Rock Lichen data from Sunset Crater collected by Rikke Naesborg and Richard Michalet

Rikke, Matt, Cameron, Richard

Analysis Summary

- Dead trees and non-lichen species were removed from lichen community analyses.
- Lichen communities were adequately sampled, based on species accumulation curves, with moth resistant trees accumulating slightly more lichen species.
- Lichen communities (abundance, richness, diversity, composition) were significantly, generally negatively, affected by moth susceptibility.
- Several tree variables, including light availability, leaf litter abundance and rock abundance, were impacted by moth susceptibility.
- Analysis of causal pathways supported an indirect link between moth susceptibility and impacts on lichen communities via decreasing rock (i.e. habitat) availability through increased leaf abscission and accumulation on rocks under trees.
- These results support a genetically based link between intraspecific variation in susceptibility to an insect herbivore and community dynamics in an arid ecosystem.
- Given the possible impacts of cliimate change on this system, this study supports the conclusion that community and ecosystem impacts need to be considered in an evolutionary context.

```r
# 0. Supporting functions and libraries
## Support functions

dif <- function(x){
    out=x[1]
    for (i in 2:length(x)){
        out=out-x[i]
    }
    return(out)
}

## Libraries
my.libs <- c("vegan", "ecodist")
if (any(!(my.libs %in% installed.packages()[, 1]))){
    sapply(my.libs[!(my.libs %in% installed.packages()[, 1])],
           install.packages)
}else{}
sapply(my.libs, require, character.only = TRUE)
```

Load Data

The following are variable descriptions (Variable, Type, Range, Definition):

- Moth,categorical,0 or 1,Was the tree susceptible (0) or resistant (1) to moth attack?
- Live/Dead,categorical,0 or 1,Was the tree dead (0) or alive (1)?
- Litter %,continuous,0 to 100,Percent cover inside quadrat
- Rocks > 3cm? %,continuous,0 to 100,Percent cover of rocks > 3cm? inside quadrat
- Rocks < 3cm? %,continuous,0 to 100,Percent cover of rocks < 3cm? inside quadrat

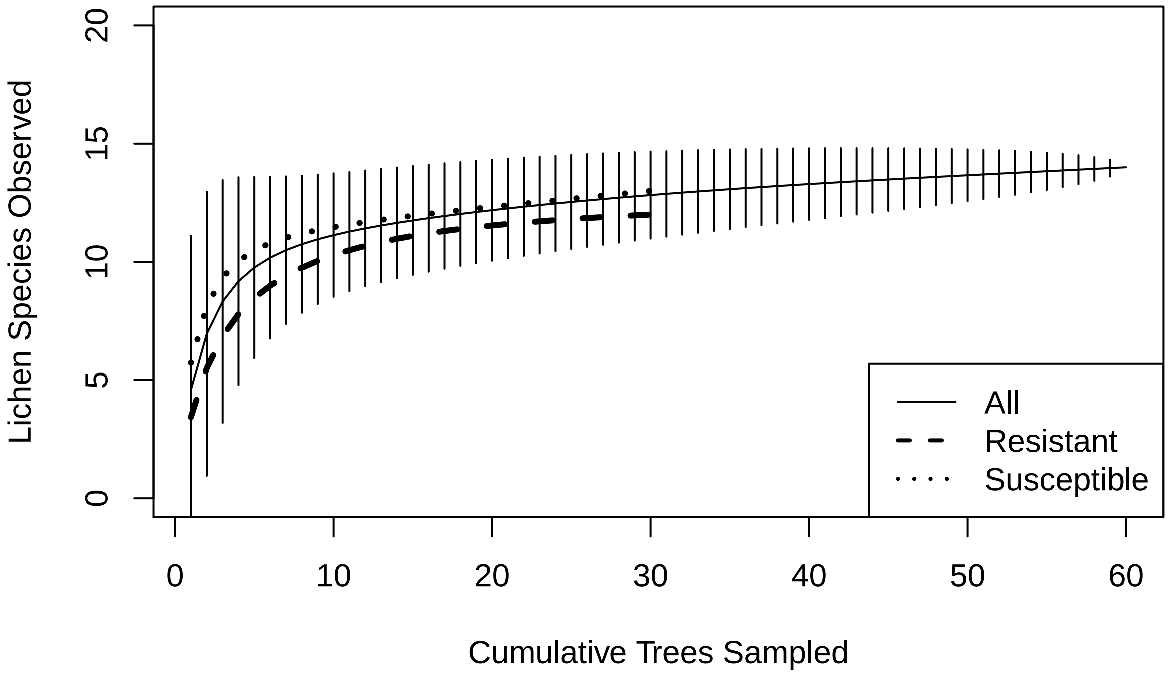
- Shrubs %,continuous,0 to 100,Percent cover of shrubs inside quadrat
- Grass %,continuous,0 to 100,Percent cover of grass inside quadrat
- Branches %,continuous,0 to 100,Percent cover of branches on ground inside quadrat
- Distance,continuous,0 to 100,“Distance from main trunk, converted to percent of crown radius at that azimuth”
- Azimuth,continuous,0 to 360,Compass direction from main trunk
- Slope,continuous,0 to 90,Topographical steepness
- Aspect,continuous,0 to 360,Compass direction of slope
- Light,continuous,,Amount of light available to epiliths

```
## Data are in ../data/scrl
l.dat <- read.csv("../data/scrl/sp & env combined.csv")

## Summary of data
summary(l.dat)

## remove dead trees
l.dat <- l.dat[l.dat[, "Live.Dead"] != 0, ]

## Lichen species list
spp.l <- c("Acacon", "Acasup", "Acaobp", "Sterile.sp", "Brown.cr",
"Lobalp", "Canros", "Calare", "Phydub", "Rhichr", "Xanlin", "Xanpli",
"Xanele", "GrBr.cr", "Gray.cr")
spp.moss <- c("Synrur", "Cerpur.Bryarg")

## Create a community matrix
com <- l.dat[, colnames(l.dat) %in% spp.l]
com.moss <- l.dat[, colnames(l.dat) %in% spp.moss]

## Add the tree labels to the rownames
rownames(com) <- paste(l.dat[, "Moth"], l.dat[, "Tree.pairs"], sep = "_")
rownames(com.moss) <- paste(l.dat[, "Moth"], l.dat[, "Tree.pairs"], sep = "_")
rownames(l.dat) <- paste(l.dat[, "Moth"], l.dat[, "Tree.pairs"], sep = "_")
```

Species accumulation

Are the communities on each tree type adequately sampled?

```
spa.all <- specaccum(com)
spa.res <- specaccum(com[l.dat[, "Moth"] == 0, ])
spa.sus <- specaccum(com[l.dat[, "Moth"] == 1, ])

plot(spa.all,
     ylim = c(0, 20),
     xlab = "Cumulative Trees Sampled",
     ylab = "Lichen Species Observed")
lines(spa.res$sites, spa.res$richness,
      ylim = c(0, 20), lty = 2, lwd = 3)
lines(spa.sus$sites, spa.sus$richness,
      ylim = c(0, 20), lty = 3, lwd = 3)
legend("bottomright",
       legend = c("All", "Resistant", "Susceptible"),
       lty = c(1, 2, 3), lwd = c(1, 2, 2))
```



Community Analyses

The following is an analysis of the univariate community metrics, abundance, richness and diversity (Shannon's index). These three metrics were strongly intercorrelated and all responded to the environmental effects of moth infection. Litter cover, light and rock abundance were all also responsive to moth infection. Other variables, such as branches, grasses and mosses were not impacted by moth. Differences are analyzed with univariate t-tests in order to account for the paired structure of the data. Also presented here are Shapiro-Wilks tests for homogeneity of variance, which as assumption of the t-test. These effects should be shown using dotplots with mean differences as dots and confidence intervals as horizontal lines.

```
## Calculate metrics
abun <- apply(com, 1, sum)
rich <- apply(com, 1, function(x) sum(sign(x)))
shan <- apply(com, 1, diversity, index = "shannon")
## Examine intercorrelations
cor.test(abun, rich, method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  abun and rich
## z = 8.3541, p-value < 2.2e-16
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.7845321
```

```
cor.test(abun, shan, method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  abun and shan
## z = 5.9531, p-value = 2.63e-09
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5497907
```

```r
cor.test(rich, shan, method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  rich and shan
## z = 7.8403, p-value = 4.496e-15
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.7471965
```

```r
## Test for moth effects via a one-way test on the
## differences between susceptible and resistance trees.
## diff = Susceptible - Resistant
## If diff is less than 0, susceptibility had a negative effect
## If diff is greater than 0, susceptibility had a positive effect

### Test for violation of normality of differences
shapiro.test(tapply(abun , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(abun, l.dat[, "Tree.pairs"], diff)
## W = 0.95143, p-value = 0.1847
```

```r
shapiro.test(tapply(rich , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(rich, l.dat[, "Tree.pairs"], diff)
## W = 0.9541, p-value = 0.2175
```

```r
shapiro.test(tapply(shan , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(shan, l.dat[, "Tree.pairs"], diff)
## W = 0.95291, p-value = 0.2021
```

```r
shapiro.test(tapply(l.dat[, "Litter.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff)
## W = 0.94289, p-value = 0.1088
```

```
shapiro.test(tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Light...average"], l.dat[, "Tree.pairs"], diff)
## W = 0.96945, p-value = 0.5242
```

```
shapiro.test(tapply(l.dat[, "Light...N"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Light...N"], l.dat[, "Tree.pairs"], diff)
## W = 0.98758, p-value = 0.9727
```

```
shapiro.test(tapply(l.dat[, "Light...S"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Light...S"], l.dat[, "Tree.pairs"], diff)
## W = 0.98356, p-value = 0.9102
```

```
shapiro.test(tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Big.rocks.."], l.dat[, "Tree.pairs"], diff)
## W = 0.94775, p-value = 0.1471
```

```
### Too small for habitat
shapiro.test(tapply(l.dat[, "Small.rocks.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Small.rocks.."], l.dat[, "Tree.pairs"], diff)
## W = 0.79098, p-value = 4.509e-05
```

```
### Too few observations
shapiro.test(tapply(l.dat[, "Synrur"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Synrur"], l.dat[, "Tree.pairs"], diff)
## W = 0.3766, p-value = 3.307e-10
```

```
shapiro.test(tapply(l.dat[, "Branches.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tapply(l.dat[, "Branches.."], l.dat[, "Tree.pairs"], diff)
## W = 0.17962, p-value = 7.766e-12
```

```r
shapiro.test(tapply(l.dat[, "Grass.."] , l.dat[, "Tree.pairs"], diff))
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  tapply(l.dat[, "Grass.."], l.dat[, "Tree.pairs"], diff)
## W = 0.17962, p-value = 7.766e-12
```

```r
shapiro.test(tapply(l.dat[, "Cerpur.Bryarg"] , l.dat[, "Tree.pairs"], diff))
```

```
## 
##  Shapiro-Wilk normality test
## 
## data:  tapply(l.dat[, "Cerpur.Bryarg"], l.dat[, "Tree.pairs"], diff)
## W = 0.2428, p-value = 2.398e-11
```

```r
### Test for effect of moth
t.test(tapply(abun, l.dat[, "Tree.pairs"], diff))
```

```
## 
##  One Sample t-test
## 
## data:  tapply(abun, l.dat[, "Tree.pairs"], diff)
## t = -2.3568, df = 29, p-value = 0.0254
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -2.8925956 -0.2047377
## sample estimates:
## mean of x 
## -1.548667
```

```r
t.test(tapply(rich, l.dat[, "Tree.pairs"], diff))
```

```
## 
##  One Sample t-test
## 
## data:  tapply(rich, l.dat[, "Tree.pairs"], diff)
## t = -2.8358, df = 29, p-value = 0.008247
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -3.9588011 -0.6411989
## sample estimates:
## mean of x 
##      -2.3
```

```r
t.test(tapply(shan, l.dat[, "Tree.pairs"], diff))
```

```
## 
##  One Sample t-test
## 
## data:  tapply(shan, l.dat[, "Tree.pairs"], diff)
## t = -2.4328, df = 29, p-value = 0.02138
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.78351424 -0.06781091
## sample estimates:
##  mean of x
## -0.4256626
```

```r
t.test(tapply(l.dat[, "Litter.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff)
## t = 2.8665, df = 29, p-value = 0.00765
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##   4.317792 25.822208
## sample estimates:
## mean of x
##     15.07
```

```r
t.test(tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Light...average"], l.dat[, "Tree.pairs"], diff)
## t = -9.2728, df = 29, p-value = 3.557e-10
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -18.47119 -11.79547
## sample estimates:
## mean of x
## -15.13333
```

```r
t.test(tapply(l.dat[, "Light...N"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Light...N"], l.dat[, "Tree.pairs"], diff)
## t = -8.0191, df = 29, p-value = 7.634e-09
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -20.05142 -11.90191
## sample estimates:
## mean of x
## -15.97667
```

```r
t.test(tapply(l.dat[, "Light...S"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Light...S"], l.dat[, "Tree.pairs"], diff)
## t = -7.5187, df = 29, p-value = 2.748e-08
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -18.17717 -10.40283
## sample estimates:
## mean of x
##    -14.29
```

```r
t.test(tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Big.rocks.."], l.dat[, "Tree.pairs"], diff)
## t = -2.4617, df = 29, p-value = 0.02001
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -17.728936  -1.638397
## sample estimates:
## mean of x
## -9.683667
```

```r
                                        # Ignore these results
t.test(tapply(l.dat[, "Small.rocks.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Small.rocks.."], l.dat[, "Tree.pairs"], diff)
## t = -2.0792, df = 29, p-value = 0.04655
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -9.86878788 -0.08121212
## sample estimates:
## mean of x
##    -4.975
```

```r
t.test(tapply(l.dat[, "Branches.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Branches.."], l.dat[, "Tree.pairs"], diff)
## t = 1, df = 29, p-value = 0.3256
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1484226  0.4324226
## sample estimates:
## mean of x
##     0.142
```

```r
t.test(tapply(l.dat[, "Grass.."] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Grass.."], l.dat[, "Tree.pairs"], diff)
## t = -1, df = 29, p-value = 0.3256
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.15023133  0.05156466
## sample estimates:
##   mean of x
## -0.04933333
```

```
t.test(tapply(l.dat[, "Synrur"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Synrur"], l.dat[, "Tree.pairs"], diff)
## t = 0.36285, df = 29, p-value = 0.7194
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1020054  0.1460054
## sample estimates:
## mean of x
##     0.022
```

```
t.test(tapply(l.dat[, "Cerpur.Bryarg"] , l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(l.dat[, "Cerpur.Bryarg"], l.dat[, "Tree.pairs"], diff)
## t = -1.2357, df = 29, p-value = 0.2265
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.04602247  0.01135580
## sample estimates:
##   mean of x
## -0.01733333
```

Lichen communities as a whole responded to the effects of moth infection. Using PerMANOVA, a permutational analysis of variance, with the tree pairs as blocks in the model, we observed a significant effect of moth susceptibility on lichen community composition. This should be shown using an NMDS ordination plot.

```
### Composition analysis
com.ds <- cbind(com, ds = rep(0.01, nrow(com)))
adonis2(com.ds ~ Moth, data = l.dat, strata = factor(l.dat$Tree.pairs), perm = 9999)
```

```
## Permutation test for adonis under reduced model
## Terms added sequentially (first to last)
## Permutation: free
## Number of permutations: 9999
##
## adonis2(formula = com.ds ~ Moth, data = l.dat, permutations = 9999, strata = factor(l.dat$Tree.pairs
##          Df SumOfSqs      R2      F Pr(>F)
## Moth      1   0.8849 0.04412 2.6772 0.0218 *
## Residual 58  19.1700 0.95588
## Total    59  20.0548 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Indicator species

Although susceptibility to moth infection generally had a negative impact on lichen, several species showed significant responses. Here, because of the paired design, we analyze the response of the differences using a one-way t-test on the differences of each species. I don't think we need to plot this, but perhaps the results could be shown as a table.

```
ind.spp <- list()
ind.spp$p.value[1] <- t.test(tapply(com[, "Acacon"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[2] <- t.test(tapply(com[, "Acasup"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[3] <- t.test(tapply(com[, "Canros"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[4] <- t.test(tapply(com[, "Lobalp"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[5] <- t.test(tapply(com[, "Phydub"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[6] <- t.test(tapply(com[, "Acaobp"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[7] <- t.test(tapply(com[, "Sterile.sp"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[8] <- t.test(tapply(com[, "Calare"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[9] <- t.test(tapply(com[, "Rhichr"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[10] <- t.test(tapply(com[, "Xanlin"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[11] <- t.test(tapply(com[, "Xanpli"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[12] <- t.test(tapply(com[, "Xanele"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$p.value[13] <- t.test(tapply(com[, "GrBr.cr"], l.dat[, "Tree.pairs"], diff))$p.value
ind.spp$t[1] <- t.test(tapply(com[, "Acacon"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[2] <- t.test(tapply(com[, "Acasup"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[3] <- t.test(tapply(com[, "Canros"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[4] <- t.test(tapply(com[, "Lobalp"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[5] <- t.test(tapply(com[, "Phydub"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[6] <- t.test(tapply(com[, "Acaobp"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[7] <- t.test(tapply(com[, "Sterile.sp"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[8] <- t.test(tapply(com[, "Calare"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[9] <- t.test(tapply(com[, "Rhichr"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[10] <- t.test(tapply(com[, "Xanlin"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[11] <- t.test(tapply(com[, "Xanpli"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[12] <- t.test(tapply(com[, "Xanele"], l.dat[, "Tree.pairs"], diff))$statistic
ind.spp$t[13] <- t.test(tapply(com[, "GrBr.cr"], l.dat[, "Tree.pairs"], diff))$statistic
## This species had zero abundance, perhaps it was only observed under dead trees
t.test(tapply(com[, "Brown.cr"], l.dat[, "Tree.pairs"], diff))
```

```
##
##  One Sample t-test
##
## data:  tapply(com[, "Brown.cr"], l.dat[, "Tree.pairs"], diff)
## t = NaN, df = 29, p-value = NA
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  NaN NaN
## sample estimates:
## mean of x
##         0
```

```
## Assign names to vectors
names(ind.spp$p.value) <- c("Acacon", "Acasup", "Canros", "Lobalp", "Phydub",
"Acaobp", "Sterile.sp", "Calare", "Rhichr", "Xanlin", "Xanpli",
"Xanele", "GrBr.cr")
names(ind.spp$t) <- c("Acacon", "Acasup", "Canros", "Lobalp", "Phydub",
"Acaobp", "Sterile.sp", "Calare", "Rhichr", "Xanlin", "Xanpli",
"Xanele", "GrBr.cr")
ind.spp$p.value <- p.adjust(ind.spp$p.value, method = "fdr")
do.call(cbind, ind.spp)
```

```
##               p.value          t
## Acacon     0.0129109 -3.3776290
## Acasup     0.0129109 -3.2420906
## Canros     0.0129109 -3.5818840
## Lobalp     0.1638226 -2.0413612
## Phydub     0.1674381 -1.9226188
## Acaobp     0.3847787 -1.0747324
## Sterile.sp 0.3847787 -1.0000000
## Calare     0.2082316 -1.6076070
## Rhichr     0.2082316 -1.5803288
## Xanlin     0.5872370 -0.6169756
## Xanpli     0.7968379 -0.2598207
## Xanele     0.2082316 -1.5662320
## GrBr.cr    0.3847787  1.0000000
```

Exploring causal pathways

I suggest that we develop these into a structural equation model. Here, I've analyzed several "causal pathways" in parts using correlation tests of the differences (susceptible - resistant) for each variable. Here are two hypothesized causal pathways, where lichen refers to the lichen community abundance, richness, diversity and composition:

- moth decreases light which decreases lichen
- moth increases litter which covers/decreases rocks which support/increase lichen

Only the litter pathway is supported. Although light is impacted by moth susceptibility, it isn't correlated with lichen. Litter, however, is positively affected by the moth infection. In addition, litter is negatively correlated with rock abundance and rock abundance is positively correlated with lichen. This is also supported by the indirect pathway between moth and lichen, which are negatively correlated, as would be expected by the product of the signs along the pathway (i.e. positive * negative * positive = negative).

```
### Test for correlations among effects
cor.test(tapply(l.dat[, "Light...average"], l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(l.dat[, "Light...average"], l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Big.rocks
## T = 229, p-value = 0.6972
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## 0.05287356
```

```
cor.test(tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Big.rocks.."], l
## T = 43, p-value = 3.161e-13
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.8022989
```

```r
cor.test(tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Light...average"
## T = 193, p-value = 0.395
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.1126437
```

```r
## Abun
cor.test(tapply(abun, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(abun, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(abun, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Light...average"], l.dat[, "Tre
## z = 0.92788, p-value = 0.3535
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.1196779
```

```r
cor.test(tapply(abun, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Litter.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(abun, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(abun, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs
## z = -2.9978, p-value = 0.00272
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.3866516
```

```r
cor.test(tapply(abun, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(abun, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(abun, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Big.rocks.."], l.dat[, "Tree.pa
## z = 3.319, p-value = 0.0009035
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4280785
```

```
## rich
cor.test(tapply(rich, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(rich, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(rich, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Light...average"], l.dat[, "Tre
## z = 1.3285, p-value = 0.184
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.1756536
```

```
cor.test(tapply(rich, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Litter.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(rich, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(rich, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs
## z = -4.0214, p-value = 5.785e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.5317081
```

```
cor.test(tapply(rich, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(rich, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties
```

```
##
##  Kendall's rank correlation tau
##
## data:  tapply(rich, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Big.rocks.."], l.dat[, "Tree.pa
## z = 4.0573, p-value = 4.964e-05
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.5364555
```

```
## shan
cor.test(tapply(shan, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Light...average"] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(shan, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties

##
##  Kendall's rank correlation tau
##
## data:  tapply(shan, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Light...average"], l.dat[, "Tre
## z = 1.5352, p-value = 0.1247
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.1983864
```

```
cor.test(tapply(shan, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Litter.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(shan, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties

##
##  Kendall's rank correlation tau
##
## data:  tapply(shan, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Litter.."], l.dat[, "Tree.pairs
## z = -3.4632, p-value = 0.0005338
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##        tau
## -0.4475229
```

```
cor.test(tapply(shan, l.dat[, "Tree.pairs"], diff),
        tapply(l.dat[, "Big.rocks.."] , l.dat[, "Tree.pairs"], diff), method = "k")
```

```
## Warning in cor.test.default(tapply(shan, l.dat[, "Tree.pairs"], diff),
## tapply(l.dat[, : Cannot compute exact p-value with ties

##
##  Kendall's rank correlation tau
##
## data:  tapply(shan, l.dat[, "Tree.pairs"], diff) and tapply(l.dat[, "Big.rocks.."], l.dat[, "Tree.pa
## z = 3.3918, p-value = 0.0006944
## alternative hypothesis: true tau is not equal to 0
## sample estimates:
##       tau
## 0.4382956
```