LCN: Lichen interaction network study

MK Lau

```
### Loading data objects
### Data notes:
## Trees were removed from the analysis genotype RL6 and N1.31
## No physciods
## Lecanoras merged
# Loading data
xgal.size <- read.csv("../data/lcn/ONC_Xgal_SizeData_May2011.csv")
garden.data <- read.csv("../data/lcn/LCO_data_ONC_PIT.csv")
# rm genotype RL6 and N1.31
garden.data <- garden.data[garden.data$Geno != "RL6", ]
garden.data <- garden.data[garden.data$Tree != "N1.31", ]
# separate onc
garden.data[, 1] <- as.character(garden.data[, 1])
g1 <- substr(garden.data[, 1], 2, 2)
g1[g1 != "P"] <- "onc"
onc <- garden.data[g1 == "onc", ]
colnames(onc)[which(colnames(onc) == "Ls")] <- "Lh"
pit <- garden.data[g1 == "P", ]
# tree overlap between years
unique(onc$Tree[onc$Year == "2010"]) %in%
  unique(onc$Tree[onc$Year == "2011"])
```

```
##  [1] TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
## [15] TRUE TRUE TRUE TRUE
```

```
unique(onc$Tree[onc$Year == "2011"]) %in%
  unique(onc$Tree[onc$Year == "2010"])
```

```
##  [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [23] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [34] FALSE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE  TRUE  TRUE  TRUE
## [45]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [56]  TRUE  TRUE
```

```
# Checking the data
if (!(all(table(onc[, 1]) == 100))) {
  for (i in 1:1000) {
    print("Warning: check input data!!!")
  }
}
# Separate trees
# onc
colnames(onc)[7:ncol(onc)] <- substr(colnames(onc)[7:ncol(onc)], 1, 2)
onc.q <- split(onc, paste(onc[, 1], onc[, 2]))
onc.q <- lapply(onc.q, function(x) x[, 7:ncol(x)])
# pit
colnames(pit)[7:ncol(pit)] <- substr(colnames(pit)[7:ncol(pit)], 1, 2)
pit.q <- split(pit, paste(pit[, 1], pit[, 2]))
pit.q <- lapply(pit.q, function(x) x[, 7:ncol(x)])
```

```r
# Get genotype
onc.geno <- unlist(sapply(
  names(onc.q),
  function(x) strsplit(x, split = " ")[[1]][2]
))
pit.geno <- unlist(sapply(
  names(pit.q),
  function(x) strsplit(x, split = " ")[[1]][2]
))
# Xgal size data
xgs <- xgal.size[-1:-7, -(ncol(xgal.size) - 1):-ncol(xgal.size)]
xgs.cols <- xgal.size[7, -(ncol(xgal.size) - 1):-ncol(xgal.size)]
colnames(xgs) <- gsub("\\#", "", as.character(unlist(xgs.cols)))
xgs <- xgs[, 1:13]
xgs <- apply(xgs, 2, gsub, pattern = "\\,", replacement = "")
xgs.dim <- xgs[, "Measurement"]
xgs.geno <- xgs[, "Genotype"]
xgs.tree <- xgs[, "Tree"]
xgs <- xgs[, grep("Thallus", colnames(xgs))]
# fix genotypes
# t6
xgs.geno[grep("T6", xgs.geno)] <- "T6"
xgs.geno[grep("H10", xgs.geno)] <- "H-10"
# Coercing to numeric
xgs <- apply(xgs, 2, as.numeric)
# Dealing with NA values
xgs.geno <- xgs.geno[grep("Dimension", xgs.dim)]
xgs.tree <- xgs.tree[grep("Dimension", xgs.dim)]
xgs <- xgs[grep("Dimension", xgs.dim), ]
xgs.dim <- xgs.dim[grep("Dimension", xgs.dim)]
# Convert to cm
xgs <- xgs * 0.1
xgs.ellipse <- pi * xgs[xgs.dim == "Dimension 1", ] *
  xgs[xgs.dim == "Dimension 2", ]
xgs.geno <- xgs.geno[xgs.dim == "Dimension 1"]
xgs.tree <- xgs.tree[xgs.dim == "Dimension 1"]
# package all xgs related data
xgs.data <- data.frame(
  tree = xgs.tree, geno = xgs.geno,
  mean.thallus = apply(xgs.ellipse, 1,
    mean,
    na.rm = TRUE
  ),
  median.thallus = apply(xgs.ellipse, 1,
    median,
    na.rm = TRUE
  ),
  xgs.ellipse
)
# remove trees not done (i.e. all NA)
xgs.data <- xgs.data[apply(
  xgs.data[, grep(
    "Thallus",
```

```
    colnames(xgs.data)
  )], 1,
  function(x) !(all(is.na(x)))
), ]
# Roughness in the Garden
rough <- read.csv("../data/lcn/ONC_raw_roughness.csv")
# Isolate roughness
rough <- rough[, 1:5]
# Isolate north quadrats
rough <- rough[grepl("North", rough[, 3]), ]
# Average roughness
avg.rough <- tapply(rough[, 5], rough[, 1], mean)
r.tree <- names(avg.rough)
r.tree <- sub("-", "\\.", r.tree)
r.tree <- sub("\\.0", "\\.", r.tree)
names(avg.rough) <- r.tree
# match roughness to to ses values
load("../data/lcn/lcn_onc_ses.rda")
onc.ses <- unlist(os[, 1])
onc.ses[is.na(onc.ses)] <- 0
names(onc.ses) <- rownames(os)
if (!(all(names(onc.ses) == names(onc.q)))) {
  print("Holy crap!")
}
ses.tree <- as.character(sapply(
  names(onc.ses),
  function(x) unlist(strsplit(x, split = " "))[1]
))
onc.rough <- avg.rough[match(ses.tree, r.tree)]
if (!(all(ses.tree == names(onc.rough)))) {
  print("Holy Crap!")
}

# Lichen Network Models
# onc
cn.onc <- lapply(split(onc[, -1:-6], onc[, "Tree"]), coNet,
  ci.p = 95
)
cn.sign.onc <- lapply(split(onc[, -1:-6], onc[, "Tree"]), coNet,
  ci.p = 95
)
cn.d.onc <- distNet(cn.onc, method = "euclidean")
# pit
cn.pit <- lapply(
  split(pit[, -1:-6], pit[, "Tree"]),
  coNet, ci.p = 95
)
cn.sign.pit <- lapply(
  split(pit[, -1:-6], pit[, "Tree"]),
  coNet, ci.p = 95
)
cn.d.pit <- distNet(cn.pit, method = "bc")
# genotype means and mean distances
```

```
onc.tree <- do.call(rbind, strsplit(names(onc.geno), " "))[, 1]
cn.mu.onc <- list()
for (i in 1:length(unique(onc.geno))) {
  cn.mu.onc[[i]] <- meanNet(cn.onc[onc.geno == unique(onc.geno)[i]])
}
names(cn.mu.onc) <- unique(onc.geno)
cn.mu.d.onc <- distNet(cn.mu.onc, method = "bc")
# mean bark roughness calculations
prb.mu.onc <- tapply(onc.rough, onc.geno, mean)
prb.mu.d.onc <- dist(prb.mu.onc)
# network statistics
ns.onc <- lapply(lapply(cn.onc, function(x) {
  abs(sign(x))
}), enaR:::structure.statistics)
ns.onc <- do.call(rbind, ns.onc)
# Ratio P / N
ns.rpn <- unlist(lapply(cn.onc, function(x) {
  mean(x[x > 0]) / mean(x[x < 0])
}))

# modularity
cn.mod.onc <- matrix(nrow = length(cn.onc), ncol = 2)
for (i in 1:length(cn.onc)) {
  if (sum(sign(cn.onc[[i]])) >= 3) {
    mod.tmp <- computeModules(cn.onc[[i]])
    cn.mod.onc[i, 1] <- slot(mod.tmp, "likelihood")
    cn.mod.onc[i, 2] <- nrow(slot(mod.tmp, "modules")) - 1
  } else {
    cn.mod.onc[i] <- NA
  }
}
cn.mod.onc[is.na(cn.mod.onc)] <- 0
names(cn.mod.onc) <- c("mod.lik", "mod.n")
# graph level centralization
dcen.onc <- unlist(lapply(cn.onc, function(x) {
  sna::centralization(x, FUN = sna::degree, normalize = FALSE)
}))
onc.ns <- cbind(
  ns.onc,
  Cen = dcen.onc,
  mod.lik = cn.mod.onc[, 1], mod.n = cn.mod.onc[, 2]
)
if (!(all(onc.tree == names(cn.onc)))) {
  print("Danger Will Robinson!")
}
# species centralities
cen.spp <- lapply(cn.onc[names(cn.onc) %in% na.omit(onc.dat)$tree.id],
  sna::degree,
  rescale = FALSE
)
cen.spp <- do.call(rbind, cen.spp)
colnames(cen.spp) <- colnames(cn.onc[[1]])
# Community data
```

```r
onc.com <- do.call(rbind, lapply(onc.q, function(x) apply(x, 2, sum)))
onc.R <- apply(sign(onc.com), 1, sum)
onc.H <- vegan::diversity(onc.com)
onc.com.gm <- apply(onc.com, 2,
  function(x, g) tapply(x, g, mean),
  g = onc.geno
)
onc.com.gm.rel <- apply(onc.com.gm, 2, function(x) x / max(x))
onc.com.rel <- apply(onc.com, 2, function(x) x / max(x))
onc.com.rel <- cbind(onc.com.rel,
  ds = rep(
    min(onc.com.rel[onc.com.rel != 0]) / 1000,
    nrow(onc.com.rel)
  )
)
onc.com <- cbind(onc.com,
  ds = rep(
    min(onc.com[onc.com != 0]) / 1000,
    nrow(onc.com)
  )
)
# pit genotype mean community
pit.com <- do.call(rbind, lapply(pit.q, function(x) apply(x, 2, sum)))
pit.com.gm <- apply(pit.com, 2,
  function(x, g) tapply(x, g, mean),
  g = pit.geno
)
pit.com.gm.rel <- apply(pit.com.gm, 2, function(x) x / max(x))
pit.com.gm.rel[is.na(pit.com.gm.rel)] <- 0
# Lichen community metrics
# Percent Total Cover
ptc.onc <- unlist(lapply(
  onc.q,
  function(x) {
    sum(apply(
      x, 1,
      function(x) sign(sum(x))
    ))
  }
))
# Species richness
spr.onc <- apply(
  onc.com[, colnames(onc.com) != "ds"], 1,
  function(x) sum(sign(x))
)
# Diversity
spd.onc <- diversity(onc.com[, colnames(onc.com) != "ds"])
# Evenness
spe.onc <- spd.onc / log(specnumber(onc.com[, colnames(onc.com) != "ds"]))
spe.onc[is.na(spe.onc)] <- 0

## "mean" distance matrices
cn.mu.d <- as.matrix(cn.mu.d.onc)
```

```r
prb.mu.d <- as.matrix(prb.mu.d.onc)
prb.mu.d <- prb.mu.d[
  match(rownames(cn.mu.d), rownames(prb.mu.d)),
  match(rownames(cn.mu.d), rownames(prb.mu.d))
]
prb.mu.d <- as.dist(prb.mu.d)
onc.com.mu <- apply(onc.com[, -ncol(onc.com)], 2,
  function(x, g) tapply(x, g, mean),
  g = onc.geno
)
onc.com.mu <- onc.com.mu[match(rownames(cn.mu.d),
                               rownames(onc.com.mu)), ]
onc.com.mu.d <- vegdist(onc.com.mu)
if (!(all(rownames(as.matrix(prb.mu.d)) ==
          rownames(as.matrix(cn.mu.d.onc))))) {
  warning("Warning: distance matrices are not aligned!")
}

# Bipartite analysis
nperm <- 20
if (!(file.exists("../data/lcn/nest_rel_onc.rda"))) {
  nest.onc <- nestedness(onc.com.rel[, colnames(onc.com.rel) != "ds"],
    n.nulls = 999
  )
  dput(nest.onc, "../data/lcn/nest_rel_onc.rda")
} else {
  nest.onc <- dget("../data/lcn/nest_rel_onc.rda")
}
if (!(file.exists("../data/lcn/null_mod_onc.csv"))) {
  obs.mod.onc <- bipartite::computeModules(
    onc.com.rel[, colnames(onc.com.rel) != "ds"]
  )
  mods.onc <- tail(
    apply(
      slot(obs.mod.onc, "modules"), 2,
      function(x) {
        sum(sign(x[2:length(x)]) *
          (1:(length(x) - 1)))
      }
    ),
    sum(dim(onc.com[, colnames(onc.com) != "ds"]))
  )
  mods.onc <- list(
    sp = tail(
      mods.onc,
      ncol(onc.com[, colnames(onc.com) != "ds"])
    ),
    tree = head(mods.onc, nrow(onc.com))
  )
  sim.onc <- lapply(1:nperm, sim.com, x = onc.q)
  sim.onc <- lapply(sim.onc, function(x) x / max(x))
  nul.mod.onc <- lapply(sim.onc, function(x) bipartite::computeModules(x))
  nul.mod.onc <- unlist(lapply(nul.mod.onc, slot, "likelihood"))
```

```
  dput(mods.onc, "../data/lcn/mod_list_onc.rda")
  write.csv(slot(obs.mod.onc, "likelihood"),
    "../data/lcn/obs_mod_onc.csv",
    row.names = FALSE
  )
  write.csv(nul.mod.onc,
    "../data/lcn/null_mod_onc.csv",
    row.names = FALSE
  )
} else {
  obs.mod.onc <- read.csv("../data/lcn/obs_mod_onc.csv")[1]
  nul.mod.onc <- read.csv("../data/lcn/null_mod_onc.csv")[, 1]
  z.mod.onc <- (obs.mod.onc - mean(nul.mod.onc)) / sd(nul.mod.onc)
  mods.onc <- dget("../data/lcn/mod_list_onc.rda")
}
pval.mod.onc <- length(nul.mod.onc[nul.mod.onc >= obs.mod.onc]) /
  length(nul.mod.onc)
if (pval.mod.onc == 0) {
  pval.mod.onc <- 1 / nperm
}
z.mod.onc <- (obs.mod.onc - mean(nul.mod.onc)) / sd(nul.mod.onc)
bp.mod.onc <- round(unlist(c(
  nperm = nperm, obs = obs.mod.onc,
  mu.sim = mean(nul.mod.onc), sd.sim = sd(nul.mod.onc),
  z = z.mod.onc, p.value = pval.mod.onc
)), 5)

## NMDS ordinations
# community ordination
if (!file.exists("../data/lcn/onc_nmds.csv")) {
  nms.info.onc <- capture.output(nms.onc <- nmds.min(nmds(
    vegdist(onc.com.rel), 2, 2
  )))
  write.csv(nms.onc, "../data/lcn/onc_nmds.csv",
    row.names = FALSE
  )
  write.table(nms.info.onc,
    "../data/lcn/onc_nmds_info.txt",
    col.names = FALSE, row.names = FALSE
  )
} else {
  nms.onc <- read.csv("../data/lcn/onc_nmds.csv")
}
# Network ordination
if (!(file.exists("../data/lcn/conet_nmds.csv"))) {
  cn.nmds.stats.onc <- capture.output(
    cn.nms.onc <- nmds.min(nmds(cn.d.onc, 2, 2))
  )
  write.csv(cn.nms.onc, "../data/lcn/conet_nmds.csv",
    row.names = FALSE
  )
  write.table(cn.nmds.stats.onc,
    "../data/lcn/conet_nmds_info.txt",
```

```
    col.names = FALSE, row.names = FALSE
  )
} else {
  cn.nms.onc <- read.csv("../data/lcn/conet_nmds.csv")
}
# Vector fitting
nv.onc <- envfit(cn.nms.onc, data.frame(onc.com[, colnames(onc.com) != "ds"],
  R = onc.rough,
  C = onc.ns[, c("C")],
  A = ptc.onc
))
cv.onc <- envfit(nms.onc, data.frame(onc.com[, colnames(onc.com) != "ds"],
  R = onc.rough,
  C = onc.ns[, c("C")],
  A = ptc.onc
))

# get araujo coordinates
coord <- read.csv("../data/lcn/lcn_coord_onc.csv")
rownames(coord) <- coord[, 1]
coord <- coord[, -1]
# packing into a dataframe
tree <- onc.geno
for (i in 1:length(unique(onc.geno))) {
  tree[onc.geno == unique(onc.geno)[i]] <-
    1:length(tree[onc.geno == unique(onc.geno)[i]])
}
tree <- factor(tree)
tree.id <- do.call(rbind, strsplit(names(ptc.onc), split = " "))[, 1]
# add chemistry data
onc.nc <- read.csv("../data/lcn/ONC_phytochem_NC.csv")
onc.tan <- read.csv("../data/lcn/ONC_phytochem_tannin.csv")
onc.nc[, 1] <- as.character(paste0("N", gsub("-", "\\.", onc.nc[, 1])))
onc.tan[, 1] <- as.character(paste0("N", gsub("-", "\\.", onc.tan[, 1])))
# rename headers
# mass is in mg
colnames(onc.nc)[1:4] <- c("tree.id", "sample.mass", "N", "C")
colnames(onc.tan)[1] <- "tree.id"
colnames(onc.tan)[grep("X.CT", colnames(onc.tan))] <- "CT"
# add C:N ratio
onc.nc$rCN <- onc.nc$N / onc.nc$C
# pH data
onc.ph <- read.csv("../data/lcn/ONC_Bark_lichen_pH_data.csv")
onc.ph[, "tree.id"] <- gsub("-", ".", onc.ph[, "tree.id"])
onc.ph[, "tree.id"] <- gsub("\\.0", "\\.", onc.ph[, "tree.id"])
# N7.16 is possibly N7.10
onc.ph[onc.ph[, "tree.id"] == "N7.16", "tree.id"] <- "N7.10"
# updated pH from Lamit
onc.ph[!is.na(onc.ph[, "pH2"]), "pH"] <-
  apply(onc.ph[!is.na(onc.ph[, "pH2"]), c("pH", "pH2")], 1, mean)
# collect into a single df
onc.dat <- data.frame(tree.id,
  PC = ptc.onc, SR = spr.onc,
```

```
  SD = spd.onc, SE = spe.onc,
  geno = factor(onc.geno), tree = tree,
  BR = onc.rough, onc.ns[, c("L", "Cen")]
)
# Get to match onc.dat
onc.ph <- onc.ph[onc.ph[, "tree.id"] %in% onc.dat[, "tree.id"], ]
onc.ph <- onc.ph[match(onc.dat[, "tree.id"], onc.ph[, "tree.id"]), ]
# append chemistry to onc.dat
onc.dat <- data.frame(onc.dat,
  C = onc.nc[match(
    onc.dat[, "tree.id"],
    onc.nc[, "tree.id"]
  ), "C"],
  N = onc.nc[match(
    onc.dat[, "tree.id"],
    onc.nc[, "tree.id"]
  ), "N"],
  CN = onc.nc[match(
    onc.dat[, "tree.id"],
    onc.nc[, "tree.id"]
  ), "rCN"],
  CT = onc.tan[match(
    onc.dat[, "tree.id"],
    onc.tan[, "tree.id"]
  ), "CT"],
  pH = onc.ph[, "pH"]
)
# Plot calculations
pw.onc <- onc.com.rel[, colnames(onc.com.rel) != "ds"]
pw.onc <- pw.onc[
  order(apply(pw.onc, 1, sum), decreasing = TRUE),
  order(apply(pw.onc, 2, sum), decreasing = TRUE)
]
rownames(pw.onc) <- onc.geno
col.pal <- RColorBrewer::brewer.pal((max(unlist(mods.onc))), "Paired")
# Network list NA removed
cn.d.onc.na <- distNet(cn.onc[names(cn.onc) %in% na.omit(onc.dat)$tree.id],
  method = "euclidean"
)
# Figure ordinations
# Communities
if (file.exists("../data/lcn/nms_com_onc.rda")) {
  nms.com <- dget(file = "../data/lcn/nms_com_onc.rda")
} else {
  set.seed(12345)
  nms.com <- nmds(vegdist(onc.com.rel), 2, 3)
  dput(nms.com, file = "../data/lcn/nms_com_onc.rda")
}
# Networks
if (file.exists("../data/lcn/nms_cn_onc.rda")) {
  nms.cn <- dget(file = "../data/lcn/nms_cn_onc.rda")
} else {
  set.seed(12345)
```

```
  nms.cn <- nmds(cn.d.onc.na, 1, 2)
  dput(nms.cn, file = "../data/lcn/nms_cn_onc.rda")
}
ord.com <- nmds.min(nms.com, 3)
```

```
## Minimum stress for given dimensionality:  0.1008923
## r^2 for minimum stress configuration:  0.9357192
```

```
ord.cn <- nmds.min(nms.cn, 2)
```

```
## Minimum stress for given dimensionality:  0.008516247
## r^2 for minimum stress configuration:  0.999471
```

```
# Vectors for plotting
# Composition
vec.com <- envfit(ord.com,
  env = onc.dat[, c("CT", "SR", "SE", "Cen"), ],
  perm = 10000,
  choices = c(1, 2), na.rm = TRUE
)
# Network similarity
vec.cn <- envfit(ord.cn,
  env = na.omit(onc.dat)[, c("CT", "SR", "SE", "Cen")],
  perm = 10000,
  choices = c(1, 2)
)
# onc
if (!("mod_obsval_onc.csv" %in% dir("../data/lcn"))) {
  mod.onc <- slot(
    bipartite::computeModules(rel(onc.com[, -ncol(onc.com)]),
      deep = FALSE
    ),
    "likelihood"
  )
  write.csv(mod.onc, file = "../data/lcn/mod_obsval_onc.csv", row.names = FALSE)
} else {
  mod.onc <- read.csv(file = "../data/lcn/mod_obsval_onc.csv")[, 1]
}
if (!("mod_simvals_onc.csv" %in% dir("../data/lcn"))) {
  onc.sweb <- simulate(vegan::nullmodel(onc.com[, -ncol(onc.com)],
    method = "swsh_samp_c"
  ), 99)
  for (i in 1:dim(onc.sweb)[3]) {
    onc.sweb[, , i] <- rel(onc.sweb[, , i])
  }
  onc.smod <- apply(onc.sweb, 3, bipartite::computeModules)
  mods.onc.sweb <- unlist(lapply(onc.smod, slot, name = "likelihood"))
  write.csv(mods.onc.sweb,
    file = "../data/lcn/mod_simvals_onc.csv",
    row.names = FALSE
  )
  # nest.onc <- bipartite::nestedness(onc.com.rel)
} else {
  mods.onc.sweb <- read.csv(
    file = "../data/lcn/mod_simvals_onc.csv"
```

```
  )[, 1]
}

# pit
if (!("mod_obsval_pit.csv" %in% dir("../data/lcn"))) {
  mod.pit <- slot(
    bipartite::computeModules(rel(pit.com), deep = FALSE),
    "likelihood"
  )
  write.csv(mod.pit,
    file = "../data/lcn/mod_obsval_pit.csv",
    row.names = FALSE
  )
} else {
  mod.pit <- read.csv(
    file = "../data/lcn/mod_obsval_pit.csv"
  )[, 1]
}
if (!("mod_simvals_pit.csv" %in% dir("../data/lcn"))) {
  pit.sweb <- simulate(vegan::nullmodel(pit.com, method = "swsh_samp_c"), 99)
  for (i in 1:dim(pit.sweb)[3]) {
    pit.sweb[, , i] <- rel(pit.sweb[, , i])
  }
  pit.smod <- apply(pit.sweb, 3, bipartite::computeModules)
  mods.pit.sweb <- unlist(lapply(pit.smod, slot, name = "likelihood"))
  write.csv(mods.pit.sweb,
    file = "../data/lcn/mod_simvals_pit.csv",
    row.names = FALSE
  )
  # nest.pit <- bipartite::nestedness(pit.com.rel)
} else {
  mods.pit.sweb <- read.csv(
    file = "../data/lcn/mod_simvals_pit.csv"
  )[, 1]
}

### Wild data
###
###
x <- read.csv("../data/lcn/lco_Apr2012.csv")
# remove notes
x <- x[, colnames(x) != "NOTES."]
x <- x[, colnames(x) != "dead"]
#
x <- na.omit(x)
# remove gnu.44 = FREMONT
x <- x[x$tree != "gnu.44", ]
# rm ll.6, tree with super smooth bark
x <- x[x$tree != "ll.6", ]
x$tree <- factor(as.character(x$tree))
# condense species
# lecanora, there can be only one!
lec.sp <- apply(x[, c(6, 8, 10, 18)], 1, function(x) sign(any(x != 0)))
```

```r
# no physcioids!
# phy.spp <- apply(x[,c(13,14,15,16)],
# 1,function(x) sign(any(x!=0)))
x <- cbind(x, lec = lec.sp)
x <- x[, -c(6, 8, 10, 18)]
x <- x[, colnames(x) != "physcioid"]
# break into quadrat list (x.q)
quads <- paste(x$tree, x$quadrat)
colnames(x)[5:ncol(x)] <- c(
  "Xg", "Cs", "Xm", "fgb", "Rs",
  "Pm", "Pa", "Pu", "Ch", "Ls"
)
x <- x[colnames(x) != "fgb"]
x.q <- split(x, quads)
wild.com <- split(x, x$tree)
wild.com <- do.call(rbind, lapply(wild.com, function(x) apply(x[, -1:-4], 2, sum)))
wild.com.rel <- apply(wild.com, 2, function(x) x / max(x))
wild.com.rel[is.na(wild.com.rel)] <- 0
wild.q <- lapply(split(x, x$tree), function(x) x[, -1:-4])
# data from lamit
env <- read.csv("../data/lcn/Uinta2012_all_data_from_Lamit.csv")
env <- env[is.na(env$Pct.Roughness) == FALSE, ]
env[, 1] <- sub(
  "\\?", "",
  sub(
    "\\.0", "\\.",
    sub(
      "\\_", "\\.",
      sub("\\-", "\\.", tolower(as.character(env[, 1])))
    )
  )
)
env[env[, 1] == "ll.6_(znu.29)", 1] <- "ll.6"
env[env[, 1] == "gnu.85.1ftaway", 1] <- "gnu.85"
env$Quad.Loc <- as.character(sapply(
  as.character(env$Quad.Loc),
  function(x) {
    unlist(strsplit(x, split = "_"))[2]
  }
))
env$Quad.Loc <- sub("\\-", "\\.", env$Quad.Loc)
env$Quad.Loc <- paste("n", env$Quad.Loc, sep = "")
# remove southern aspect
env <- env[env$Aspect != "South", ]
env.tid <- paste(env$Tree.ID, env$Quad.Loc)
# check that the datasets are compatible
all(names(x.q) %in% env.tid)
```

```
## [1] TRUE
```

```r
# match observations
all(env.tid[match(names(x.q), env.tid)] == names(x.q))
```

```
## [1] TRUE
```

```r
# delimit co-occurrence and match
env <- env[match(names(x.q), env.tid), ]
x.split <- paste(x$tree, x$quadrat, sep = "_")
env.split <- paste(env$Tree.ID, env$Quad.Loc)
x.split <- as.character(x$tree)
env.split <- as.character(env$Tree.ID)
# percent rough bark
prb.wild <- tapply(env$Pct.Roughness, env.split, mean)
# age
age <- read.csv("../data/lcn/UintaMaster_LichenHeritNL_FallSpring_2012_ForLau.csv")
dbh <- age$DBH.cm_01
age.final <- age$AgeFinal.U
age <- data.frame(tree.id = age[, 1], age.final = age$AgeFinal.U)
age[, 1] <- tolower(age[, 1])
age[, 1] <- sub("_", "\\.", age[, 1])
age[, 1] <- sub("-", "\\.", age[, 1])
age[, 1] <- sub("\\?", "", age[, 1])
age[, 1] <- sub("\\.0", "\\.", age[, 1])
age[age[, 1] == "gnu.85.1ftaway", 1] <- "gnu.85"
# predict age
gnu19.dbh <- dbh[age$tree.id == "gnu.19"]
new <- data.frame(dbh = seq(min(dbh), max(dbh), by = 0.1))
age.final <- na.omit(age.final)
pred.age <- predict(lm(age.final ~ dbh, data = age), new)
gnu19.age <- as.numeric(pred.age[new[, 1] == gnu19.dbh])
#
tree.age <- age[match(names(prb.wild), age[, 1]), 2]
tree.age[is.na(tree.age)] <- gnu19.age
names(tree.age) <- age[match(names(prb.wild), age[, 1]), 1]
age <- tree.age
# percent cover
pc.wild <- unlist(lapply(
  wild.q,
  function(x) {
    sum(apply(
      x, 1,
      function(x) sign(sum(x))
    ))
  }
))
# richness
sr.wild <- unlist(lapply(
  wild.q, function(x) sum(sign(apply(x, 2, sum))))
  )
# networks
cn.wild <- lapply(wild.q, coNet)
cn.mu.wild <- meanNet(cn.wild)
cn.d.wild <- distNet(cn.wild, method = "bc")
# network stats
ns.wild <- do.call(rbind, lapply(lapply(cn.wild, function(x) {
  abs(sign(x))
}), enaR:::structure.statistics))
# centralization
```

```r
dcen.wild <- unlist(lapply(cn.wild, function(x) {
  sna::centralization(x, FUN = sna::degree, normalize = FALSE)
}))
# wild data frame
wild.dat <- data.frame(
  tree = names(tree.age),
  age = tree.age, BR = prb.wild,
  PC = pc.wild, SR = sr.wild,
  L = ns.wild[, "L"], Cen = dcen.wild
)
```

Results

```r
### REML
## Create a list to generate a results table
h2.tab <- matrix("", 1, 4)
colnames(h2.tab) <- c("Response", "H2", "R2", "p-value")

## Total cover ~ genotype
ptc.reml <- lme4::lmer(I(PC^(1 / 2)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
ptc.reml.pval <- RLRsim::exactRLRT(ptc.reml)
fligner.test(onc.dat$PC^(1 / 2), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$PC^(1/2) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 13.751, df = 12, p-value =
## 0.3169
```

```r
shapiro.test(residuals(ptc.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ptc.reml)
## W = 0.93193, p-value = 0.004822
```

```r
ptc.reml.result <- c(
  "Percent Lichen Cover",
  H2(ptc.reml, g = onc.dat$geno),
  R2(ptc.reml),
  ptc.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, ptc.reml.result)

## Species richness ~ genotype
spr.reml <- lme4::lmer(I(SR^(1 / 2)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
spr.reml.pval <- RLRsim::exactRLRT(spr.reml)
```

```
spr.reml.pval
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.080424, p-value = 0.3517
```

```
shapiro.test(residuals(spr.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(spr.reml)
## W = 0.95213, p-value = 0.03324
```

```
fligner.test(onc.dat$SR^(1 / 2), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$SR^(1/2) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 13.276, df = 12, p-value =
## 0.3493
```

```
spr.reml.result <- c(
  "Lichen Species Richness",
  H2(spr.reml, g = onc.dat$geno),
  R2(spr.reml),
  spr.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, spr.reml.result)

## Bark roughness REML
prb.reml <- lme4::lmer(I(BR^(1 / 2)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
prb.reml.pval <- RLRsim::exactRLRT(prb.reml)
fligner.test(onc.dat$BR^(1 / 2), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$BR^(1/2) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 6.1915, df = 12, p-value =
## 0.9061
```

```
shapiro.test(residuals(prb.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(prb.reml)
## W = 0.98395, p-value = 0.6926
```

```
prb.reml.result <- c(
  "Percent Rough Bark",
  H2(prb.reml, g = onc.dat$geno),
  R2(prb.reml),
  prb.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, prb.reml.result)

## pH ~ genotype
ph.reml <- lme4::lmer(I(pH^(1 / 2)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
ph.reml.pval <- RLRsim::exactRLRT(ph.reml)
fligner.test(log(onc.dat$pH), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  log(onc.dat$pH) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 22.971, df = 12, p-value =
## 0.02797
```

```
shapiro.test(residuals(ph.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ph.reml)
## W = 0.76737, p-value = 9.03e-08
```

```
ph.reml.result <- c(
  "pH",
  H2(ph.reml, g = onc.dat$geno),
  R2(ph.reml),
  ph.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, ph.reml.result)

## condensed tannins  REML
ct.reml <- lme4::lmer(I(CT^(1 / 4)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
ct.reml.pval <- RLRsim::exactRLRT(ct.reml)
fligner.test(onc.dat$CT^(1 / 4), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$CT^(1/4) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 7.8941, df = 12, p-value =
## 0.7933
```

```
shapiro.test(residuals(ct.reml))
```

```
##
##  Shapiro-Wilk normality test
```

```
##
## data:  residuals(ct.reml)
## W = 0.75061, p-value = 4.002e-08
```

```
ct.reml.result <- c(
  "Condensed Tannins (CT)",
  H2(ct.reml, g = onc.dat$geno),
  R2(ct.reml),
  ct.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, ct.reml.result)

## CN ratio REML
cnr.reml <- lme4::lmer(I(CN^(1)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
```

```
## boundary (singular) fit: see ?isSingular
```

```
cnr.reml.pval <- RLRsim::exactRLRT(cnr.reml)
fligner.test(onc.dat$CN^(1 / 1), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$CN^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 8.1116, df = 12, p-value =
## 0.7763
```

```
shapiro.test(residuals(cnr.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(cnr.reml)
## W = 0.92434, p-value = 0.002442
```

```
cnr.reml.result <- c(
  "Carbon-Nitrogen (CN) Ratio",
  H2(cnr.reml, g = onc.dat$geno),
  R2(cnr.reml),
  cnr.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, cnr.reml.result)

## Correlations among metrics
cor.test(onc.dat[, "SR"], onc.dat[, "PC"], data = onc.dat)
```

```
##
##  Pearson's product-moment correlation
##
## data:  onc.dat[, "SR"] and onc.dat[, "PC"]
## t = 8.3456, df = 55, p-value = 2.393e-11
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6047186 0.8437321
## sample estimates:
```

```
##       cor
## 0.7475023
```

```
## Were these correlated with bark roughness?
ptc.prb.lm <- lm(I(PC^(1 / 2)) ~ I(BR^(1 / 2)), data = onc.dat)
summary(ptc.prb.lm)
```

```
##
## Call:
## lm(formula = I(PC^(1/2)) ~ I(BR^(1/2)), data = onc.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9770 -1.6378  0.6333  1.9603  3.4658
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.4142     1.0901   4.049 0.000162 ***
## I(BR^(1/2))   0.4942     0.1896   2.607 0.011730 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.485 on 55 degrees of freedom
## Multiple R-squared:   0.11,  Adjusted R-squared:  0.09381
## F-statistic: 6.797 on 1 and 55 DF,  p-value: 0.01173
```

```
fligner.test(onc.dat$PC, onc.dat$BR)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$PC and onc.dat$BR
## Fligner-Killeen:med chi-squared = 27.401, df = 24, p-value =
## 0.2861
```

```
shapiro.test(residuals(ptc.prb.lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ptc.prb.lm)
## W = 0.95045, p-value = 0.02061
```

```
spr.prb.lm <- lm(I(SR^(1)) ~ I(BR^(1 / 2)), data = onc.dat)
summary(spr.prb.lm)
```

```
##
## Call:
## lm(formula = I(SR^(1)) ~ I(BR^(1/2)), data = onc.dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0420 -1.3123 -0.1178  1.2308  4.3519
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.5015     0.8002   3.126  0.00283 **
```

```
## I(BR^(1/2))   0.1709     0.1392   1.228  0.22456
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.824 on 55 degrees of freedom
## Multiple R-squared:  0.0267, Adjusted R-squared:  0.009003
## F-statistic: 1.509 on 1 and 55 DF,  p-value: 0.2246
```

```
fligner.test(onc.dat$SR^(1), onc.dat$BR)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$SR^(1) and onc.dat$BR
## Fligner-Killeen:med chi-squared = 26.046, df = 24, p-value =
## 0.3508
```

```
shapiro.test(residuals(spr.prb.lm))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(spr.prb.lm)
## W = 0.97168, p-value = 0.2008
```

```
## COM ~ genotype + Bark roughness + PTC + SPR
set.seed(2)
rcom.ng.perm <- vegan::adonis2(onc.com.rel^(1 / 1) ~ BR + PC + SR,
  data = onc.dat, perm = 10000, mrank = TRUE
)
set.seed(2)
rcom.perm <- vegan::adonis2(onc.com.rel^(1 / 1) ~ geno + BR + PC + SR,
  data = onc.dat, perm = 10000, mrank = TRUE
)
set.seed(2)
com.ng.perm <- vegan::adonis2(onc.com^(1 / 1) ~ BR + PC + SR,
  data = onc.dat, perm = 10000, mrank = TRUE
)
set.seed(2)
com.perm <- vegan::adonis2(onc.com^(1 / 1) ~ geno + BR + PC + SR,
  data = onc.dat, perm = 10000, mrank = TRUE
)
h2.tab[4, "p-value"] <- unlist(rcom.perm)["Pr(>F)1"]
h2.tab[4, "H2"] <- H2(rcom.perm, g = onc.dat$geno)
h2.tab[4, "R2"] <- R2(rcom.perm)
h2.tab[4, "Response"] <- "Lichen Community Composition"

## Is network similarity correlated with community composition?
ecodist::mantel(cn.d.onc ~ vegdist(onc.com.rel), mrank = TRUE)
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## 0.09198784 0.07200000 0.92900000 0.12000000 0.05120132 0.13656424
```

```
spr.d <- dist(onc.dat$SR)
ptc.d <- dist(onc.dat$PC)
prb.d <- dist(onc.dat$BR)
### rough -> cover -> rich -> net
```

```
ecodist::mantel(cn.d.onc ~ vegdist(onc.com.rel) + spr.d + ptc.d + prb.d, mrank = TRUE)
```

```
##    mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## 0.06853395 0.15400000 0.84700000 0.31300000 0.02256902 0.13046001
```

```
## Mantels
ecodist::mantel(cn.mu.d.onc ~ onc.com.mu.d)
```

```
##     mantelr       pval1       pval2       pval3   llim.2.5%  ulim.97.5%
##  0.29000439  0.07700000  0.92400000  0.07700000 -0.03054894  0.45045206
```

```
## Was lichen network similarity determined by genotype?
set.seed(1234)
cn.perm <- vegan::adonis2(cn.d.onc.na ~ geno +
  BR + pH + CN + CT +
  PC + SR + SE,
by = "term",
data = na.omit(onc.dat),
permutations = 10000, mrank = FALSE
)

cn.perm.ng <- vegan::adonis2(cn.d.onc ~ BR + PC + SR,
  data = onc.dat, permutations = 10000, mrank = TRUE
)
cn.perm.ng
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = cn.d.onc ~ BR + PC + SR, data = onc.dat, permutations = 10000, mrank = TRUE)
##          Df SumOfSqs      R2       F    Pr(>F)
## BR        1    61.42 0.03968  4.1680   0.04020 *
## PC        1    49.47 0.03197  3.3573   0.06839 .
## SR        1   655.76 0.42373 44.5034 9.999e-05 ***
## Residual 53   780.96 0.50462
## Total    56  1547.61 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cn.perm
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = cn.d.onc.na ~ geno + BR + pH + CN + CT + PC + SR + SE, data = na.omit(onc.da
##          Df SumOfSqs      R2       F    Pr(>F)
## geno     12   367.65 0.26937  2.3065   0.02860 *
## BR        1    23.63 0.01732  1.7792   0.18828
## pH        1     8.96 0.00656  0.6745   0.41866
## CN        1    37.70 0.02762  2.8379   0.08849 .
## CT        1    76.22 0.05585  5.7383   0.03310 *
## PC        1    28.50 0.02088  2.1458   0.14349
```

```
## SR        1   332.23 0.24342 25.0117 9.999e-05 ***
## SE        1    51.59 0.03780  3.8843   0.04470 *
## Residual 33   438.33 0.32117
## Total    52  1364.81 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
h2.tab[5, "p-value"] <- as.matrix(cn.perm)[1, "Pr(>F)"]
h2.tab[5, "H2"] <- H2(cn.perm, g = onc.dat[, "geno"], perm = 10000)
h2.tab[5, "R2"] <- R2(cn.perm)
h2.tab[5, "Response"] <- "Lichen Network"
# db rda for network similarity
dbr.cn.geno <- vegan::dbrda(cn.d.onc ~ geno, data = onc.dat, distance = "bray")
anova(dbr.cn.geno, permutations = 5000)
```

```
## Permutation test for dbrda under reduced model
## Permutation: free
## Number of permutations: 5000
##
## Model: vegan::dbrda(formula = cn.d.onc ~ geno, data = onc.dat, distance = "bray")
##          Df Variance      F Pr(>F)
## Model    12    8.045 1.5057  0.146
## Residual 44   19.591
```

```
H2(dbr.cn.geno)
```

```
## [1] 0.2911089
```

```
## What aspects of networks explained the similiarity?
## L = number of edges, LD = link density, C = connectivity,
## dcen = degree centrality
link.reml <- lme4::lmer(I(log(L + 0.00000001)) ~ (1 | geno),
  data = onc.dat, REML = TRUE
)
link.reml.pval <- RLRsim::exactRLRT(link.reml, nsim = 50000)
fligner.test(log(onc.dat$L + 0.0000001), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  log(onc.dat$L + 1e-07) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 11.991, df = 12, p-value =
## 0.4464
```

```
shapiro.test(residuals(link.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(link.reml)
## W = 0.83643, p-value = 2.036e-06
```

```
link.reml.result <- c(
 "Number of Network Links",
 H2(link.reml, g = onc.dat$geno),
 R2(link.reml),
 link.reml.pval$"p.value"
```

```
)
h2.tab <- rbind(h2.tab, link.reml.result)

# network centrality
cen.reml <- lme4::lmer(I(Cen^(1 / 2)) ~ (1 | geno),
  data = onc.dat, REML = TRUE
)
cen.reml.pval <- RLRsim::exactRLRT(cen.reml, nsim = 50000)
fligner.test(onc.dat$L^(1 / 1), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$L^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 14.241, df = 12, p-value =
## 0.2856
```

```
shapiro.test(residuals(cen.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(cen.reml)
## W = 0.90072, p-value = 0.0002041
```

```
cen.reml.result <- c(
 "Network Centrality",
 H2(cen.reml, g = onc.dat$geno),
 R2(cen.reml),
 cen.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, cen.reml.result)

# network modularity
mod.reml <- lme4::lmer(I(onc.ns[, "mod.lik"]^(1 / 4)) ~ (1 | geno),
  data = onc.dat, REML = TRUE
)
mod.reml.pval <- RLRsim::exactRLRT(mod.reml)
fligner.test(onc.ns[, "mod.lik"]^(1 / 4), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.ns[, "mod.lik"]^(1/4) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 13.439, df = 12, p-value =
## 0.3379
```

```
shapiro.test(residuals(mod.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(mod.reml)
## W = 0.54001, p-value = 4.252e-12
```

```
mod.reml.result <- c(
 "Network Modularity",
 H2(mod.reml, g = onc.dat$geno),
 R2(mod.reml),
 mod.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, mod.reml.result)

## Added diversity and evenness

## Species diversity ~ genotype
spd.reml <- lme4::lmer(I(SD^(1 / 2)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
spd.reml.pval <- RLRsim::exactRLRT(spd.reml)
shapiro.test(residuals(spd.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(spd.reml)
## W = 0.89237, p-value = 0.0001793
```

```
fligner.test(onc.dat$SD^(1 / 2), onc.dat$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$SD^(1/2) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 17.299, df = 12, p-value =
## 0.1387
```

```
spd.reml.result <- c(
 "Lichen Species Diversity",
 H2(spd.reml, g = onc.dat$geno),
 R2(spd.reml),
 spd.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, spd.reml.result)

## Species diversity ~ genotype
spe.reml <- lme4::lmer(I(SE^(1 / 4)) ~ (1 | geno),
  data = na.omit(onc.dat), REML = TRUE
)
spe.reml.pval <- RLRsim::exactRLRT(spe.reml)
shapiro.test(residuals(spe.reml))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(spe.reml)
## W = 0.67851, p-value = 1.705e-09
```

```
fligner.test(onc.dat$SD^(1 / 2), onc.dat$geno)
```

```
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  onc.dat$SD^(1/2) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 17.299, df = 12, p-value =
## 0.1387
```

```
spe.reml.result <- c(
 "Lichen Species Evenness",
 H2(spe.reml, g = onc.dat$geno),
 R2(spe.reml),
 spe.reml.pval$"p.value"
)
h2.tab <- rbind(h2.tab, spe.reml.result)

# network stats in relation to other variables
L.aov <- aov(I(log(L + 0.000001)) ~ BR + PC + SR, data = onc.dat)
summary(L.aov)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## BR           1  102.3   102.3   2.776   0.1016
## PC           1  239.6   239.6   6.504   0.0137 *
## SR           1  957.0   957.0  25.980 4.71e-06 ***
## Residuals   53 1952.2    36.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
shapiro.test(residuals(L.aov))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(L.aov)
## W = 0.9629, p-value = 0.07794
```

```
cen.aov <- aov(I(Cen^(1 / 2)) ~ BR + PC + SR, data = onc.dat)
summary(cen.aov)
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## BR           1   3.77    3.77   2.174    0.146
## PC           1   6.46    6.46   3.724    0.059 .
## SR           1  56.48   56.48  32.552 5.31e-07 ***
## Residuals   53  91.95    1.73
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
shapiro.test(residuals(cen.aov))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(cen.aov)
## W = 0.97222, p-value = 0.2126
```

```
# are these metrics correlated with network similarity
L.d <- dist(onc.dat$L)
cen.d <- dist(onc.dat$Cen)
mod.d <- dist(cn.mod.onc)
```

```
cn.L.cen.perm <- adonis2(cn.d.onc ~ L + Cen, data = onc.dat, mrank = TRUE)

## So, are there patterns in the centrality of individual lichen species?
sppcen.test <- apply(cen.spp[, apply(cen.spp, 2, sum) >= 2], 2, function(x) {
  lme4::lmer(I(x^(1 / 2)) ~ (1 | geno), data = na.omit(onc.dat), REML = TRUE)
})
```

```
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
## boundary (singular) fit: see ?isSingular
```

```
sppcen.pval <- lapply(sppcen.test, RLRsim::exactRLRT)
sppcen.tab <- do.call(rbind, lapply(sppcen.pval, function(x) {
  c(x[["statistic"]], x[["p.value"]])
}))
sppcen.h2 <- round(unlist(lapply(sppcen.test, H2)), 3)
sppcen.h2
```

```
##    Xg    Cs    Ls    Ch    Xm    Pm    Rs
## 0.000 0.076 0.000 0.152 0.190 0.000 0.000
```

```
## Mean centrality of species
sort(apply(cen.spp, 2, mean), decreasing = TRUE)
```

```
##         Cs         Ch         Ls         Rs         Xg         Pm
## 0.73023360 0.51060368 0.41242791 0.19765745 0.15651469 0.08511420
##         Xm         Pu         Pa
## 0.06858041 0.02264151 0.00000000
```

```
summary(aov(value ~ X2, data = melt(cen.spp)))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## X2            8  27.04   3.380   8.086  3e-10 ***
## Residuals   468 195.63   0.418
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
TukeyHSD(aov(value ~ X2, data = melt(cen.spp)))
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ X2, data = melt(cen.spp))
##
## $X2
##              diff        lwr          upr     p adj
## Cs-Ch  0.21962992 -0.1717750  0.611034785 0.7158140
## Ls-Ch -0.09817577 -0.4895806  0.293229097 0.9973174
## Pa-Ch -0.51060368 -0.9020086 -0.119198814 0.0018322
## Pm-Ch -0.42548948 -0.8168944 -0.034084614 0.0215980
## Pu-Ch -0.48796217 -0.8793670 -0.096557305 0.0037079
## Rs-Ch -0.31294623 -0.7043511  0.078458638 0.2386358
## Xg-Ch -0.35408900 -0.7454939  0.037315874 0.1124897
## Xm-Ch -0.44202327 -0.8334281 -0.050618401 0.0139219
## Ls-Cs -0.31780569 -0.7092106  0.073599181 0.2201386
## Pa-Cs -0.73023360 -1.1216385 -0.338828730 0.0000004
```

```
## Pm-Cs -0.64511940 -1.0365243 -0.253714530 0.0000145
## Pu-Cs -0.70759209 -1.0989970 -0.316187221 0.0000011
## Rs-Cs -0.53257615 -0.9239810 -0.141171278 0.0008956
## Xg-Cs -0.57371891 -0.9651238 -0.182314042 0.0002163
## Xm-Cs -0.66165319 -1.0530581 -0.270248317 0.0000075
## Pa-Ls -0.41242791 -0.8038328 -0.021023042 0.0301134
## Pm-Ls -0.32731371 -0.7187186  0.064091158 0.1867880
## Pu-Ls -0.38978640 -0.7811913  0.001618467 0.0519144
## Rs-Ls -0.21477046 -0.6061753  0.176634410 0.7402505
## Xg-Ls -0.25591322 -0.6473181  0.135491646 0.5178196
## Xm-Ls -0.34384750 -0.7352524  0.047557371 0.1376135
## Pm-Pa  0.08511420 -0.3062907  0.476519070 0.9990336
## Pu-Pa  0.02264151 -0.3687634  0.414046379 1.0000000
## Rs-Pa  0.19765745 -0.1937474  0.589062322 0.8187315
## Xg-Pa  0.15651469 -0.2348902  0.547919558 0.9456749
## Xm-Pa  0.06858041 -0.3228245  0.459985283 0.9998045
## Pu-Pm -0.06247269 -0.4538776  0.328932178 0.9999034
## Rs-Pm  0.11254325 -0.2788616  0.503948121 0.9931235
## Xg-Pm  0.07140049 -0.3200044  0.462805357 0.9997355
## Xm-Pm -0.01653379 -0.4079387  0.374871082 1.0000000
## Rs-Pu  0.17501594 -0.2163889  0.566420812 0.9000620
## Xg-Pu  0.13387318 -0.2575317  0.525278048 0.9786988
## Xm-Pu  0.04593890 -0.3454660  0.437343773 0.9999909
## Xg-Rs -0.04114276 -0.4325476  0.350262105 0.9999962
## Xm-Rs -0.12907704 -0.5204819  0.262327830 0.9830852
## Xm-Xg -0.08793428 -0.4793391  0.303470594 0.9987764
```

```
apply(cen.spp, 2, sd) / sqrt(nrow(cen.spp))
```

```
##         Xg         Cs         Ls         Ch         Xm         Pm
## 0.05925513 0.14349031 0.13434259 0.12920446 0.04002923 0.04918653
##         Pa         Pu         Rs
## 0.00000000 0.02264151 0.08722534
```

```
## Lichen size distribution
## X. gallericulata thalli are about 0.22 +/- 0.003 cm^2 on average
## with an average median size of 0.12 +/- 0.001 cm^2
## and, size does not vary significantly with genotype.
xgs.reml <- lme4::lmer(I(mean.thallus) ~ (1 | geno),
  data = xgs.data[xgs.data$geno %in%
    names(which(table(xgs.data$geno) > 2)), ],
  REML = TRUE
)
xgs.median.reml <- lme4::lmer(median.thallus ~ (1 | geno),
  data = xgs.data[xgs.data$geno %in%
    names(which(table(xgs.data$geno) > 2)), ],
  REML = TRUE
)
RLRsim::exactRLRT(xgs.reml)
```

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

```
## data:
## RLRT = 2.4792, p-value = 0.0473
```

```
RLRsim::exactRLRT(xgs.median.reml)
```

```
##
##  simulated finite sample distribution of RLRT.
##
##  (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.092023, p-value = 0.3389
```

```
fligner.test(xgs.data$mean.thallus, xgs.data$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  xgs.data$mean.thallus and xgs.data$geno
## Fligner-Killeen:med chi-squared = 13.244, df = 17, p-value =
## 0.7197
```

```
fligner.test(xgs.data$median.thallus, xgs.data$geno)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  xgs.data$median.thallus and xgs.data$geno
## Fligner-Killeen:med chi-squared = 19.374, df = 17, p-value =
## 0.3075
```

```
mean(xgs.data$mean.thallus)
```

```
## [1] 0.1808442
```

```
sd(xgs.data$mean.thallus) / (length(xgs.data$mean.thallus) - 1)
```

```
## [1] 0.001845945
```

```
mean(xgs.data$median.thallus)
```

```
## [1] 0.1170852
```

```
sd(xgs.data$median.thallus) / (length(xgs.data$median.thallus) - 1)
```

```
## [1] 0.001223999
```

```
# ONC and Wild Stand (Uintah)
all.dat <- rbind(
  wild.dat[, c("BR", "PC", "SR", "L", "Cen")],
  onc.dat[, c("BR", "PC", "SR", "L", "Cen")]
)
# Network distances
cn.all <- cn.wild
for (i in 1:length(cn.wild)) {
  cn.all[[i]] <- cn.wild[[i]][
    match(rownames(cn.onc[[1]]), rownames(cn.wild[[i]])),
    match(colnames(cn.onc[[1]]), colnames(cn.wild[[i]]))
  ]
```

```
}
cn.all <- append(cn.all, cn.onc)
cn.d.all <- distNet(cn.all, method = "bc")
cn.nms.geno <- c(rep("wild", length(cn.wild)), onc.geno)
if (!exists("cn.nms.all")) {
  set.seed(12345)
  cn.nms.all <- nmds.min(nmds(cn.d.all, 2, 2))
  vec.all <- envfit(cn.nms.all, all.dat)
  # jitter identical points
  cn.nms.all[cn.nms.geno == "H10", ] <- cn.nms.all[
    cn.nms.geno == "H10", ] - 0.2
}
```

```
## Minimum stress for given dimensionality:  0.04194367
## r^2 for minimum stress configuration:  0.9915263
```

Tables

```
h2.tab[, "H2"] <- round(as.numeric(h2.tab[, "H2"]), digits = 5)
h2.tab[, "R2"] <- round(as.numeric(h2.tab[, "R2"]), digits = 5)
h2.tab[, "p-value"] <- round(as.numeric(h2.tab[, "p-value"]),
                             digits = 5)
h2.tab <- h2.tab[order(h2.tab[, "H2"], decreasing = TRUE), ]
h2.xtab <- xtable::xtable(h2.tab,
  caption =
    "Genotypic effects of cottonwood trees on the associated lichen community.",
  label = "tab:h2_table"
)
print(h2.xtab,
  type = "latex",
  include.rownames = FALSE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

Response	H2	R2	p-value
Condensed Tannins (CT)	0.25556	0.25556	0.0193
Network Centrality	0.20166	0.20166	0.03958
Number of Network Links	0.17016	0.17016	0.06674
Lichen Community Composition	0.16093	0.24287	0.0032
Percent Lichen Cover	0.1368	0.1368	0.0841
Lichen Network	0.06385	0.26937	0.0286
Lichen Species Evenness	0.05732	0.05732	0.2461
Network Modularity	0.05731	0.05731	0.267
Lichen Species Diversity	0.02908	0.02908	0.3532
Lichen Species Richness	0.02807	0.02807	0.3517
Carbon-Nitrogen (CN) Ratio	0	0	0.4608

Table 1: Genotypic effects of cottonwood trees on the associated lichen community.

```
# community permanova
rcom.ng.perm.xtab <- xtable::xtable(rcom.ng.perm,
  caption =
    "PerMANOVA Pseudo-F Table showing the predictors of community similarity.",
  label = "tab:com_ng_perm"
)
print(rcom.ng.perm.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	SumOfSqs	R2	F	Pr(>F)
BR	1	0.44	0.04	3.74	0.0088
PC	1	3.86	0.34	32.85	0.0001
SR	1	0.78	0.07	6.60	0.0001
Residual	53	6.23	0.55		
Total	56	11.31	1.00		

Table 2: PerMANOVA Pseudo-F Table showing the predictors of community similarity.

```
rcom.perm.xtab <- xtable::xtable(rcom.perm,
  caption =
    "PerMANOVA Pseudo-F Table showing the predictors of community similarity.",
  label = "tab:rcom_perm"
)
print(rcom.perm.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	SumOfSqs	R2	F	Pr(>F)
geno	12	2.75	0.24	1.82	0.0032
BR	1	0.12	0.01	0.99	0.3901
PC	1	2.67	0.24	21.27	0.0001
SR	1	0.62	0.05	4.90	0.0010
Residual	41	5.15	0.46		
Total	56	11.31	1.00		

Table 3: PerMANOVA Pseudo-F Table showing the predictors of community similarity.

```
# network permanova
cn.perm.ng.xtab <- xtable::xtable(cn.perm.ng,
  caption =
    "PerMANOVA Pseudo-F Table showing the predictors of network similarity.",
  label = "tab:cn_perm_ng"
)
print(cn.perm.ng.xtab,
  type = "latex",
  include.rownames = TRUE,
```

```
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	SumOfSqs	R2	F	Pr(>F)
BR	1	61.42	0.04	4.17	0.0402
PC	1	49.47	0.03	3.36	0.0684
SR	1	655.76	0.42	44.50	0.0001
Residual	53	780.96	0.50		
Total	56	1547.61	1.00		

Table 4: PerMANOVA Pseudo-F Table showing the predictors of network similarity.

```
cn.perm.xtab <- xtable::xtable(cn.perm,
  caption =
    "PerMANOVA Pseudo-F Table showing the predictors of network similarity.",
  label = "tab:cn_perm"
)
print(cn.perm.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	SumOfSqs	R2	F	Pr(>F)
geno	12	367.65	0.27	2.31	0.0286
BR	1	23.63	0.02	1.78	0.1883
pH	1	8.96	0.01	0.67	0.4187
CN	1	37.70	0.03	2.84	0.0885
CT	1	76.22	0.06	5.74	0.0331
PC	1	28.50	0.02	2.15	0.1435
SR	1	332.23	0.24	25.01	0.0001
SE	1	51.59	0.04	3.88	0.0447
Residual	33	438.33	0.32		
Total	52	1364.81	1.00		

Table 5: PerMANOVA Pseudo-F Table showing the predictors of network similarity.

```
# network metrics anova
L.aov.xtab <- xtable::xtable(L.aov,
  caption =
    "ANOVA F Table showing the predictors of the number of network links.",
  label = "tab:L_aov"
)
print(L.aov.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
BR	1	102.25	102.25	2.78	0.1016
PC	1	239.57	239.57	6.50	0.0137
SR	1	956.96	956.96	25.98	0.0000
Residuals	53	1952.23	36.83		

Table 6: ANOVA F Table showing the predictors of the number of network links.

```
cen.aov.xtab <- xtable::xtable(cen.aov,
  caption =
    "ANOVA F Table showing the predictors of network centralization.",
  label = "tab:cen_aov"
)
print(cen.aov.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
BR	1	3.77	3.77	2.17	0.1463
PC	1	6.46	6.46	3.72	0.0590
SR	1	56.48	56.48	32.55	0.0000
Residuals	53	91.95	1.73		

Table 7: ANOVA F Table showing the predictors of network centralization.

```
# networks and network metrics
# permanova
cn.L.cen.perm.xtab <- xtable::xtable(cn.L.cen.perm,
  caption =
    "PerMANOVA Pseudo-F Table showing the predictors of network similarity.",
  label = "tab:cn_L_cen_perm"
)
print(cn.L.cen.perm.xtab,
  type = "latex",
  include.rownames = TRUE,
  include.colnames = TRUE
)
```

% latex table generated in R 3.6.1 by xtable 1.8-4 package % Tue Oct 15 16:43:16 2019

	Df	SumOfSqs	R2	F	Pr(>F)
L	1	1330.80	0.86	734.67	0.0010
Cen	1	118.99	0.08	65.69	0.0010
Residual	54	97.82	0.06		
Total	56	1547.61	1.00		

Table 8: PerMANOVA Pseudo-F Table showing the predictors of network similarity.

Plots

Figure: Genotype barplots Community composition NMDS with vectors

```
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1) / 1)
chp.coord <- ch.plot(ord.com[, 1:2], onc.geno,
  cex = 2, mu.pch = 19,
  pt.col = "white",
  bar.col = "darkgrey"
)
text(chp.coord, labels = rownames(chp.coord))
plot(vec.com, col = "black", lwd = 7)
```

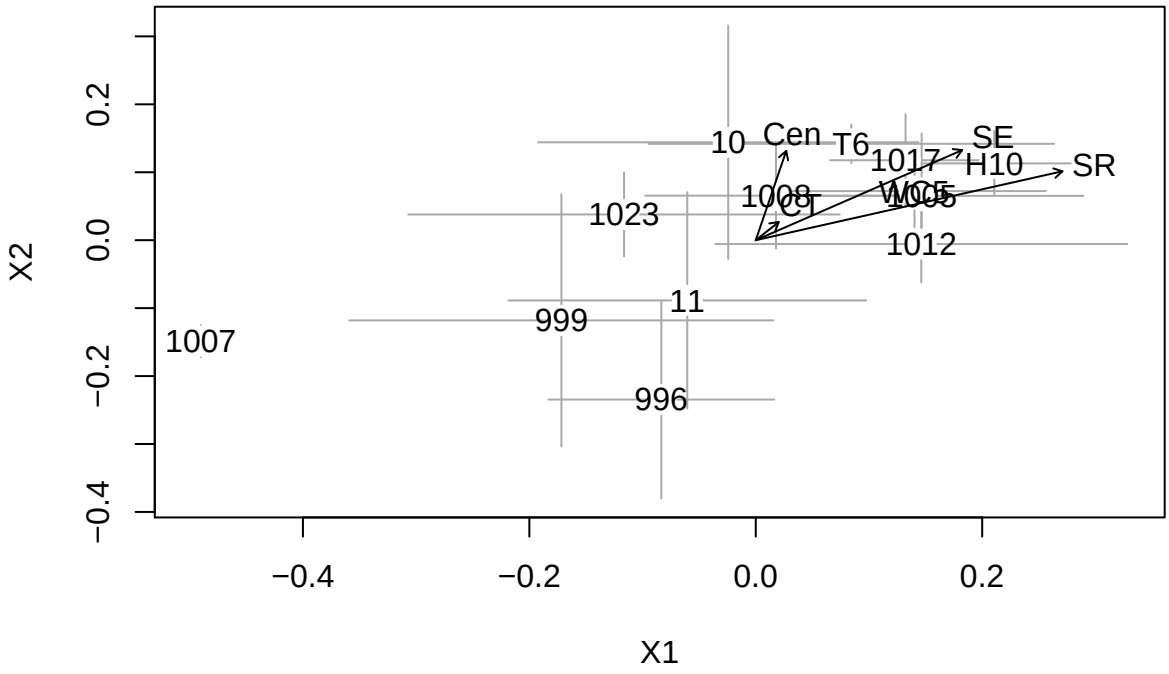


Figure: Lichen networks

```
par(mfrow = c(2, 2), mar = c(0, 0.1, 1.0, 0.1))
set.seed(123)
net.col <- sign(meanNet(cn.mu.onc))
net.col[net.col == -1] <- 2
net.col[net.col == 1] <- 1
net.elwd <- (abs(meanNet(cn.mu.onc)) * 10)^2
coord <- gplot(abs(meanNet(cn.mu.onc)),
  gmode = "digraph",
  displaylabels = TRUE,
  edge.lwd = net.elwd,
  edge.col = net.col,
  vertex.col = "black",
  vertex.cex = 0.5,
  arrowhead.cex = 0.5,
  label.cex = 1,
```

```
  main = "All Genotypes"
)
cn.mu.plot <- cn.mu.onc[names(cn.mu.onc) %in%
  c("996", "WC5", "1008")]
cn.mu.plot <- cn.mu.plot[
  order(unlist(lapply(
    cn.mu.plot,
    function(x) sum(abs(sign(x)))
  )))
]
for (i in 1:length(cn.mu.plot)) {
  net.col <- sign(cn.mu.plot[[i]])
  net.col[net.col == -1] <- 2
  net.col[net.col == 1] <- 1
  net.elwd <- (abs(cn.mu.plot[[i]]) * 10)^2
  set.seed(123)
  gplot(abs(cn.mu.plot[[i]]),
    gmode = "digraph",
    displaylabels = TRUE,
    coord = coord,
    edge.lwd = net.elwd,
    edge.col = net.col,
    vertex.col = "black",
    vertex.cex = 0.5,
    arrowhead.cex = 0.5,
    label.cex = 1,
    main = names(cn.mu.plot)[i]
  )
}
```

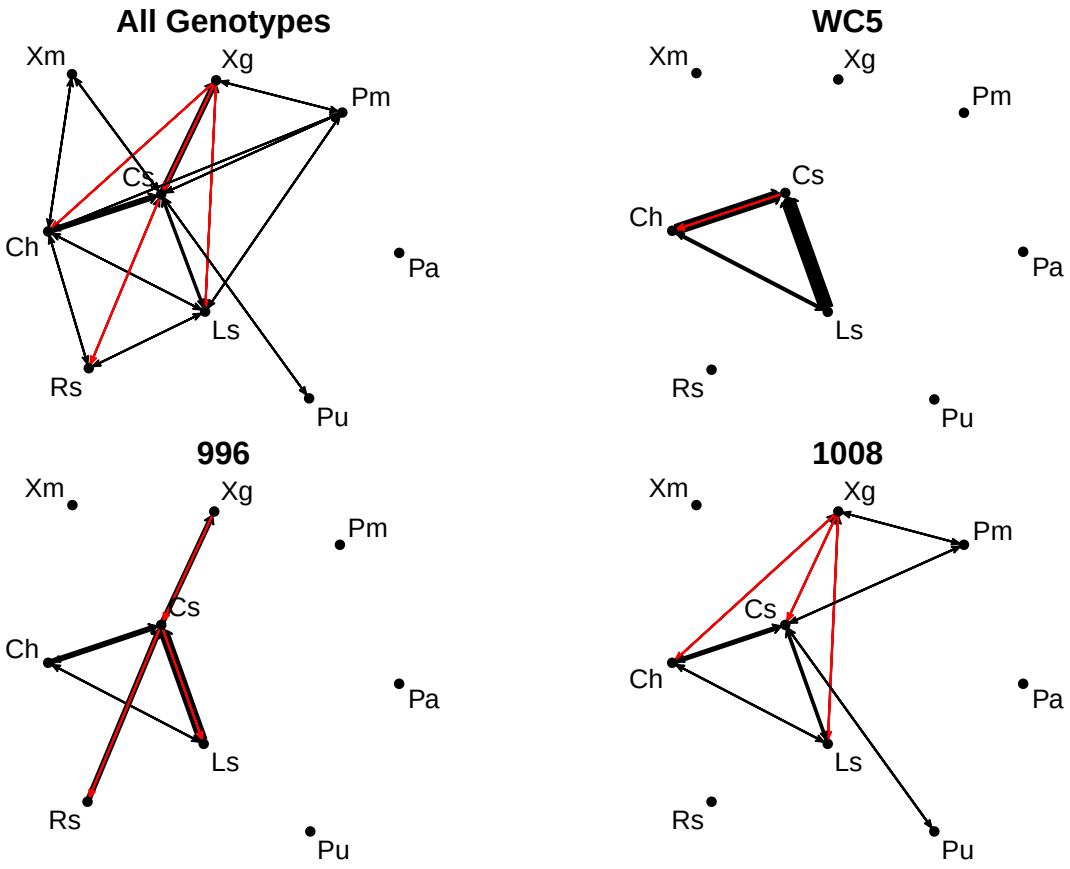


Figure: Genotype network similarity by genotype

```
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
chp.coord <- ch.plot(cn.nms.onc, onc.geno,
  cex = 2.65, lwd = 2.5, mu.pch = 15,
  pt.col = "white",
  bar.col = "darkgrey"
)
text(chp.coord, labels = rownames(chp.coord), cex = 0.65)
plot(vec.cn, col = "black", lwd = 5)
```

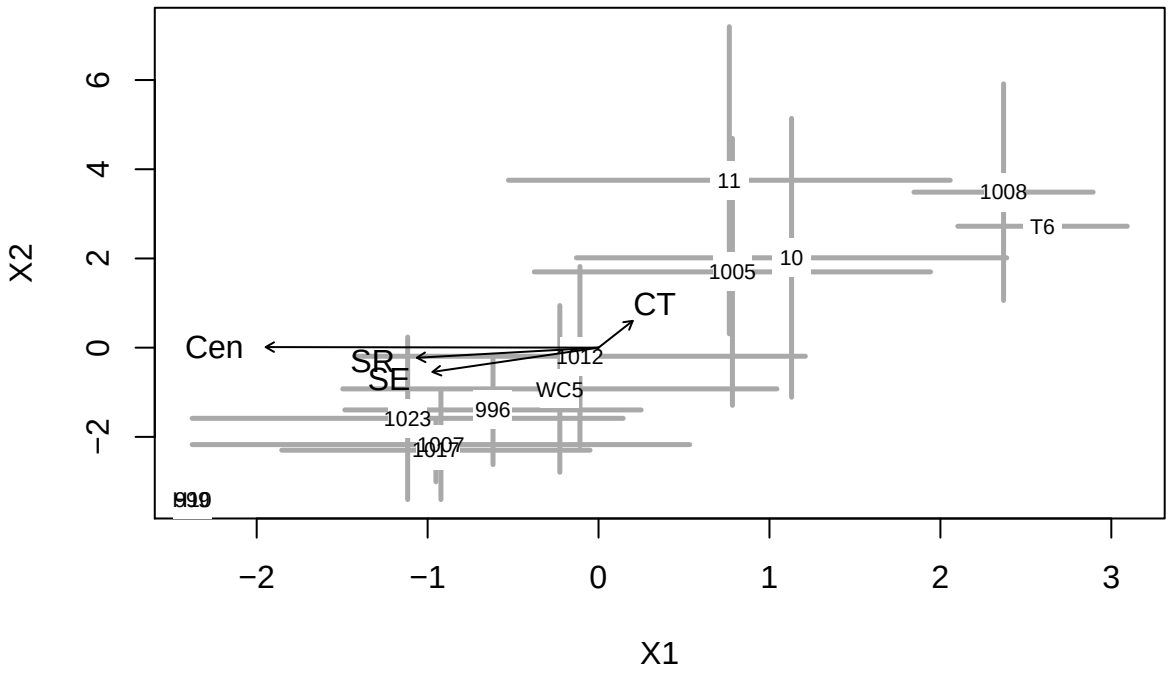
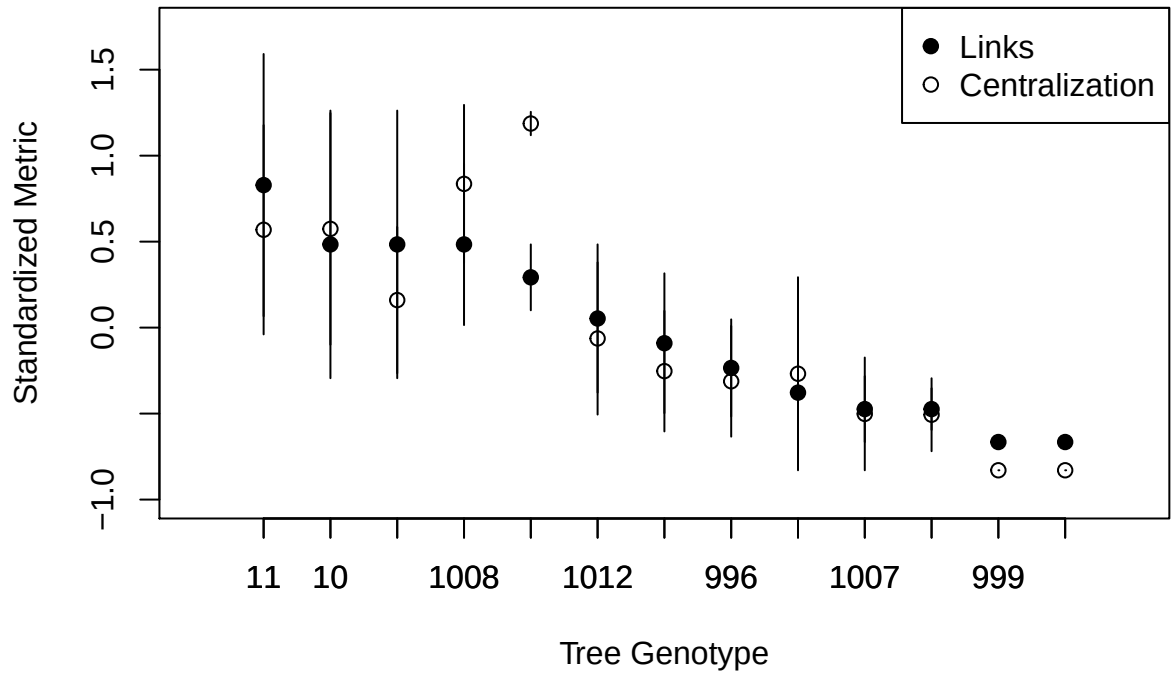


Figure: A) Lichen networks

Figure: (A) Linkage and centrality by genotype and (B) Total cover and species richness predict L and Cen

```
mdc.plot(onc.dat[, "geno"], onc.dat[, "L"],
  ylim = c(-1, 1.75),
  xlab = "Tree Genotype", ylab = "Standardized Metric",
  ord = order(tapply(onc.dat[, "L"], onc.dat[, "geno"], mean), decreasing = TRUE)
)
mdc.plot(onc.dat[, "geno"], onc.dat[, "Cen"],
  add = TRUE, pch = 1,
  ord = order(tapply(onc.dat[, "L"], onc.dat[, "geno"], mean), decreasing = TRUE)
)
legend("topright", legend = c("Links", "Centralization"), pch = c(19, 1), bty = "none")
```



Supplementary Figure: Lichen size distribution

```
plot(density(xgs.data$median.thallus),
  xlab = "Median Lichen Thallus Area (cm^2)",
  main = ""
)
abline(v = median(xgs.data$median.thallus, na.rm = TRUE), lty = 2)
```

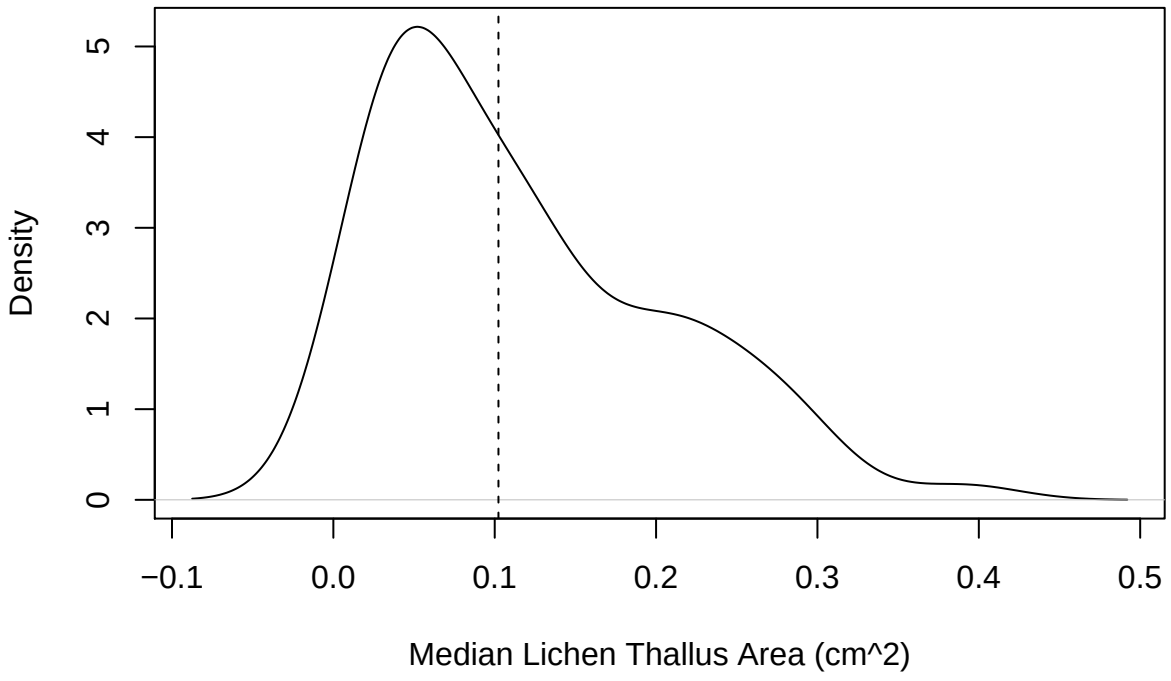
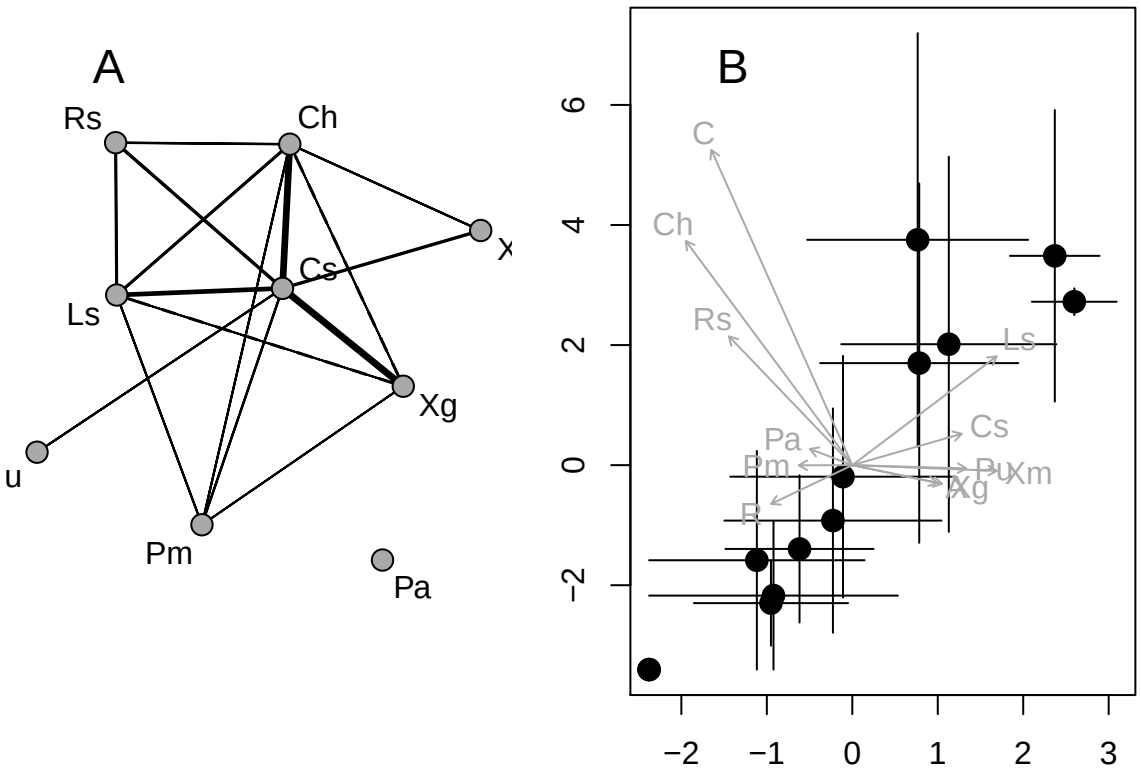
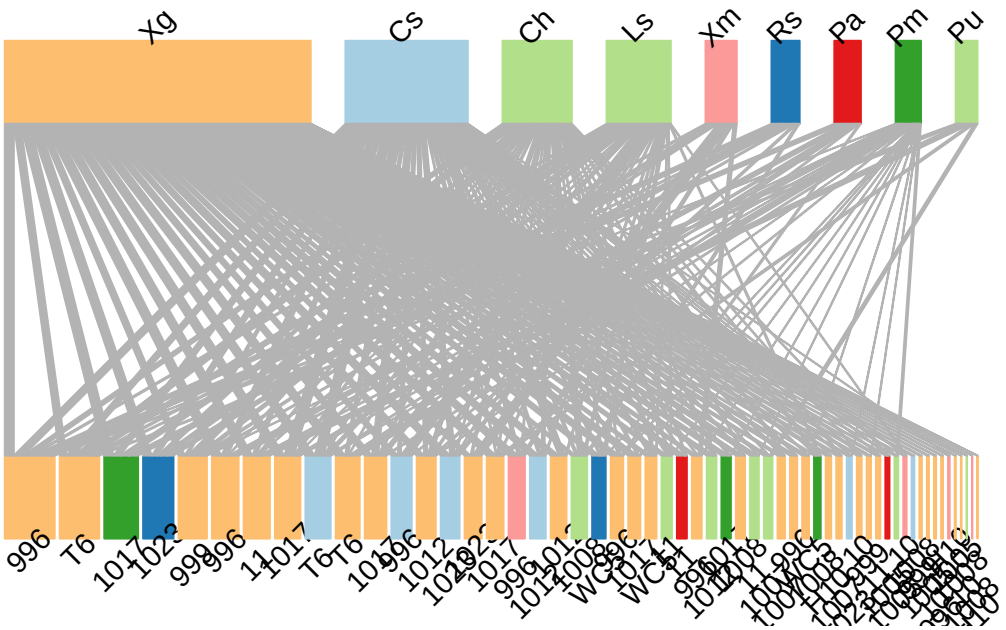


Figure 2

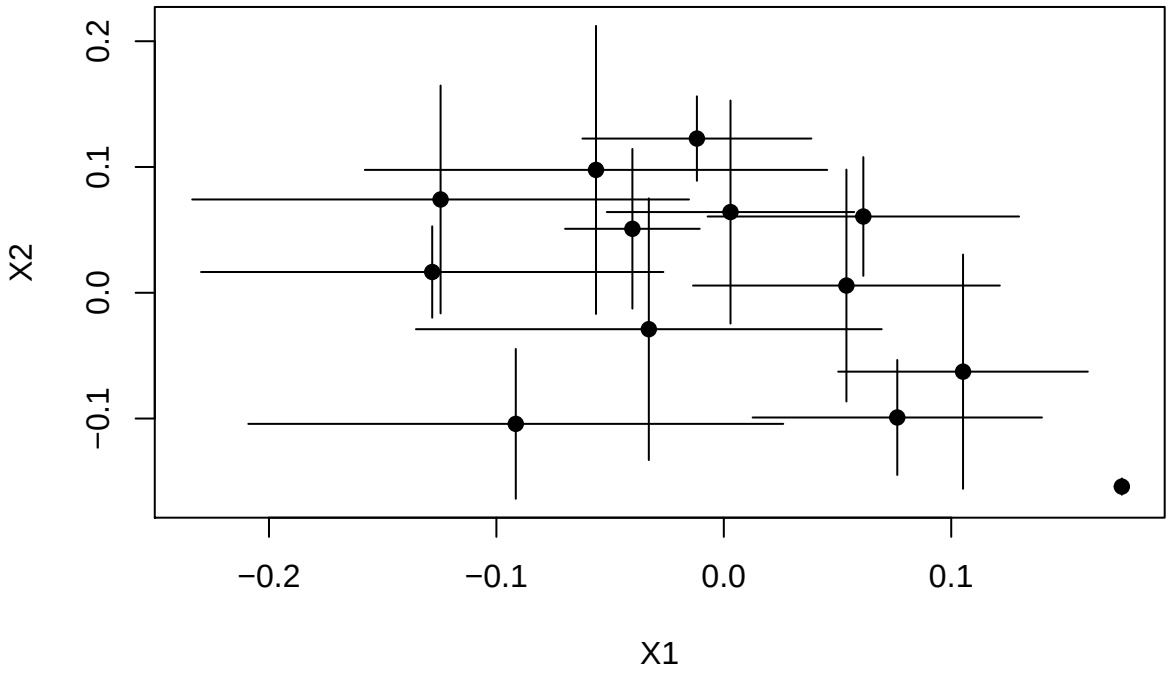
```
par(mfrow = c(1, 2), mar = c(5.1, 4.1, 4.1, 2.1) / 2)
gplot(meanNet(cn.mu.onc),
  gmode = "graph",
  displaylabels = TRUE,
  edge.lwd = meanNet(cn.mu.onc) * 20,
  vertex.col = "darkgrey"
)
legend("topleft", legend = "A", bty = "n", cex = 1.5)
chp.coord <- ch.plot(cn.nms.onc, onc.geno, cex = 1.5)
plot(nv.onc, col = "darkgrey")
legend("topleft", legend = "B", bty = "n", cex = 1.5)
```




```
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
bipartite::plotweb(pw.onc,
  method = "normal",
  text.rot = 45,
  col.low = col.pal[mods.onc$tree],
  col.high = col.pal[mods.onc$sp],
  bor.col.low = col.pal[mods.onc$tree],
  bor.col.high = col.pal[mods.onc$sp],
  col.interaction = "grey70",
  bor.col.interaction = "grey70",
  labsize = 1.5
)
```




```
ch.plot(nms.onc, onc.geno)
```

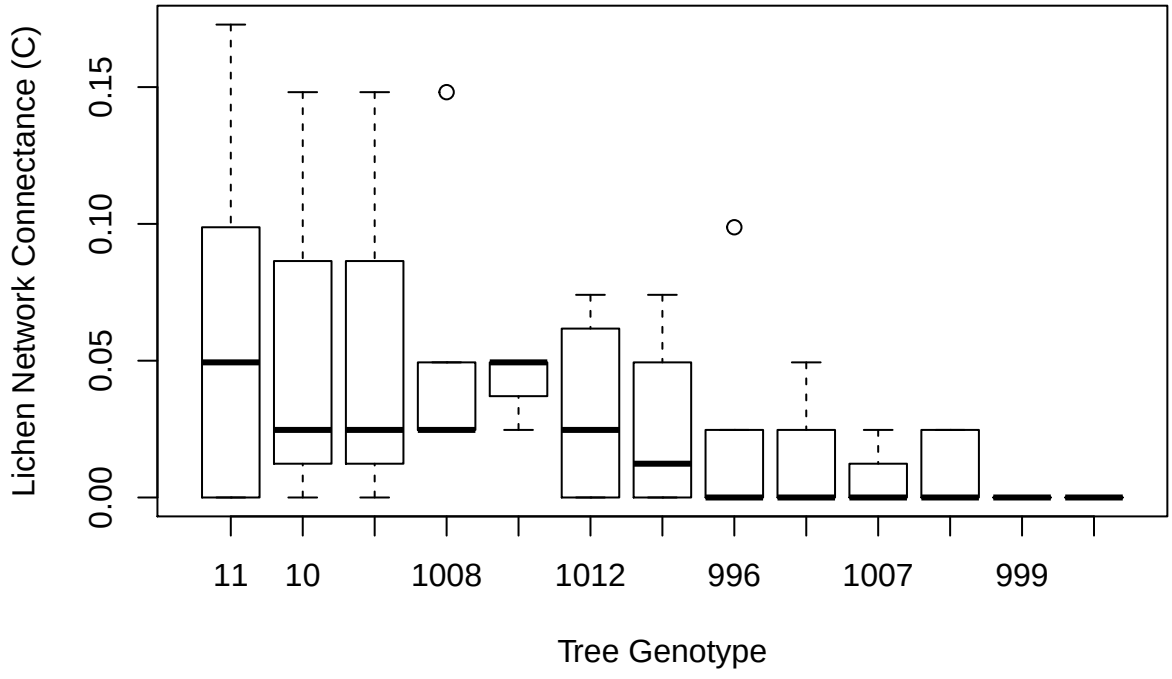



```
##                X1          X2
## 10   -0.091479884 -0.104244255
## 1005 -0.124540472  0.074171596
## 1007  0.175012652 -0.154102531
## 1008  0.061349598  0.060637950
## 1012 -0.056201035  0.097682862
## 1017 -0.040197275  0.050863036
## 1023  0.053916695  0.005735269
## 11   -0.032978298 -0.028980365
## 996   0.076271011 -0.099167595
## 999   0.105185439 -0.062733516
```

```
## H10  -0.011830997  0.122603983
## T6    0.002941633  0.064173827
## WC5  -0.128224482  0.016507373
```

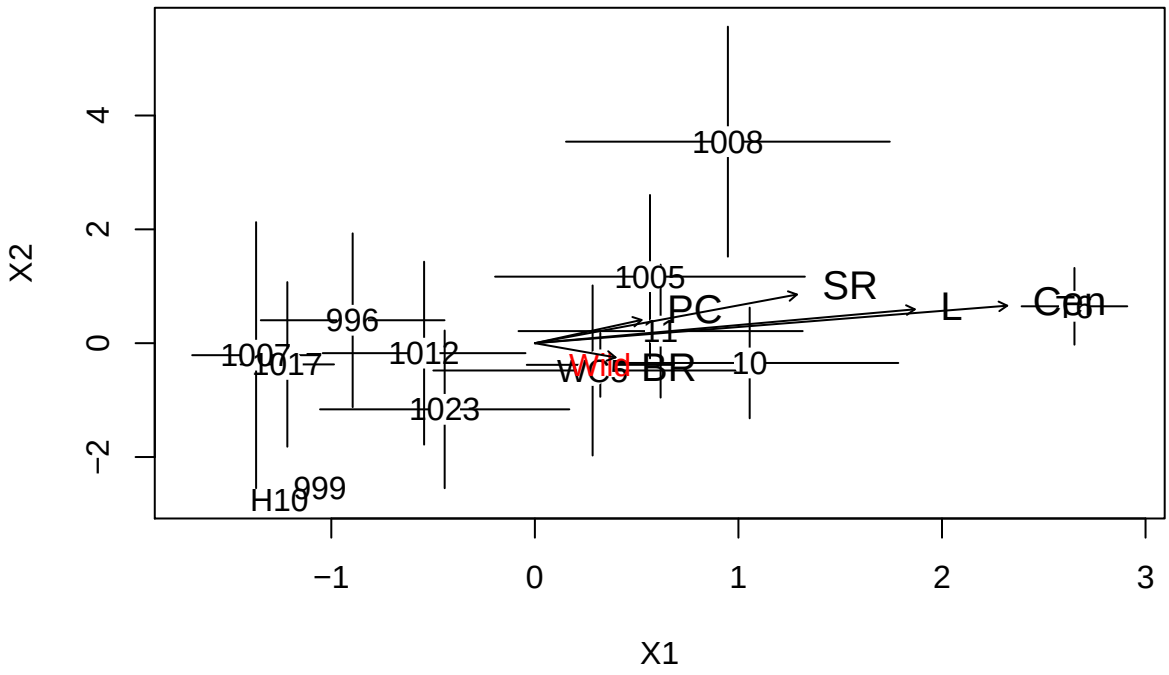
```
## plot(cv.onc, col = "grey30")
## legend("topleft", legend = "A")
```

```
g.order <- tapply(ns.onc[, "C"], onc.geno, mean)
g.order <- names(g.order)[order(g.order, decreasing = TRUE)]
onc.g <- factor(onc.geno, levels = g.order)
plot(ns.onc[, "C"] ~ onc.g, xlab = "Tree Genotype", ylab = "Lichen Network Connectance (C)")
```



Which wild uintah trees are similar to garden trees?

```
coords <- ch.plot(cn.nms.all, cn.nms.geno, mu.pch = "", cex = 2)
points(coords, pch = 19, col = "white", cex = 2)
text(coords[!grepl("wild", rownames(coords)), ],
  labels = rownames(coords)[!grepl("wild", rownames(coords))],
  col = "black"
)
text(coords[grep("wild", rownames(coords)), 1],
  coords[grep("wild", rownames(coords)), 2],
  labels = "Wild", col = "red"
)
plot(vec.all, col = "black", cex = 1.23)
```



Send results to manuscript

```
manuscript.dir <- "../../lcn_manuscript"
### Send tables and figures to manuscript directory
if (exists("manuscript.dir")) {
  tabs.figs <- dir(manuscript.dir)
  tab.fig.update <- dir("../results/lcn_notebook_files/figure-latex/",
    full.names = TRUE
  )[
    dir("../results/lcn_notebook_files/figure-latex/") %in% tabs.figs
  ]
  tab.fig.update <- c(
    tab.fig.update,
    dir("../docs", full.names = TRUE)[dir("../docs") %in% tabs.figs]
  )
  sapply(tab.fig.update, file.copy, to = manuscript.dir, overwrite = TRUE)
  # supplementary figures
  si.dir <- paste0(manuscript.dir, "/supplement")
  si <- dir(si.dir)
  si.update <- dir("../results/lcn_notebook_files/figure-latex/",
    full.names = TRUE
  )[
    dir("../results/lcn_notebook_files/figure-latex/") %in% si
  ]
  si.update <- c(si.update, dir("../docs", full.names = TRUE)[dir("../docs") %in% si])
  sapply(si.update, file.copy,
    to = si.dir,
    overwrite = TRUE
  )
}
```

```
## named list()
```