

# LCN: Lichen interaction network study

*MK Lau*

## Results

```
### REML

### We know from Lamit's dissertation work that lichen communities are
### heritable, largely driven by bark roughness
### Do we find similar patterns?

## Create a list to generate a results table
h2.tab <- matrix("", 8, 4)
colnames(h2.tab) <- c("Response", "H2", "R2", "p-value")

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

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 2.9627, p-value = 0.0352
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
shapiro.test(residuals(ptc.reml))

##
## Shapiro-Wilk normality test
##
## data: residuals(ptc.reml)
## W = 0.95096, p-value = 0.02174
h2.tab[1, "p-value"] <- ptc.reml.pval$"p.value"
h2.tab[1, "H2"] <- H2(ptc.reml, g = onc.dat$geno)
h2.tab[1, "R2"] <- R2(ptc.reml)

## Warning: 'r.squaredGLMM' now calculates a revised statistic. See the help
## page.
```

```

R2(ptc.reml)

##          R2c
## 0.1727875
h2.tab[1, "Response"] <- "Percent Lichen Cover"

## Species richness ~ genotype
spr.reml <- lme4::lmer(I(SR^(1/2)) ~ (1 | geno),
                     data = 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 = 1.0001, p-value = 0.1355
shapiro.test(residuals(spr.reml))

##
## Shapiro-Wilk normality test
##
## data: residuals(spr.reml)
## W = 0.97364, p-value = 0.2467
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
h2.tab[2, "p-value"] <- spr.reml.pval$"p.value"
h2.tab[2, "H2"] <- H2(spr.reml, g = onc.dat$geno)
h2.tab[2, "R2"] <- R2(spr.reml)
R2(spr.reml)

##          R2c
## 0.09814791
h2.tab[2, "Response"] <- "Lichen Species Richness"

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

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

```

```

##
## data:
## RLRT = 10.69, p-value = 1e-04
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 = 0.97975, p-value = 0.4529
h2.tab[3, "p-value"] <- prb.reml.pval$"p.value"
h2.tab[3, "H2"] <- H2(prb.reml, g = onc.dat$geno)
h2.tab[3, "R2"] <- R2(prb.reml)
R2(prb.reml)

##          R2c
## 0.3783496
h2.tab[3, "Response"] <- "Percent Rough Bark"

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

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 4.3224, p-value = 0.015
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 = 0.74892, p-value = 2.431e-08
## CN ratio REML
cnr.reml <- lme4::lmer(I(CN^(1/1)) ~ (1 | geno), data = onc.dat, REML = TRUE)

## boundary (singular) fit: see ?isSingular
cnr.reml.pval <- RLRsim::exactRLRT(cnr.reml)
cnr.reml.pval

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0, p-value = 1
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 = 0.92183, p-value = 0.001754
## Is species richness correlated with percent cover?
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)

rcom.ng.perm
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = onc.com.rel^(1/1) ~ BR + PC + SR, data = onc.dat, permutations = 10000, mrank = TRUE)
##          Df SumOfSqs      R2      F    Pr(>F)
## BR         1   0.4398 0.03889   3.7408 0.006399 **
## PC         1   3.8618 0.34151  32.8482 9.999e-05 ***
## SR         1   0.7754 0.06857   6.5958 0.000200 ***
## Residual  53   6.2309 0.55102
## Total     56  11.3079 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

rcom.perm
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 10000
##
## vegan::adonis2(formula = onc.com.rel^(1/1) ~ geno + BR + PC + SR, data = onc.dat, permutations = 10000, mrank = TRUE)
##          Df SumOfSqs      R2      F    Pr(>F)
## geno      12   2.7463 0.24287   1.8221 0.0029 **
## BR         1   0.1248 0.01104   0.9938 0.3841
## PC         1   2.6711 0.23622  21.2661 9.999e-05 ***
```

```

## SR      1    0.6159 0.05447  4.9036    0.0011 **
## Residual 41    5.1498 0.45541
## Total   56   11.3079 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

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 0.08300000 0.91800000 0.13900000 0.05293002 0.13002104

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 0.31000000 0.02651488 0.13049408

## Partial Mantels using RFLP distance
ecodist::mantel(cn.mu.d.onc ~ rflp.d)

##      mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## -0.00603936 0.57200000 0.42900000 0.97600000 -0.13858547 0.17659829

ecodist::mantel(onc.com.mu.d ~ rflp.d)

##      mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## 0.1179051 0.2760000 0.7250000 0.4980000 -0.2412808 0.2447645

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

##      mantelr      pval1      pval2      pval3  llim.2.5% ulim.97.5%
## 0.29000439 0.08900000 0.91200000 0.08900000 -0.02092565 0.42837710

## Was lichen network similarity determined by genotype?
set.seed(1234)
cn.perm <- vegan::adonis2(cn.d.onc ~ geno + BR + PC + SR,
                          data = onc.dat, permutations = 10000, mrank = TRUE)
set.seed(1234)
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
##
##      Df SumOfSqs      R2      F      Pr(>F)
## BR      1      61.42 0.03968  4.1680    0.03770 *

```

```

## PC      1    49.47 0.03197  3.3573   0.06779 .
## 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 ~ geno + BR + PC + SR, data = onc.dat, permutations = 10000, mrank
##      Df SumOfSqs      R2      F    Pr(>F)
## geno   12   450.52 0.29111  2.6902 0.009399 **
## BR      1    29.11 0.01881  2.0858 0.146985
## PC      1    30.01 0.01939  2.1504 0.144086
## SR      1   465.78 0.30097 33.3755 9.999e-05 ***
## Residual 41   572.18 0.36972
## Total   56   1547.61 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 0.1362
## 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)
link.reml.pval

##
## simulated finite sample distribution of RLRT.
##

```



```

## (p-value based on 50000 simulated values)
##
## data:
## RLRT = 2.0484, p-value = 0.06602
fligner.test(log(onc.dat$L + 0.000001), onc.dat$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
h2.tab[6, "p-value"] <- link.reml.pval$"p.value"
h2.tab[6, "H2"] <- H2(link.reml, g = onc.dat$geno)
h2.tab[6, "R2"] <- R2(link.reml)
R2(link.reml)

## R2c
## 0.1701568
h2.tab[6, "Response"] <- "Number of Network Links"

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

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 50000 simulated values)
##
## data:
## RLRT = 2.7801, p-value = 0.04076
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

h2.tab[7, "p-value"] <- cen.reml.pval$"p.value"
h2.tab[7, "H2"] <- H2(cen.reml, g = onc.dat$geno)
h2.tab[7, "R2"] <- R2(cen.reml)
R2(cen.reml)

##          R2c
## 0.2016649

h2.tab[7, "Response"] <- "Network Centrality"

                                # 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)
mod.reml.pval

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.23363, p-value = 0.2809

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

h2.tab[8, "p-value"] <- mod.reml.pval$"p.value"
h2.tab[8, "H2"] <- H2(mod.reml, g = onc.dat$geno)
h2.tab[8, "R2"] <- R2(mod.reml)
h2.tab[8, "Response"] <- "Network Modularity"

                                # 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

mod.aov <- aov(I(onc.ns[, "mod.lik"]^(1/4)) ~ BR + PC + SR, data = onc.dat)
summary(mod.aov)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## BR              1  0.0442   0.0442   0.787    0.379
## PC              1  0.0879   0.0879   1.564    0.217
## SR              1  1.3799   1.3799  24.558 7.76e-06 ***
## Residuals      53  2.9781   0.0562
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

shapiro.test(residuals((mod.aov)))

##
##  Shapiro-Wilk normality test
##
## data:  residuals((mod.aov))
## W = 0.9201, p-value = 0.001078

##
cor.test(onc.ns[, "L"], onc.ns[, "Cen"])

##
##  Pearson's product-moment correlation
##
## data:  onc.ns[, "L"] and onc.ns[, "Cen"]
```

```

## t = 13.37, df = 55, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.7950728 0.9244074
## sample estimates:
## cor
## 0.8744752

# 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 = 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 0.127 0.000 0.258 0.201 0.000 0.000

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

## Cs Ch Ls Rs Xg Pm
## 0.73204678 0.54157218 0.39722829 0.18378675 0.14553120 0.07914127
## Xm Pu Pa
## 0.06376775 0.02105263 0.00000000

## Ordinations
### nits = 10,
### iconf = random
### epsilon = 1e-12 = acceptable change in stress
### maxit = 500 = maximum number of iterations
ord.com <- nmds.min(nms.com, 3)

## Minimum stress for given dimensionality: 0.1008923
## r^2 for minimum stress configuration: 0.9357192
## 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: 0.01065177
## r^2 for minimum stress configuration: 0.9993026

```

```

## Minimum stress for given dimensionality: 0.01065177
## r^2 for minimum stress configuration: 0.9993026
## checking variance explained by ordinations
ord1.cn.reml <- lme4::lmer(I(ord.cn[, 1]^(1/1)) ~ (1 | geno),
  data = onc.dat, REML = TRUE)
ord2.cn.reml <- lme4::lmer(I(ord.cn[, 2]^(1/1)) ~ (1 | geno),
  data = onc.dat, REML = TRUE)
ord1.cn.reml.pval <- RLRsim::exactRLRT(ord1.cn.reml)
ord2.cn.reml.pval <- RLRsim::exactRLRT(ord2.cn.reml)
ord1.cn.reml.pval

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 1.0221, p-value = 0.134
ord2.cn.reml.pval

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.5618, p-value = 0.2049
fligner.test(ord.cn[, 1]^(1/1), onc.dat$geno)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ord.cn[, 1]^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 16.805, df = 12, p-value =
## 0.1571
fligner.test(ord.cn[, 2]^(1/1), onc.dat$geno)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ord.cn[, 2]^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 9.9165, df = 12, p-value =
## 0.6233
ord1.com.reml <- lme4::lmer(I(ord.com[, 1]^(1/1)) ~ (1 | geno),
  data = onc.dat, REML = TRUE)
ord2.com.reml <- lme4::lmer(I(ord.com[, 2]^(1/1)) ~ (1 | geno),
  data = onc.dat, REML = TRUE)
ord1.com.reml.pval <- RLRsim::exactRLRT(ord1.com.reml)
ord2.com.reml.pval <- RLRsim::exactRLRT(ord2.com.reml)
ord1.com.reml.pval

##
## simulated finite sample distribution of RLRT.

```

```

##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.1669, p-value = 0.3035
ord2.com.reml.pval

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.98197, p-value = 0.1381
fligner.test(ord.com[, 1]^(1/1), onc.dat$geno)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ord.com[, 1]^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 9.3187, df = 12, p-value =
## 0.6755
fligner.test(ord.com[, 2]^(1/1), onc.dat$geno)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ord.com[, 2]^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 16.947, df = 12, p-value =
## 0.1516
fligner.test(ord.com[, 3]^(1/1), onc.dat$geno)

##
## Fligner-Killeen test of homogeneity of variances
##
## data: ord.com[, 3]^(1/1) and onc.dat$geno
## Fligner-Killeen:med chi-squared = 14.943, df = 12, p-value =
## 0.2446
summary(lm(ord.cn[, 1] ~ SR + PC, data = onc.dat))

##
## Call:
## lm(formula = ord.cn[, 1] ~ SR + PC, data = onc.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.6007  -1.7887   0.1726   2.2110   6.7059
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.77927    1.05175   3.593 0.000706 ***
## SR            -2.89115    0.39475  -7.324 1.23e-09 ***
## PC             0.10728    0.02215   4.844 1.11e-05 ***

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.596 on 54 degrees of freedom
## Multiple R-squared:  0.5025, Adjusted R-squared:  0.4841
## F-statistic: 27.27 on 2 and 54 DF,  p-value: 6.508e-09
summary(lm(ord.cn[, 2] ~ SR + PC, data = onc.dat))

##
## Call:
## lm(formula = ord.cn[, 2] ~ SR + PC, data = onc.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4080 -0.9426 -0.6151  1.3669  2.9279
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.143124   0.420811  -2.716 0.008846 **
## SR           0.561645   0.157944   3.556 0.000793 ***
## PC          -0.013722   0.008862  -1.548 0.127384
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.439 on 54 degrees of freedom
## Multiple R-squared:  0.2223, Adjusted R-squared:  0.1935
## F-statistic: 7.718 on 2 and 54 DF,  p-value: 0.001127
summary(lm(ord.com[, 1] ~ SR + PC, data = onc.dat))

##
## Call:
## lm(formula = ord.com[, 1] ~ SR + PC, data = onc.dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.18241 -0.09091 -0.01606  0.05475  0.65204
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.5145496   0.0395271 -13.018 < 2e-16 ***
## SR           0.0527258   0.0148358   3.554 0.000798 ***
## PC           0.0058018   0.0008324   6.970 4.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1351 on 54 degrees of freedom
## Multiple R-squared:  0.8048, Adjusted R-squared:  0.7976
## F-statistic: 111.3 on 2 and 54 DF,  p-value: < 2.2e-16
summary(lm(ord.com[, 2] ~ SR + PC, data = onc.dat))

##
## Call:
## lm(formula = ord.com[, 2] ~ SR + PC, data = onc.dat)
```

```

##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54228 -0.11829  0.03558  0.16463  0.50365
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.224171   0.068196  -3.287  0.00178 **
## SR           0.015539   0.025596   0.607  0.54634
## PC           0.002973   0.001436   2.070  0.04328 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2331 on 54 degrees of freedom
## Multiple R-squared:  0.2151, Adjusted R-squared:  0.1861
## F-statistic:  7.4 on 2 and 54 DF,  p-value: 0.001444
## 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 = 0.0474
RLRsim::exactRLRT(xgs.median.reml)

##
## simulated finite sample distribution of RLRT.
##
## (p-value based on 10000 simulated values)
##
## data:
## RLRT = 0.092023, p-value = 0.3418
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)
                                # 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 <- netDist(cn.all, method = "bc")

if (!exists("cn.nms.all")){cn.nms.all <- nmds.min(nmds(cn.d.all, 2, 2))}

## Minimum stress for given dimensionality: 0.04194377
## 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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:29 2019

Response	H2	R2	p-value
Percent Rough Bark	0.37835	0.37835	1e-04
Network Centrality	0.20166	0.20166	0.04076
Percent Lichen Cover	0.17279	0.17279	0.0352
Number of Network Links	0.17016	0.17016	0.06602
Lichen Community Composition	0.16093	0.24287	0.0029
Lichen Species Richness	0.09815	0.09815	0.1355
Lichen Network	0.06252	0.29111	0.0094
Network Modularity	0.05731	0.05731	0.2809

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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:29 2019

	Df	SumOfSqs	R2	F	Pr(>F)
BR	1	0.44	0.04	3.74	0.0064
PC	1	3.86	0.34	32.85	0.0001
SR	1	0.78	0.07	6.60	0.0002
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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:29 2019

	Df	SumOfSqs	R2	F	Pr(>F)
geno	12	2.75	0.24	1.82	0.0029
BR	1	0.12	0.01	0.99	0.3841
PC	1	2.67	0.24	21.27	0.0001
SR	1	0.62	0.05	4.90	0.0011
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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:30 2019

	Df	SumOfSqs	R2	F	Pr(>F)
BR	1	61.42	0.04	4.17	0.0377
PC	1	49.47	0.03	3.36	0.0678
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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:30 2019

	Df	SumOfSqs	R2	F	Pr(>F)
geno	12	450.52	0.29	2.69	0.0094
BR	1	29.11	0.02	2.09	0.1470
PC	1	30.01	0.02	2.15	0.1441
SR	1	465.78	0.30	33.38	0.0001
Residual	41	572.18	0.37		
Total	56	1547.61	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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:30 2019

```

cen.aov.xtab <- xtable::xtable(cen.aov, caption =
  "ANOVA F Table showing the predictors of network centralization.",

```

	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.

```

                                label = "tab:cen_aov")
print(cen.aov.xtab,
      type = "latex",
      include.rownames = TRUE,
      include.colnames = TRUE
)

```

% latex table generated in R 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:30 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 3.5.2 by xtable 1.8-3 package % Mon Mar 25 15:40:31 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 = 4)

```

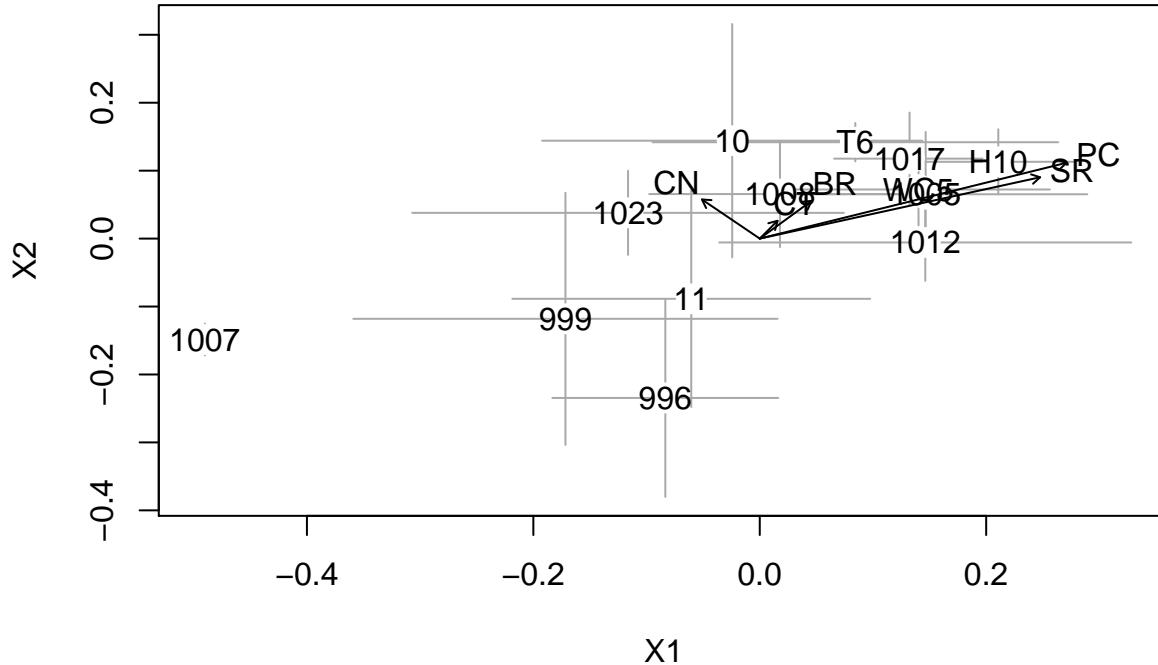


Figure: Lichen networks

```

par(mfrow = c(2, 2), mar = c(0, 0.1, 1.0, 0.1))
set.seed(123)
net.col <- sign(netMean(cn.mu.onc))
net.col[net.col == -1] <- 2
net.col[net.col == 1] <- 1
coord <- gplot(abs(netMean(cn.mu.onc)), gmode = "digraph",
  displaylabels = TRUE,
  edge.lwd = abs(netMean(cn.mu.onc)) * 20,
  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", "11", "1008")]
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
  set.seed(123)
  gplot(abs(cn.mu.plot[[i]]), gmode = "digraph",
    displaylabels = TRUE,

```

```

coord = coord,
edge.lwd = abs(cn.mu.plot[[i]]) * 20,
edge.col = net.col,
vertex.col = "black",
vertex.cex = 0.5,
arrowhead.cex = 0.5,
label.cex = 1,
main = names(cn.mu.plot)[i])
}

```

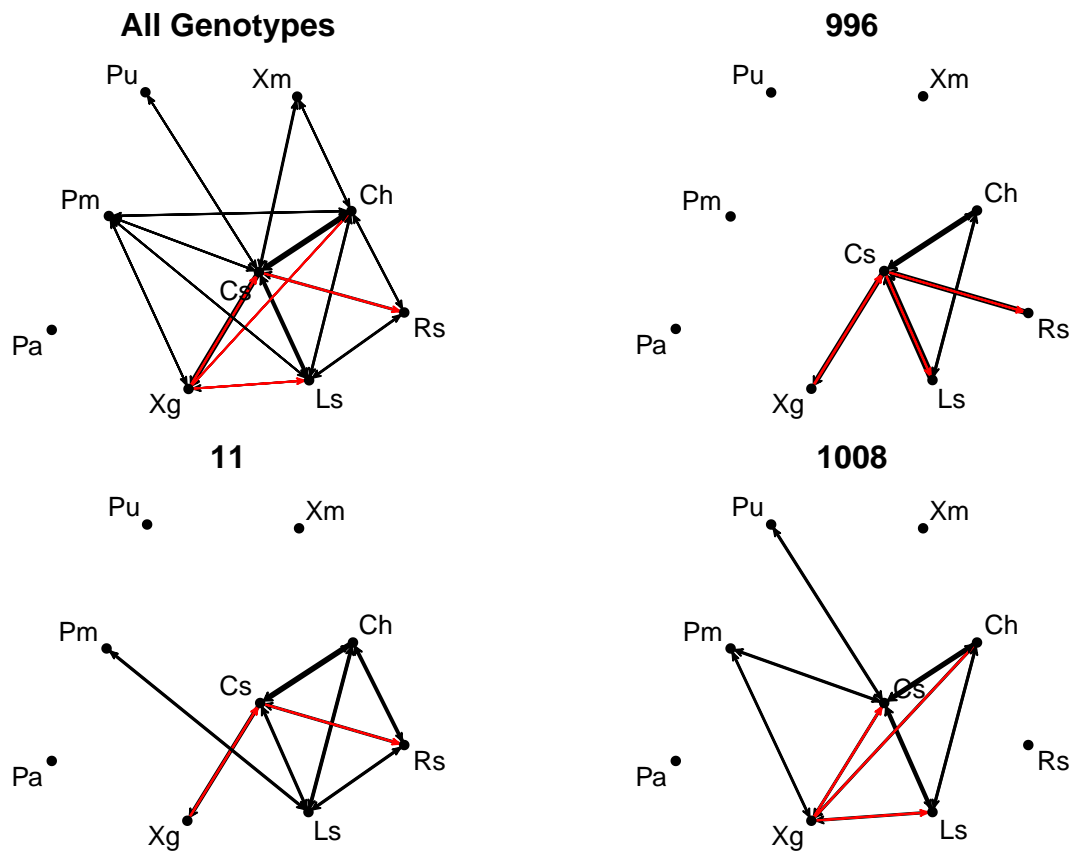


Figure: Genotype network similarity by genotype

```

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, mu.pch = 19,
  pt.col = "white",
  bar.col = "darkgrey")
text(chp.coord, labels = rownames(chp.coord))
plot(vec.cn, col = "black")

```

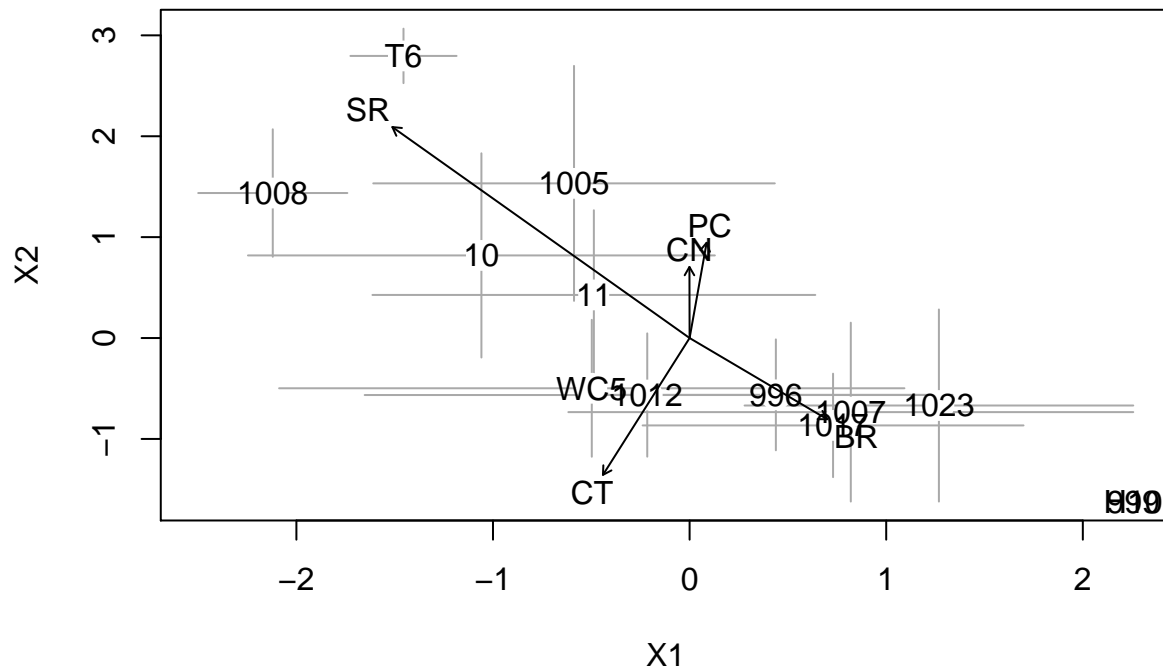
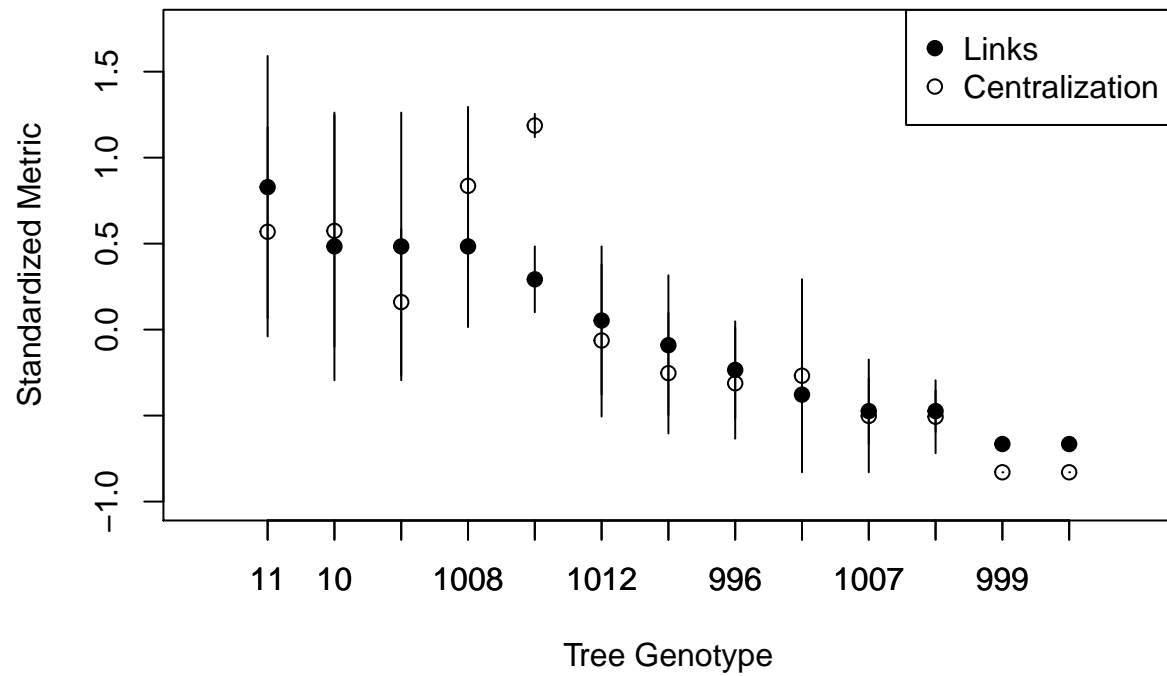


Figure: (A) Linkage and centrality by genotype and (B) Total cover and species richness predict L and Cen

```
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)
```

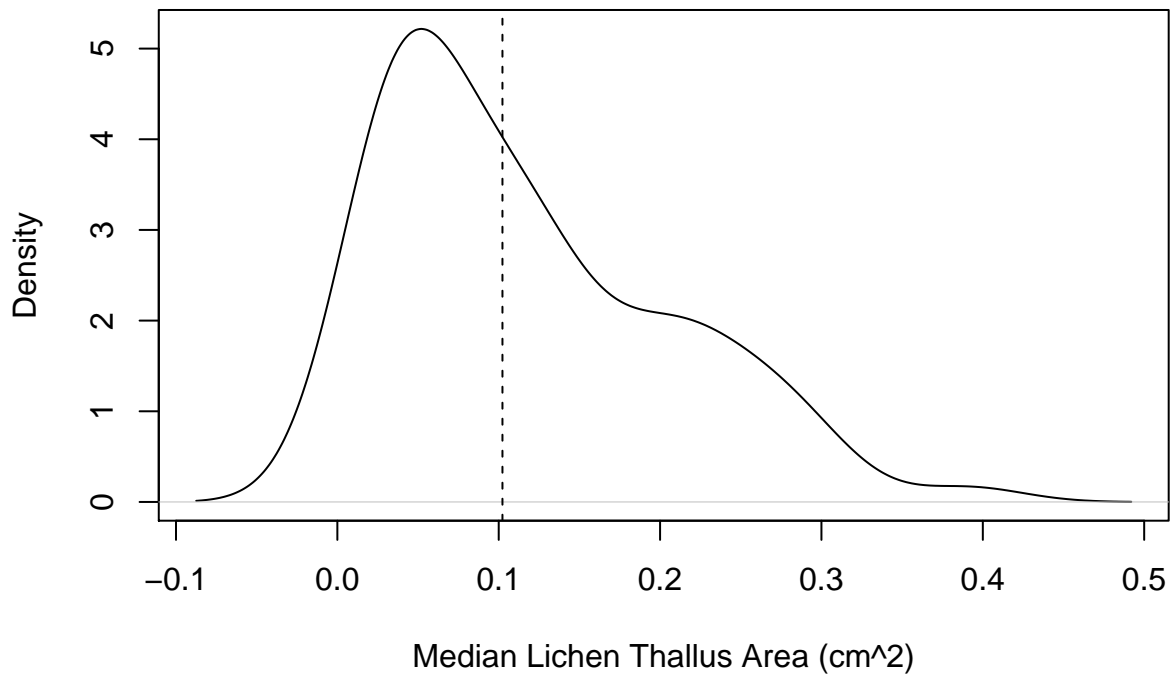
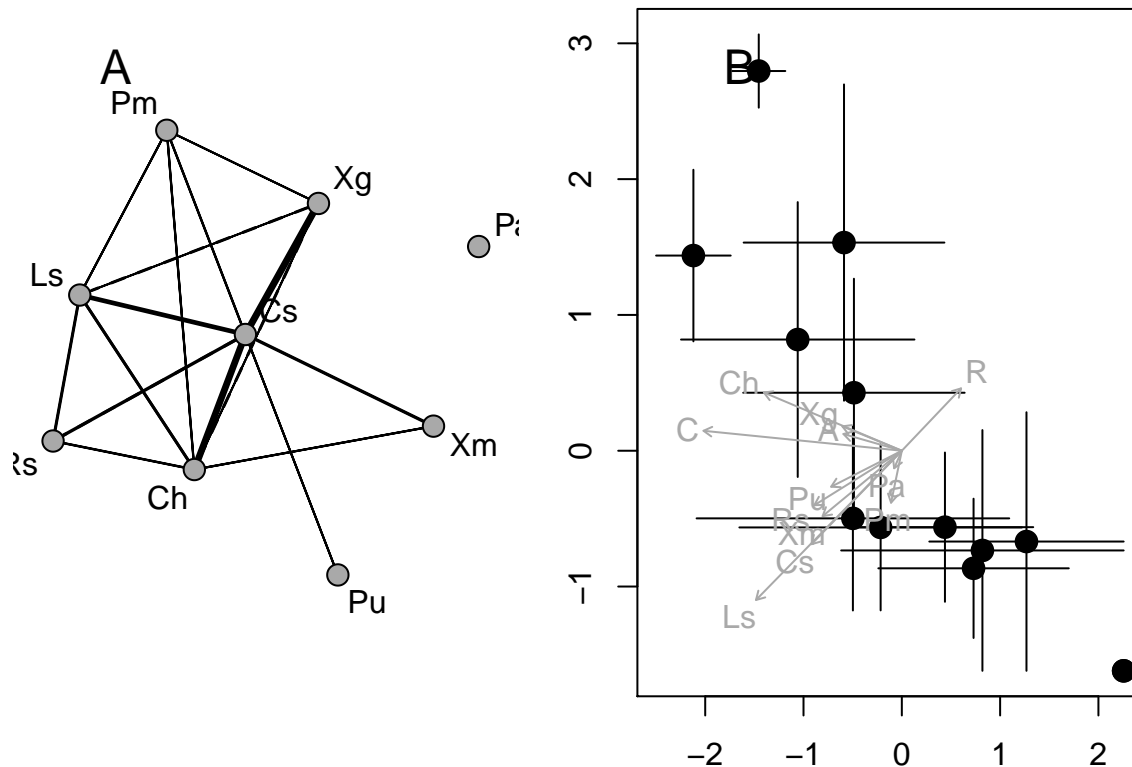




Figure 2

```
par(mfrow = c(1, 2), mar = c(5.1, 4.1, 4.1, 2.1) / 2)
gplot(netMean(cn.mu.onc), gmode = "graph",
      displaylabels = TRUE,
      edge.lwd = netMean(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)
```



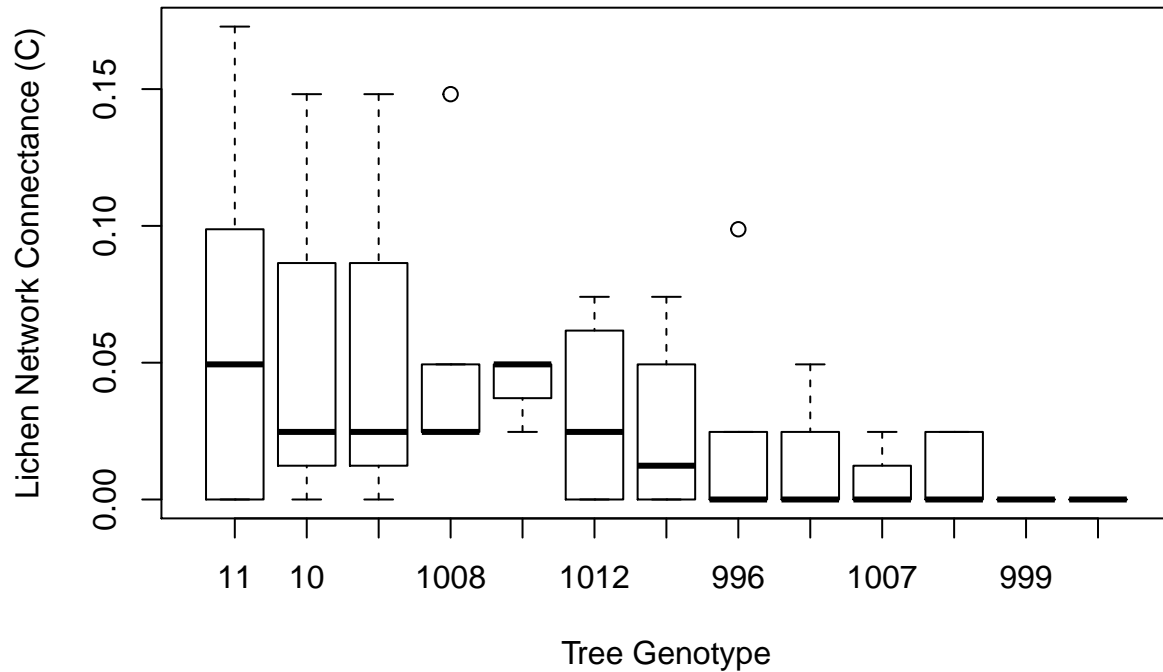
```
par(mfrow = c(1, 1), mar = c(5.1, 4.1, 4.1, 2.1))
bipartite::plotweb(pw.onc, method = "normal",
                  text.rot = 45,
                  col.low = col.pal[mods.onc$tree],
                  col.high = col.pal[mods.onc$sp],
                  bor.col.low = col.pal[mods.onc$tree],
        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



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

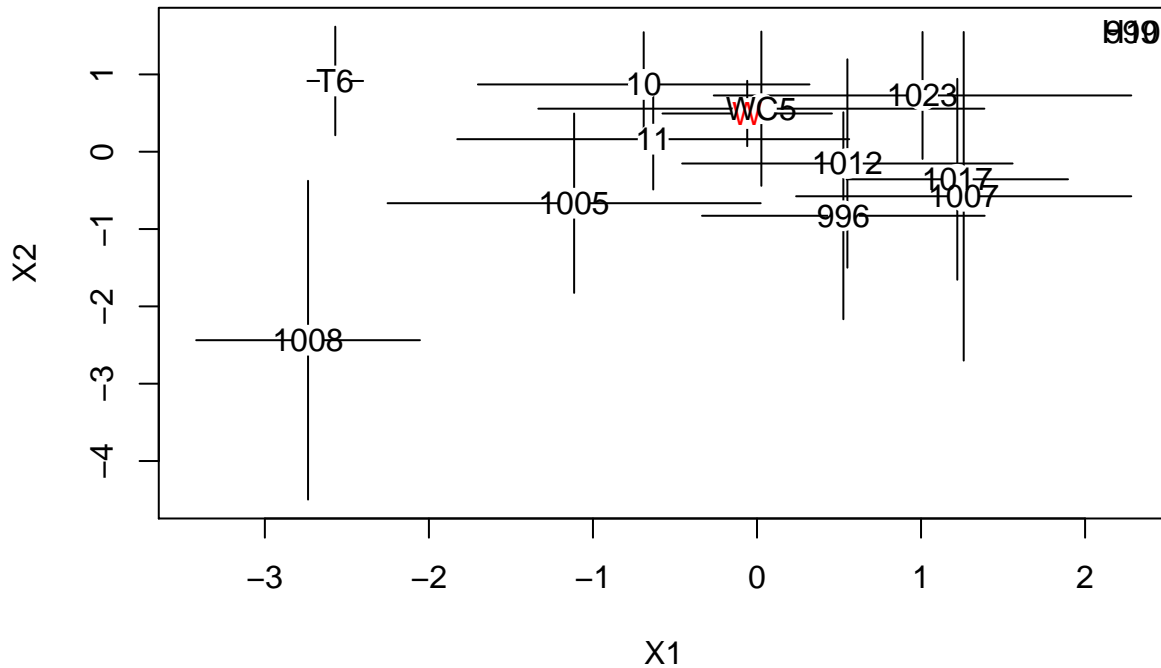
```
## plot(cv.onc, col = "grey30")
## legend("topleft", legend = "A")
```

```
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?

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



## 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()
```

## Loading and pre-processing data

```

## Check for supporting packages
pkg.list <- c("MuMIn", "lme4", "RLRsim", "vegan", "ecodist", "bipartite", "RColorBrewer", "enaR", "devtools")
# Install packages that are not installed
if (any(!(pkg.list %in% installed.packages()[, 1]))){
  sapply(pkg.list[which(!(pkg.list %in% installed.packages()[, 1]))],
    install.packages, dependencies = TRUE, repos='http://cran.us.r-project.org')
}

# Check non-CRAN packages
if (!("ComGenR" %in% installed.packages()[, 1])){
  devtools::install_github("CommunityGeneticsAnalyses/ComGenR")
}
if (!("coNet" %in% installed.packages()[, 1])){
  devtools::install_github("CommunityGeneticsAnalyses/coNet")
}

# Load libraries
sapply(c(pkg.list, "ComGenR", "coNet"), library, quietly = TRUE, character.only = TRUE)

#### Loading data objects
## onc.com = "community" occurrences summed across all cells for each tree
## onc.q = occurrence matrices separated out for each tree
## onc.geno = genotypes
## prb.onc = percent rough bark (averaged between the upper and lower)

#### Data notes:
## Trees were removed from the analysis genotype RL6 and N1.31
## No physcioids
## Lecanoras merged

# Loading data
xgal.size <- gdata::read.xls("../data/lichen_networks/ONC_Xgal_SizeData_May2011.xls")
garden.data <- read.csv("../data/lichen_networks/LCO_data_ONC_PIT.csv")
# remove 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'])
unique(onc$Tree[onc$Year == '2011']) %in% unique(onc$Tree[onc$Year == '2010'])
# 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)])
# 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:-4, -(ncol(xgal.size) - 1):ncol(xgal.size)]
xgs.cols <- xgal.size[4, -(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/lichen_networks/ONC_raw_roughness.csv')
# Isolate roughness
rough <- rough[, 1:5]
# Isolate north quadrats
rough <- rough[grep("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 ses values
load('../data/lichen_networks/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!')}
#RFLP distance values from Zink from Martinsen
rflp.d <- readLines('../data/acn/rflp_queller_goodnight.txt')
rflp.d <- lapply(rflp.d, strsplit,split='\t')
rflp.d <- lapply(rflp.d, function(x) x[[1]])
rflp.d[[61]] <- c(rflp.d[[61]], "")
rflp.d <- do.call(rbind, rflp.d)
rflp.n <- rflp.d[1, -1]
rflp.d <- rflp.d[-1, -1]
diag(rflp.d) <- 1
rflp.d <- matrix(as.numeric(rflp.d),nrow=nrow(rflp.d))
rownames(rflp.d) <- colnames(rflp.d) <- rflp.n
rflp.d <- rflp.d[rownames(rflp.d) %in% unique(onc.geno),
               colnames(rflp.d) %in% unique(onc.geno)]
rflp.d <- rflp.d[match(unique(onc.geno), rownames(rflp.d)),
               match(unique(onc.geno),rownames(rflp.d))]]
if (!(all(rownames(rflp.d) == unique(onc.geno)))){
  print('Holy crap, rflp.d names match error')
}

## Duplicate by genotype? Would this work to make the RFLPs values replicated?
# Coerce to distance matrices
rflp.d <- as.dist(rflp.d)
# Lichen Network Models
# onc
cn.onc <- lapply(split(onc[, -1:-6], onc[, "Tree"]), coNet,
               ci.p = 95, cond = TRUE)
cn.sign.onc <- lapply(split(onc[, -1:-6], onc[, "Tree"]), coNet,
                   ci.p = 95)
cn.d.onc <- netDist(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 <- netDist(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]] <- netMean(cn.onc[onc.geno == unique(onc.geno)[i]])
}
names(cn.mu.onc) <- unique(onc.geno)
cn.mu.d.onc <- netDist(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){
    ## Networks with modules = 2 9 14 19 20 25 27 28 30 31 42 44 54 57
    mod.tmp <- computeModules(cn.onc[[i]])
    cn.mod.onc[i, 1] <- slot(mod.tmp, "likelihood") ## module likelihood
    cn.mod.onc[i, 2] <- nrow(slot(mod.tmp, "modules")) - 1 ## number modules
  }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, sna::degree, rescale = FALSE)
cen.spp <- do.call(rbind, cen.spp)
colnames(cen.spp) <- colnames(cn.onc[[1]])
# Community data
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)))
# Vectors for network similarity
## ns.vec.onc <- envfit(ord, data.frame(onc.ns[, c("L", "Cen")], R = onc.rough, Cov = ptc.onc))

## Creating distance matrices that match rflp
## this is for the "mean" distance matrices
cn.mu.d <- as.matrix(cn.mu.d.onc)
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(c(all(rownames(as.matrix(rflp.d)) == rownames(as.matrix(cn.mu.d.onc))),
            all(rownames(as.matrix(rflp.d)) == rownames(as.matrix(prb.mu.d))),
            all(rownames(as.matrix(rflp.d)) == rownames(as.matrix(onc.com.mu.d))))))){
  warning("Warning: distance matrices are not aligned.")
}else{
  print("Distance matrices good to go!")
}

# Bipartite analysis
nperm <- 20
if (!(file.exists("../data/lichen_networks/nest_rel_onc.rda"))){
  nest.onc <- nestedness(onc.com.rel[, colnames(onc.com.rel) != "ds"], n.nulls = 999)
  dput(nest.onc, "../data/lichen_networks/nest_rel_onc.rda")
}else{
  nest.onc <- dget("../data/lichen_networks/nest_rel_onc.rda")
}
if (!(file.exists("../data/lichen_networks/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/lichen_networks/mod_list_onc.rda")
  write.csv(slot(obs.mod.onc, "likelihood"),
            "../data/lichen_networks/obs_mod_onc.csv",
            row.names = FALSE)
  write.csv(nul.mod.onc,
            "../data/lichen_networks/null_mod_onc.csv",
            row.names = FALSE)
}else{
  obs.mod.onc <- read.csv("../data/lichen_networks/obs_mod_onc.csv")[1]
  nul.mod.onc <- read.csv("../data/lichen_networks/null_mod_onc.csv")[1,1]
  z.mod.onc <- (obs.mod.onc - mean(nul.mod.onc)) / sd(nul.mod.onc)
  mods.onc <- dget("../data/lichen_networks/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/lichen_networks/onc_nmds.csv")){
  nms.info.onc <- capture.output(nms.onc <- nmds.min(nmds(
    vegdist(onc.com.rel), 2, 2)))
  write.csv(nms.onc, "../data/lichen_networks/onc_nmds.csv", row.names = FALSE)
  write.table(nms.info.onc,
    "../data/lichen_networks/onc_nmds_info.txt",
    col.names = FALSE, row.names = FALSE)
}else{nms.onc <- read.csv("../data/lichen_networks/onc_nmds.csv")}

                                # Network ordination
if (!(file.exists("../data/lichen_networks/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/lichen_networks/conet_nmds.csv", row.names = FALSE)
  write.table(cn.nmds.stats.onc,
    "../data/lichen_networks/conet_nmds_info.txt",
    col.names = FALSE, row.names = FALSE)
}else{cn.nms.onc <- read.csv("../data/lichen_networks/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))

                                #genotype means
omu <- apply(onc.com[,colnames(onc.com) != 'ds'], 2,
  function(x,g) tapply(x,g,mean),g=onc.geno)
oms <- tapply(onc.ses, onc.geno, mean)
oms.d <- dist(oms[match(rownames(as.matrix(rflp.d)),names(oms))])

                                #bark roughness means
oprbbmu <- tapply(onc.rough,onc.geno,mean)
oprbbmu <- oprbbmu[match(rownames(as.matrix(rflp.d)),names(oprbbmu))]

                                #get araujo coordinates
coord <- read.csv("../data/lichen_networks/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/lichen_networks/ONC_phytochem_NC.csv")
onc.tan <- read.csv("../data/lichen_networks/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

# remove trees not in onc.dat
# makes missing samples NA
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"])
# collect into a single df
onc.dat <- data.frame(tree.id, PC = ptc.onc, SR = spr.onc,
  geno = factor(onc.geno), tree = tree,
  BR = onc.rough, onc.ns[, c("L", "Cen")])
# 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")
# Figure ordinations
# Communities
if (file.exists("../data/lichen_networks/nms_com_onc.rda")){
  nms.com <- dget(file = "../data/lichen_networks/nms_com_onc.rda")
}else{
  set.seed(12345)
  nms.com <- nmms(vegdist(onc.com.rel), 2, 3)
  dput(nms.com, file = "../data/lichen_networks/nms_com_onc.rda")
}
# Networks
if (file.exists("../data/lichen_networks/nms_cn_onc.rda")){
  nms.cn <- dget(file = "../data/lichen_networks/nms_cn_onc.rda")
}else{
  set.seed(12345)
  nms.cn <- nmms(cn.d.onc, 1, 2)
  dput(nms.cn, file = "../data/lichen_networks/nms_cn_onc.rda")
}
ord.com <- nmms.min(nms.com, 3)
ord.cn <- nmms.min(nms.cn, 2)
# Vectors for plotting
# Composition
vec.env <- onc.dat[, c("BR", "PC", "SR")]
colnames(vec.env) <- c("BR", "PC", "SR")
vec.com.12 <- envfit(ord.com, env = vec.env, perm = 10000,
  choices = c(1,2))

```

```

                                # Network similarity
vec.cn <- envfit(ord.cn, env = vec.env, perm = 10000,
                choices = c(1,2))

                                # onc
if (!("mod_obsval_onc.csv" %in% dir("../data/lichen_networks"))){
  mod.onc <- slot(bipartite::computeModules(rel(onc.com[, -ncol(onc.com)]),
                                           deep = FALSE),
                 "likelihood")
  write.csv(mod.onc, file = "../data/mod_obsval_onc.csv", row.names = FALSE)
}else{
  mod.onc <- read.csv(file = "../data/lichen_networks/mod_obsval_onc.csv")[, 1]
}
if (!("mod_simvals_onc.csv" %in% dir("../data/lichen_networks"))){
  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/lichen_networks/mod_simvals_onc.csv", row.names = FALSE)
# nest.onc <- bipartite::nestedness(onc.com.rel)
}else{
  mods.onc.sweb <- read.csv(file = "../data/lichen_networks/mod_simvals_onc.csv")[, 1]
}

                                # pit
if (!("mod_obsval_pit.csv" %in% dir("../data/lichen_networks"))){
  mod.pit <- slot(bipartite::computeModules(rel(pit.com), deep = FALSE),
                 "likelihood")
  write.csv(mod.pit, file = "../data/lichen_networks/mod_obsval_pit.csv", row.names = FALSE)
}else{
  mod.pit <- read.csv(file = "../data/lichen_networks/mod_obsval_pit.csv")[, 1]
}
if (!("mod_simvals_pit.csv" %in% dir("../data/lichen_networks"))){
  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/lichen_networks/mod_simvals_pit.csv", row.names = FALSE)
# nest.pit <- bipartite::nestedness(pit.com.rel)
}else{
  mods.pit.sweb <- read.csv(file = "../data/lichen_networks/mod_simvals_pit.csv")[, 1]
}

### Wild data
###
###
x <- read.csv('../data/lichen_networks/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$tree!='gnu.44',]
#remove ll.6, weird tree with super smooth bark
x <- x[x$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)))
#no phycioids!
#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)!='phycioid']
#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/lichen_networks/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$Quad.Loc),function(x) unlist(strsplit(x,split='_'))))
env$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)
#match observations
all(env.tid[match(names(x.q),env.tid)] == names(x.q))
#delimit to co-occurrence dataset 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)
prb.wild <- tapply(env$Pct.Roughness,env.split,mean) #percent rough bark
#age
age <- read.csv('../data/lichen_networks/UintaMaster_LichenHeritNL_FallSpring_2012_ForLau.csv')
dbh <- age$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$tree.id == 'gnu.19']
new <- data.frame(dbh=seq(min(dbh),max(dbh),by=0.1))
age.final <- na.omit(age.final)
pred.age <- predict(lm(age.final~dbh,data=age),new)
plot(pred.age~new[,1])
gnu19.age <- as.numeric(pred.age[new[,1] == gnu19.dbh])
#
tree.age <- age[match(names(prb.wild),age[,1]),2]
tree.age[is.na(tree.age)] <- gnu19.age
names(tree.age) <- age[match(names(prb.wild),age[,1]),1]
age <- tree.age
# networks
cn.wild <- lapply(wild.q, coNet)
cn.mu.wild <- netMean(cn.wild)
cn.d.wild <- netDist(cn.wild, method = 'bc')
#co-occurrence patterns
wco <- do.call(rbind,lapply(wild.q,function(x,t) apply(CoCo(x,type=t),2,sum),t='pos'))
wch <- do.call(rbind,lapply(wild.q,function(x,t) apply(CoCo(x,type=t),2,sum),t='neg'))
#get ses values
#"z" "means" "pval" "simulated" "method" "statistic" "alternati
#ws <- lapply(wild.q,function(x) oecosimu(x,cs,method='r1',burn
#uses <- unlist(lapply(ws,function(x) x$oecosimu[[1]]))
#wsmu <- unlist(lapply(ws,function(x) x$oecosimu[[2]]))
#wsp <- unlist(lapply(ws,function(x) x$oecosimu[[3]]))
#wsim <- do.call(rbind,lapply(ws,function(x) x$oecosimu[[4]]))
#rownames(wsim) <- names(wild.q)
#ws <- cbind(uses,wsmu,wsp)
#write.csv(ws,file='../data/wild_ses_21mar2014.csv')

## Araujo Coordinate Values
coord <- read.csv('../data/lichen_networks/lcn_coord_onc.csv')
rownames(coord) <- coord[,1]
coord <- coord[,-1]

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

```

```
    mods.wild.sweb <- read.csv(file = "../data/lichen_networks/mod_simvals_wild.csv")[, 1]
  }

  ###Rename data objects for simplicity
  ws <- read.csv('../data/lichen_networks/wild_ses_21mar2014.csv')
```