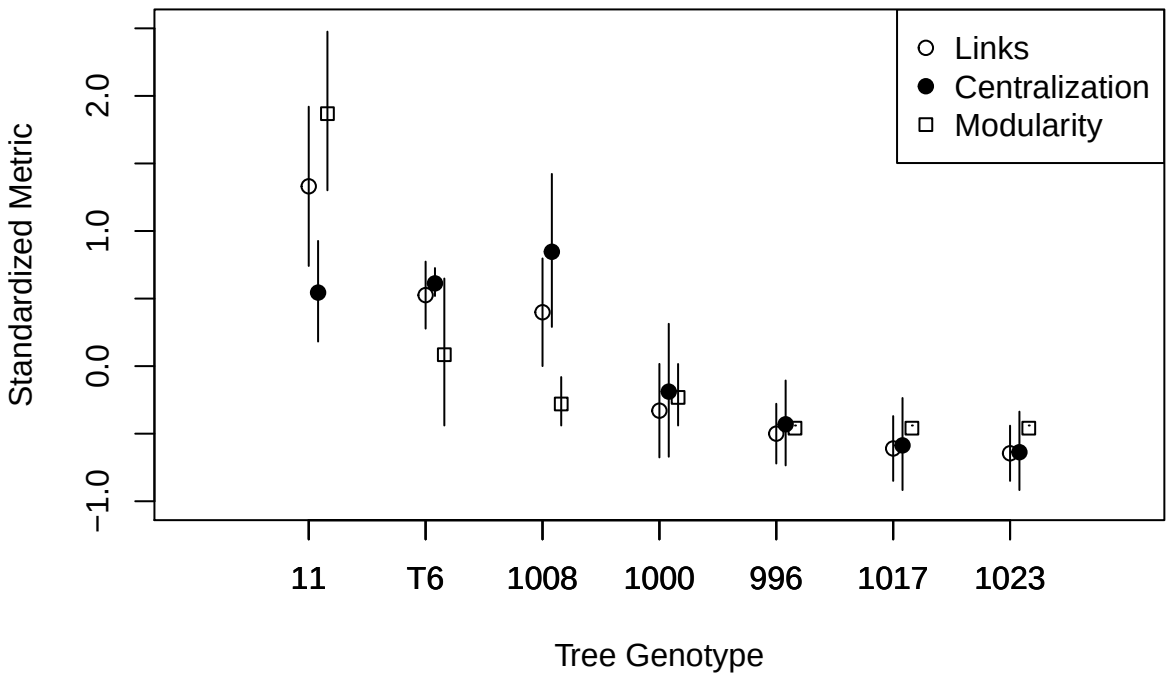
Abundance and richness by genotype

```
                                  # abundance and richness
mdc.plot(acn.dat[, "geno"], abund, ylim = c(-1.5, 2.0),
         xlab = "Tree Genotype", ylab = "Standardized Metric", pch = 1,
         ord = order(tapply(abund, acn.dat[, "geno"], mean), decreasing = TRUE))
mdc.plot(acn.dat[, "geno"], rich, add = TRUE, pch = 19, xjit = 0.01,
         ord = order(tapply(abund, acn.dat[, "geno"], mean), decreasing = TRUE))
legend("topright", legend = c("Abundance", "Richness"), pch = c(1, 19), bty = "none")
```



Network metrics by genotype

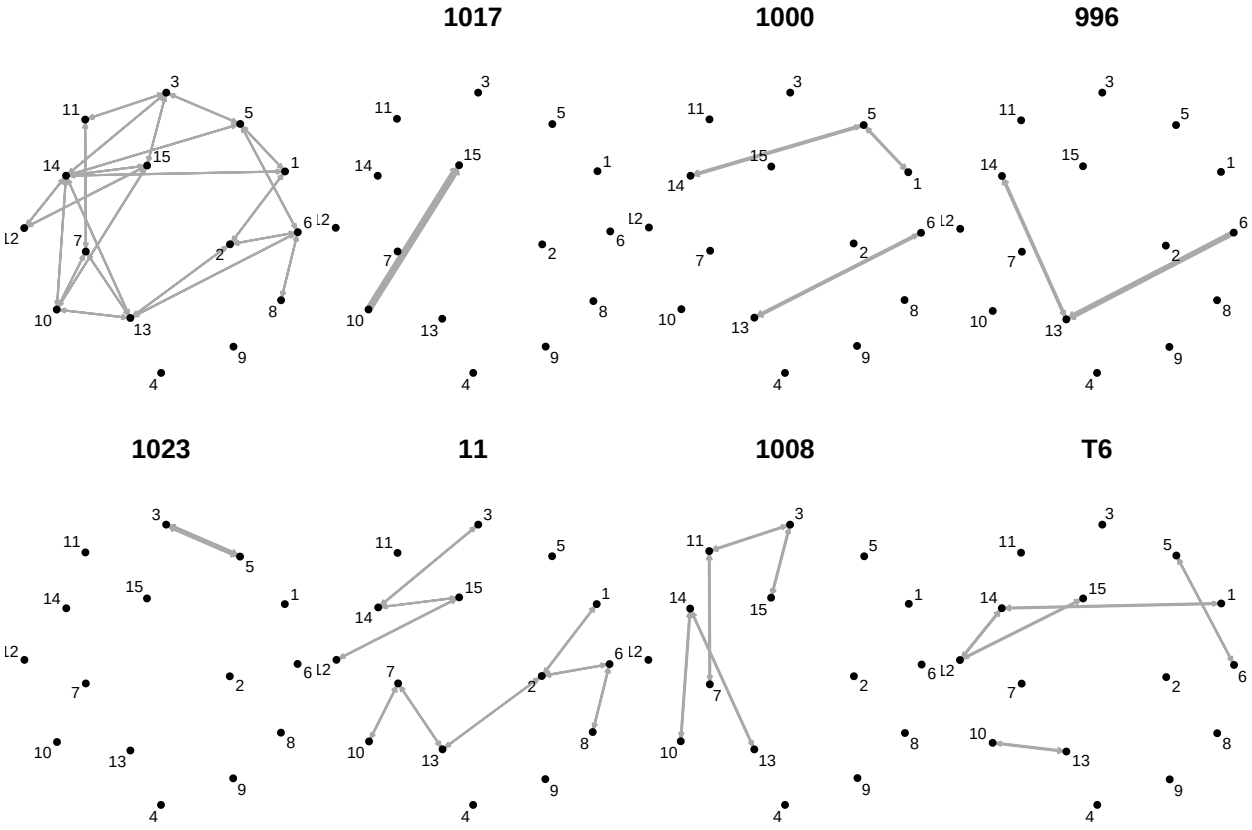
```
mdc.plot(acn.dat[, "geno"], acn.dat[, "Links"], ylim = c(-1, 2.5),
         pch = 1,
         xlab = "Tree Genotype", ylab = "Standardized Metric",
         ord = order(tapply(acn.dat[, "Links"], acn.dat[, "geno"], mean),
                     decreasing = TRUE))
mdc.plot(acn.dat[, "geno"], acn.dat[, "Centralization"],
         add = TRUE, pch = 19, xjit = 0.01,
         ord = order(tapply(acn.dat[, "Links"], acn.dat[, "geno"], mean),
                     decreasing = TRUE))
mdc.plot(acn.dat[, "geno"], acn.dat[, "Modularity"],
         add = TRUE, pch = 22, xjit = 0.02, col = "darkgrey",
         ord = order(tapply(acn.dat[, "Links"], acn.dat[, "geno"], mean),
                     decreasing = TRUE))
legend("topright", legend = c("Links", "Centralization", "Modularity"),
       pch = c(1, 19, 22), bty = "none")
```



Network Plots

```
                                        # By Genotype
par(mfrow = c(1, 2))
set.seed(12234)
net.col <- sign(netMean(cn.acn))
net.col[net.col == -1] <- 2
net.col[net.col == 1] <- "darkgrey"
# pdf(file = "../results/acn_live_nets.pdf", width = 9, height = 9)
par(mfrow = c(2, 4), mar = c(0, 0, 1, 0))
coord <- gplot(abs(netMean(cn.acn[acn.dat[, "leaf.type"] == "live"])),
               gmode = "digraph",
               displaylabels = TRUE,
               edge.lwd = (abs(netMean(cn.acn[acn.dat[, "leaf.type"] == "live"]))) * 10,
               edge.col = net.col,
               vertex.col = "black",
               vertex.cex = 0.5,
               arrowhead.cex = 0.5,
               label.cex = 0.75,
               main = "")
for (i in unique(acn.dat[, "geno"])){
    gplot(abs(netMean(cn.acn[acn.dat[, "geno"] == i &
                             acn.dat[, "leaf.type"] == "live"])),
          coord = coord,
```

```
          gmode = "digraph",
          displaylabels = TRUE,
          edge.lwd = (abs(netMean(cn.acn[acn.dat[, "geno"] == i &
                                           acn.dat[, "leaf.type"] == "live"]))) * 10,
          edge.col = net.col,
          vertex.col = "black",
          vertex.cex = 0.5,
          arrowhead.cex = 0.5,
          label.cex = 0.75,
          main = i)
}
```



Bipartite representation

```
plotModuleWeb(gsc)
```

Tables

```
vf.cn.tab <- data.frame(vec.cn.acn$vectors[c("r", "pvals")])
vf.cn.tab <- data.frame(species =
                          names(sp.name[sp.name %in% rownames(vf.cn.tab)]),
                        vf.cn.tab)
```

```
vf.cn.tab <- vf.cn.tab[order(vf.cn.tab[, "r"], decreasing = TRUE),]
print(xtable::xtable(vf.cn.tab,
                     caption = "Table of leaf modifier vector analyses.",
                     digits = 3))
```

% latex table generated in R 3.6.0 by xtable 1.8-4 package % Tue May 21 11:05:23 2019

	species	r	pvals
1	pb	0.391	0.003
13	chomp	0.215	0.029
12	chew.holes	0.191	0.029
2	pb.pred	0.126	0.126
14	scrape	0.115	0.137
5	pinch	0.109	0.145
11	thrips	0.088	0.221
15	chew.edge	0.087	0.250
6	mid.miner	0.081	0.261
7	edge.miner	0.079	0.268
3	pb.abort	0.018	0.753

Table 1: Table of leaf modifier vector analyses.