

IllumFUN_Q2bc: SEMs

Marissa Lee

12/16/2019

Q2. How are differences in fungal composition across the landscape explained by environmental variables?

Table of contents

0. Load data and pre-process ASV matrix

See IllumFUN_Q1.Rmd

A. Determine which environmental variables to include in path analysis

See IllumFUN_Q2a.Rmd

B. SEM using DPCoA1 score to represent community

1. Plot tissue-specific DPCoAs
2. Construct tissue-specific SEMs

C. Investigate SEM results with bivariate plots

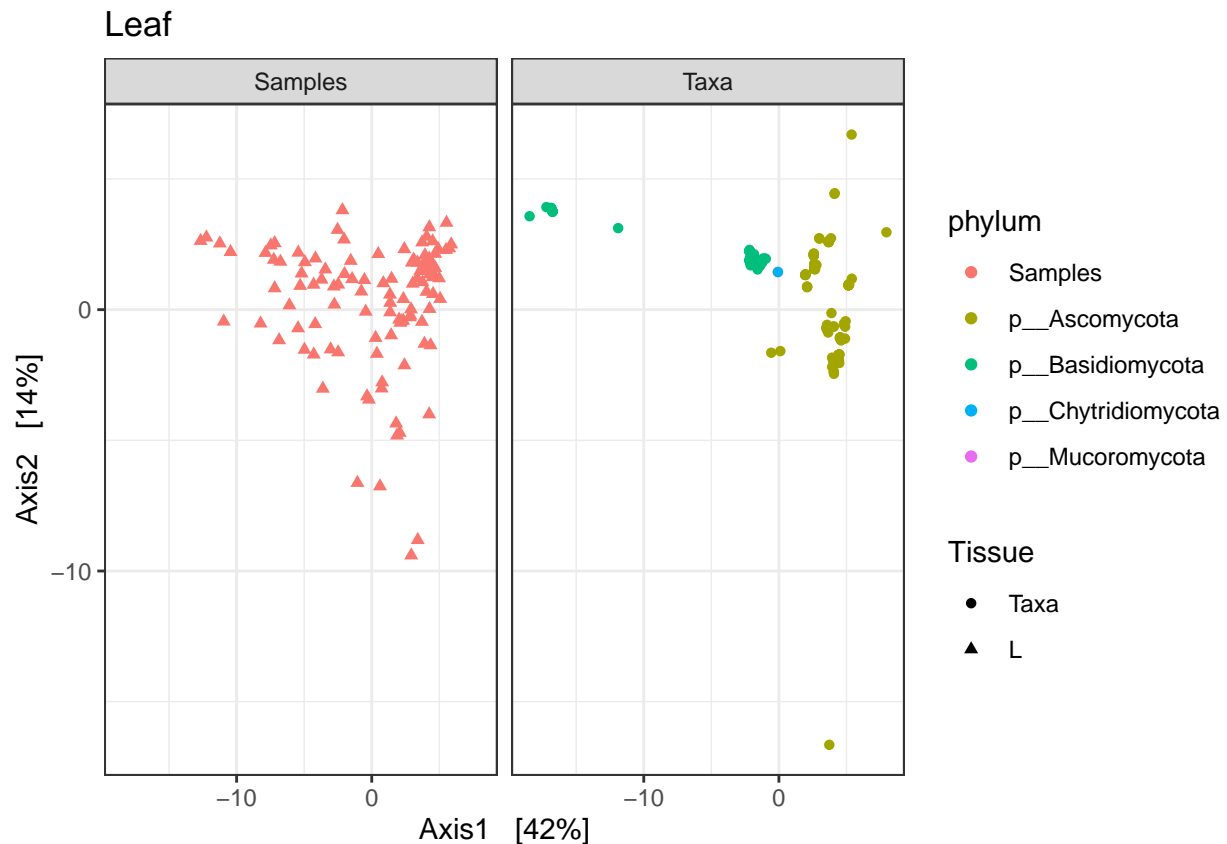
1. Set up functions, levels, colors
2. Make plotting dataframes – samples
3. Make plotting dataframes – ASVs
4. Plot – direct effects
5. Plot – indirect effects

Load packages, functions, paths

Custom functions

B. SEM using DPCoA1 score to represent fungal communities

1. Plot Tissue-specific DPCoAs (example for leaf taxa)



2. Construct Tissue-specific SEMs

Leaf (example) Full model:

```
# set up sem  
library(lavaan)
```

```
## This is lavaan 0.6-6
```

```
## lavaan is BETA software! Please report any bugs.
```

```
library(semPlot)
```

```
## Registered S3 methods overwritten by 'lme4':  
##   method                      from  
##   cooks.distance.influence.merMod car  
##   influence.merMod             car  
##   dfbeta.influence.merMod      car  
##   dfbetas.influence.merMod     car
```

```

## Registered S3 methods overwritten by 'huge':
##   method      from
##   plot.sim    BDgraph
##   print.sim   BDgraph

library(corrplot)

mat.vars <- read.csv(file = file.path(out_path, "leaf_dpcoa_SEMdata.csv"),
                     row.names = 1)
cor.mat <- cor(mat.vars[, -c(1:3)])
# pdf(file = file.path(out_path, "corrplot_sem_leaf.pdf"), width = 8, height = 8)
# corrplot(cor.mat, method = "number", type = "lower")
# dev.off()

myModel <- ' # direct effect of climate, resources, texture, and plant attributes on fungi
            Axis1 ~ MAP.mm + ph + P + K + Mn + TIN + perc.sand + max.height.m

            # effect of texture on resources
            ph + P + K + Mn + TIN ~ perc.sand

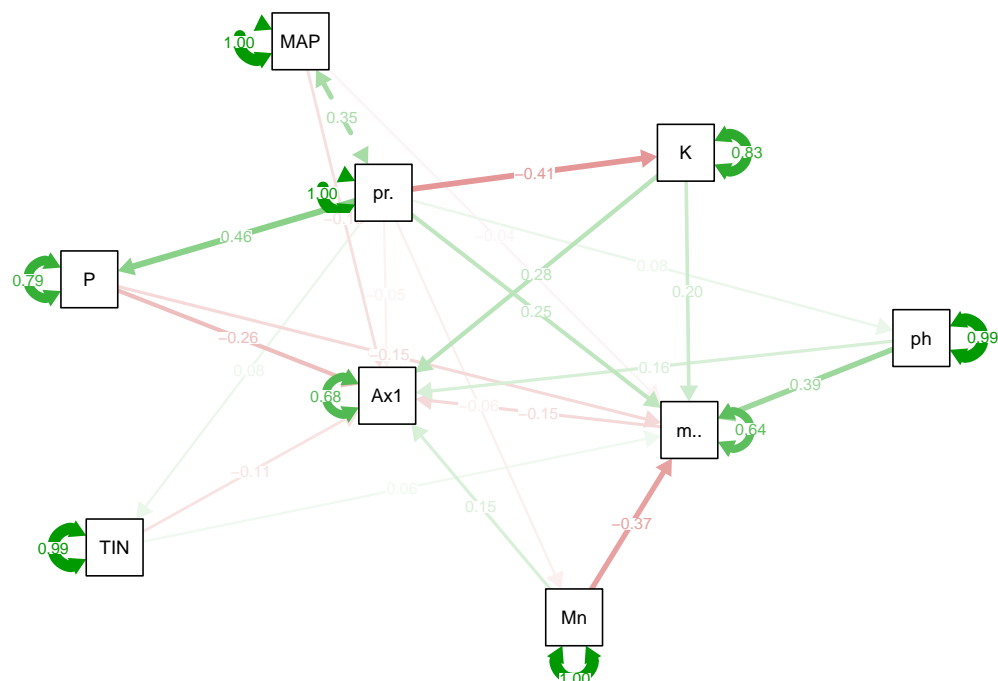
            # effect of stand age on resources

            # effect of climate, resources, texture and stand age on plant size
            max.height.m ~ MAP.mm + ph + P + K + Mn + TIN + perc.sand

            # covariances
            ,

#
fit <- sem(myModel, data=mat.vars)
semPaths(fit, what='std', layout = 'spring')

```



```
summary(fit, standardized=TRUE)
```

```
## lavaan 0.6-6 ended normally after 100 iterations
##
## Estimator ML
## Optimization method NLMINB
## Number of free parameters 27
##
## Number of observations 109
##
## Model Test User Model:
##
## Test statistic 124.208
## Degrees of freedom 15
## P-value (Chi-square) 0.000
##
## Parameter Estimates:
##
## Standard errors Standard
## Information Expected
## Information saturated (h1) model Structured
##
## Regressions:
## Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## Axis1 ~
## MAP.mm -2.027 1.359 -1.492 0.136 -2.027 -0.126
## ph 3.177 1.736 1.830 0.067 3.177 0.162
## P -4.543 1.578 -2.879 0.004 -4.543 -0.261
## K 5.362 1.736 3.089 0.002 5.362 0.275
## Mn 2.706 1.534 1.764 0.078 2.706 0.154
## TIN -2.692 1.872 -1.438 0.150 -2.692 -0.115
## perc.sand -1.002 2.051 -0.488 0.625 -1.002 -0.051
## max.height.m -3.259 2.159 -1.510 0.131 -3.259 -0.150
## ph ~
## perc.sand 0.081 0.096 0.844 0.399 0.081 0.081
## P ~
## perc.sand 0.522 0.096 5.432 0.000 0.522 0.462
## K ~
## perc.sand -0.417 0.088 -4.716 0.000 -0.417 -0.412
## Mn ~
## perc.sand -0.070 0.107 -0.653 0.514 -0.070 -0.062
## TIN ~
## perc.sand 0.064 0.080 0.800 0.424 0.064 0.076
## max.height.m ~
## MAP.mm -0.032 0.060 -0.537 0.591 -0.032 -0.044
## ph 0.352 0.069 5.074 0.000 0.352 0.390
## P -0.122 0.069 -1.768 0.077 -0.122 -0.153
## K 0.180 0.075 2.399 0.016 0.180 0.201
## Mn -0.296 0.062 -4.785 0.000 -0.296 -0.367
## TIN 0.066 0.083 0.799 0.425 0.066 0.061
## perc.sand 0.225 0.088 2.550 0.011 0.225 0.249
##
## Variances:
```

	Estimate	Std.Err	z-value	P(> z)	Std.lv	Std.all
## .Axis1	13.787	1.868	7.382	0.000	13.787	0.684
## .ph	0.052	0.007	7.382	0.000	0.052	0.994
## .P	0.052	0.007	7.382	0.000	0.052	0.787
## .K	0.044	0.006	7.382	0.000	0.044	0.831
## .Mn	0.065	0.009	7.382	0.000	0.065	0.996
## .TIN	0.036	0.005	7.382	0.000	0.036	0.994
## .max.height.m	0.027	0.004	7.382	0.000	0.027	0.638

```
# pvalue 0 (aka model doesn't fit the data)
fitm <- fitMeasures(fit)
fitm['cfi'] # for CFI, a reasonable fit would be over .9
```

```
##      cfi
## 0.5029257
```

```
fitm['rmsea'] # for RMSEA, 0.08 is ok
```

```
##      rmsea
## 0.258445
```

```
# this model isn't terrible, but doesn't fit
modindices(fit) %>%
  filter(op == "~~") %>%
  filter(mi > 10) %>%
  arrange(-mi)
```

##	lhs	op	rhs	mi	epc	sepc.lv	sepc.all	sepc.nox
## 1	ph	~~	K	31.341	0.026	0.026	0.536	0.536
## 2	K	~~	Mn	21.851	0.024	0.024	0.448	0.448
## 3	ph	~~	Mn	20.324	0.025	0.025	0.432	0.432
## 4	P	~~	K	15.008	0.018	0.018	0.371	0.371

Trimmed model:

```
# do model trimming by removing ns paths
# 0. (chisq =124, p < 0.001)
# 1. add covariances with mi > 10 (42)
# 2. examine residuals
# P and Mn (0.258)
# Mn and TIN (-0.21)
# remove TIN (31)
# 3. examine residuals
# P and Mn
# K and MAP
# remove Mn (15, df = 4, p = 0.004)
# 4. examine residuals
# K and MAP
# remove MAP from predicting height (15, df = 5, p = 0.008)
# 5. remove K from predicting height (16, df = 6, p = 0.013)
# 6. remove sand from predicting Ax1 (16, df = 7, p = 0.023)
```

```

# 7. examine residuals
# K and MAP
# option1: remove MAP (2.4, df = 3, p = 0.484, CFI = 1, RMSEA = 0, AIC = 461)
# option2: remove K (8.9, df = 5, p = 0.113, CFI = 0.94, RMSEA = 0.08, AIC = 547)

# myModel.trimmed <- '# direct effect of climate, resources, texture, and plant attributes on fungi
#       Axis1 ~ MAP.mm + ph + P + max.height.m
#
#       # effect of texture on resources
#       ph + P ~ perc.sand
#
#       # effect of stand age on resources
#
#       # effect of climate, resources, texture and stand age on plant size
#       max.height.m ~ ph + P + perc.sand
#
#       # covariances
#       #ph ~~ K
#       #P ~~ K
#       '

myModel.trimmed <- '# direct effect of climate, resources, texture, and plant attributes on fungi
       Axis1 ~ ph + P + K + max.height.m

       # effect of texture on resources
       ph + P + K ~ perc.sand

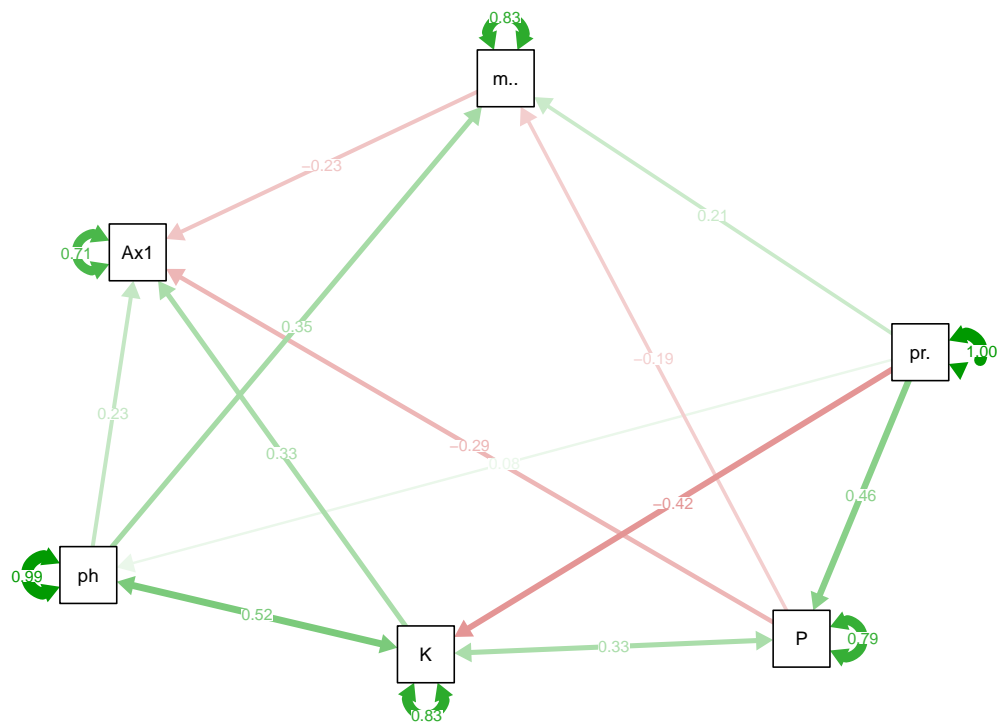
       # effect of stand age on resources

       # effect of climate, resources, texture and stand age on plant size
       max.height.m ~ ph + P + perc.sand

       # covariances
       ph ~~ K
       P ~~ K
       '

fit <- sem(myModel.trimmed, data=mat.vars)
semPaths(fit, what='std', layout = 'spring')

```



```
summary(fit, standardized=TRUE, rsquare = TRUE, fit.measures = TRUE)
```

```
## lavaan 0.6-6 ended normally after 76 iterations
##
##   Estimator           ML
##   Optimization method  NLMINB
##   Number of free parameters      17
##
##   Number of observations      109
##
## Model Test User Model:
##
##   Test statistic           2.452
##   Degrees of freedom         3
##   P-value (Chi-square)      0.484
##
## Model Test Baseline Model:
##
##   Test statistic           159.803
##   Degrees of freedom       15
##   P-value                   0.000
##
## User Model versus Baseline Model:
##
##   Comparative Fit Index (CFI)      1.000
##   Tucker-Lewis Index (TLI)        1.019
##
## Loglikelihood and Information Criteria:
##
##   Loglikelihood user model (H0)      -213.706
```

```

## Loglikelihood unrestricted model (H1)          -212.480
##
## Akaike (AIC)                                   461.412
## Bayesian (BIC)                                 507.165
## Sample-size adjusted Bayesian (BIC)           453.447
##
## Root Mean Square Error of Approximation:
##
## RMSEA                                           0.000
## 90 Percent confidence interval - lower          0.000
## 90 Percent confidence interval - upper          0.150
## P-value RMSEA <= 0.05                          0.594
##
## Standardized Root Mean Square Residual:
##
## SRMR                                           0.027
##
## Parameter Estimates:
##
## Standard errors                                Standard
## Information                                    Expected
## Information saturated (h1) model               Structured
##
## Regressions:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## Axis1 ~
##   ph           4.693   1.951   2.405   0.016   4.693   0.232
##   P            -5.124   1.455  -3.523   0.000  -5.124  -0.286
##   K             6.755   1.828   3.695   0.000   6.755   0.334
##   max.height.m -5.501   2.063  -2.666   0.008  -5.501  -0.233
## ph ~
##   perc.sand      0.081   0.096   0.844   0.399   0.081   0.081
## P ~
##   perc.sand      0.522   0.096   5.432   0.000   0.522   0.462
## K ~
##   perc.sand     -0.417   0.087  -4.784   0.000  -0.417  -0.417
## max.height.m ~
##   ph             0.300   0.075   4.001   0.000   0.300   0.350
##   P             -0.146   0.075  -1.958   0.050  -0.146  -0.192
##   perc.sand      0.178   0.085   2.107   0.035   0.178   0.207
##
## Covariances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .ph ~~
##   .K            0.024   0.005   5.011   0.000   0.024   0.516
## .P ~~
##   .K            0.016   0.004   3.782   0.000   0.016   0.333
##
## Variances:
##           Estimate Std.Err z-value P(>|z|) Std.lv Std.all
## .Axis1       15.089   2.044   7.382   0.000  15.089   0.709
## .ph           0.052   0.007   7.382   0.000   0.052   0.994
## .P           0.052   0.007   7.382   0.000   0.052   0.787
## .K           0.043   0.006   7.610   0.000   0.043   0.826

```



```
##      .max.height.m      0.032      0.004      7.382      0.000      0.032      0.828
##
## R-Square:
##              Estimate
##      Axis1      0.291
##      ph        0.006
##      P         0.213
##      K         0.174
##      max.height.m 0.172
```

```
capture.output(summary(fit, standardized=TRUE, rsquare = TRUE, fit.measures = TRUE),
               file = file.path(out_path, "sem_l_std.txt"))
```

```
summary(fit, rsquare = TRUE)
```

```
## lavaan 0.6-6 ended normally after 76 iterations
##
##      Estimator                      ML
##      Optimization method          NLMINB
##      Number of free parameters      17
##
##      Number of observations          109
##
## Model Test User Model:
##
##      Test statistic                  2.452
##      Degrees of freedom                3
##      P-value (Chi-square)             0.484
##
## Parameter Estimates:
##
##      Standard errors                  Standard
##      Information                      Expected
##      Information saturated (h1) model  Structured
##
## Regressions:
##              Estimate  Std.Err  z-value  P(>|z|)
##      Axis1 ~
##      ph              4.693    1.951    2.405    0.016
##      P              -5.124    1.455   -3.523    0.000
##      K               6.755    1.828    3.695    0.000
##      max.height.m   -5.501    2.063   -2.666    0.008
##      ph ~
##      perc.sand        0.081    0.096    0.844    0.399
##      P ~
##      perc.sand        0.522    0.096    5.432    0.000
##      K ~
##      perc.sand       -0.417    0.087   -4.784    0.000
##      max.height.m ~
##      ph               0.300    0.075    4.001    0.000
##      P              -0.146    0.075   -1.958    0.050
##      perc.sand        0.178    0.085    2.107    0.035
##
## Covariances:
```

```
##              Estimate Std.Err z-value P(>|z|)
## .ph ~~
## .K          0.024    0.005   5.011   0.000
## .P ~~
## .K          0.016    0.004   3.782   0.000
##
## Variances:
##              Estimate Std.Err z-value P(>|z|)
## .Axis1      15.089    2.044   7.382   0.000
## .ph         0.052    0.007   7.382   0.000
## .P          0.052    0.007   7.382   0.000
## .K          0.043    0.006   7.610   0.000
## .max.height.m 0.032    0.004   7.382   0.000
##
## R-Square:
##              Estimate
## Axis1        0.291
## ph           0.006
## P            0.213
## K            0.174
## max.height.m 0.172
```

```
capture.output(summary(fit, rsquare = TRUE),
  file = file.path(out_path, "sem_l_unstd.txt"))
```

```
fitm <- fitMeasures(fit)
fitm
```

```
##              npar              fmin              chisq              df
##              17.000              0.011              2.452              3.000
##              pvalue      baseline.chisq      baseline.df      baseline.pvalue
##              0.484              159.803              15.000              0.000
##              cfi              tli              nnfi              rfi
##              1.000              1.019              1.019              0.923
##              nfi              pnfi              ifi              rni
##              0.985              0.197              1.003              1.004
##              logl      unrestricted.logl              aic              bic
##              -213.706              -212.480              461.412              507.165
##              ntotal              bic2              rmsea      rmsea.ci.lower
##              109.000              453.447              0.000              0.000
##              rmsea.ci.upper      rmsea.pvalue              rmr              rmr_nomean
##              0.150              0.594              0.063              0.063
##              srmr      srmr_bentler      srmr_bentler_nomean              crmr
##              0.027              0.027              0.027              0.031
##              crmr_nomean      srmr_mplus      srmr_mplus_nomean              cn_05
##              0.031              0.027              0.027              348.443
##              cn_01              gfi              agfi              pgfi
##              505.393              0.993              0.950              0.142
##              mfi              ecvi
##              1.003              0.334
```

```
fitm['cfi'] # for CFI, a reasonable fit would be over .9
```

```
## cfi
```

```
## 1

fitm['rmsea'] # for RMSEA, 0.08 is ok

## rmsea
## 0

fitm['aic']

## aic
## 461.412

# modindices(fit) %>%
# filter(mi > 1) %>%
# arrange(-mi)

# estimated correlations
# inspect(fit, what="cor.all")
# observed correlations
# lavCor(fit)
# residuals
resid(fit, "cor")

## $type
## [1] "cor.bollen"
##
## $cov
##      Axis1  ph      P      K      mx.hg. prc.sn
## Axis1      0.000
## ph      -0.008  0.000
## P        0.022  0.075  0.000
## K       -0.010  0.020  0.034  0.000
## max.height.m 0.007 -0.013  0.026  0.049  0.000
## perc.sand  -0.061  0.000  0.000  0.005  0.001  0.000

# Large positive values indicate the model underpredicts the correlation;
# large negative values suggest overprediction of correlation.
# Usually values |r|.1 are worth closer consideration
```

Multiple regression:

```
modl <- lm(Axis1 ~ max.height.m + ph + K + P + perc.sand, data = mat.vars)
an <- anova(modl)
afss <- an$"Sum Sq"
cbind(an, PctExp=afss/sum(afss)*100)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	PctExp
max.height.m	1	23.80795	23.80795	1.505959	0.2225542617	1.039409
ph	1	250.81693	250.81693	15.865287	0.0001270287	10.950181
K	1	186.12876	186.12876	11.773472	0.0008659170	8.126021
P	1	185.04035	185.04035	11.704626	0.0008951246	8.078503
perc.sand	1	16.38987	16.38987	1.036732	0.3109683698	0.715550
Residuals	103	1628.34394	15.80916	NA	NA	71.090337

```
#capture.output(print(cbind(an,PctExp=afss/sum(afss)*100)), file = "leaf_mr.txt")
```

Leaf - including lat and lon (commented out)

Root

Multiple regression

Soil

Multiple regression

C. Investigate SEM results with bivariate plots

1. Set up functions, levels, colors

Format dataframe functions

Format plot functions

Set phylum levels and colors

2. Make plotting dataframes – samples

3. Make plotting dataframes and set up universal scale – ASVs

by class (commented out)

by phylum

4. Plot – Direct only

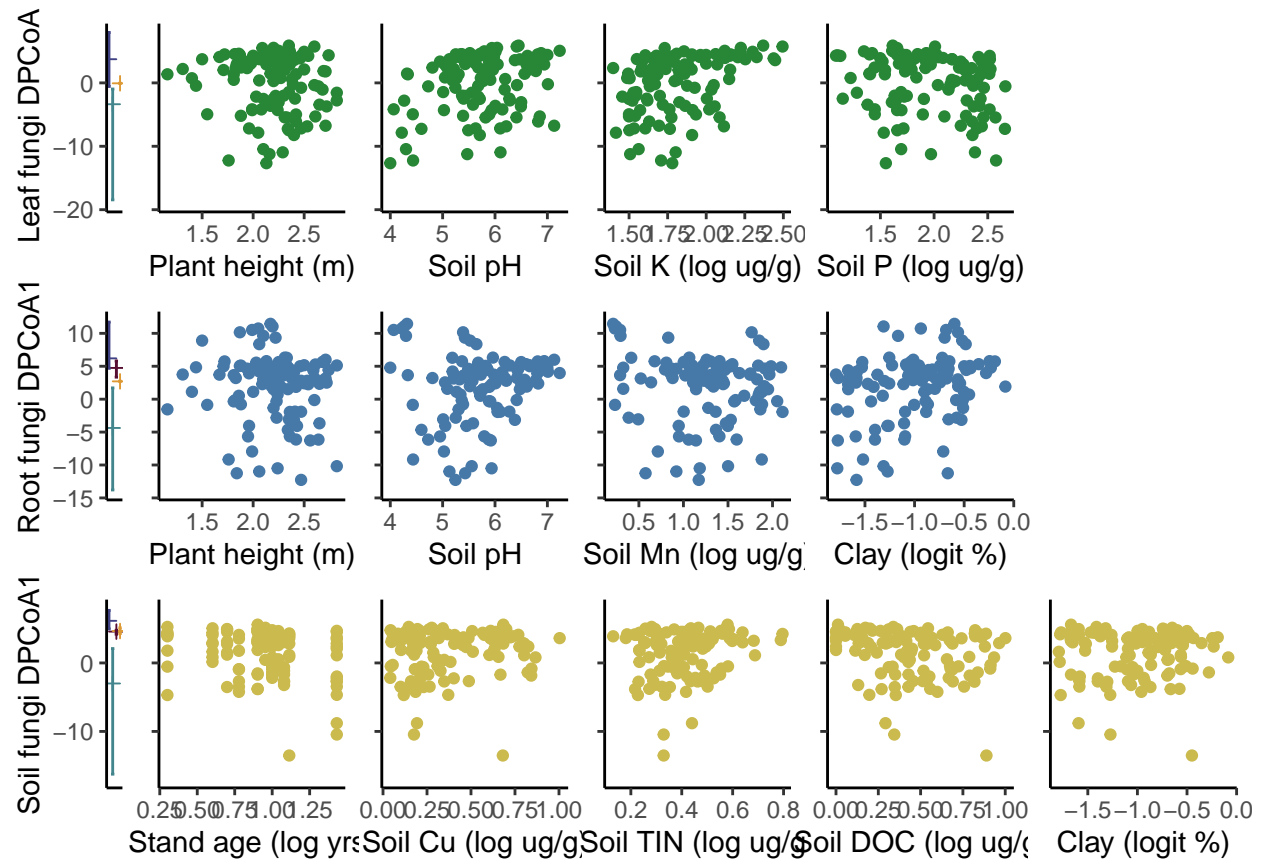
Leaf

Root

Soil

save the plots

```
## Loading required package: ggpubr
```



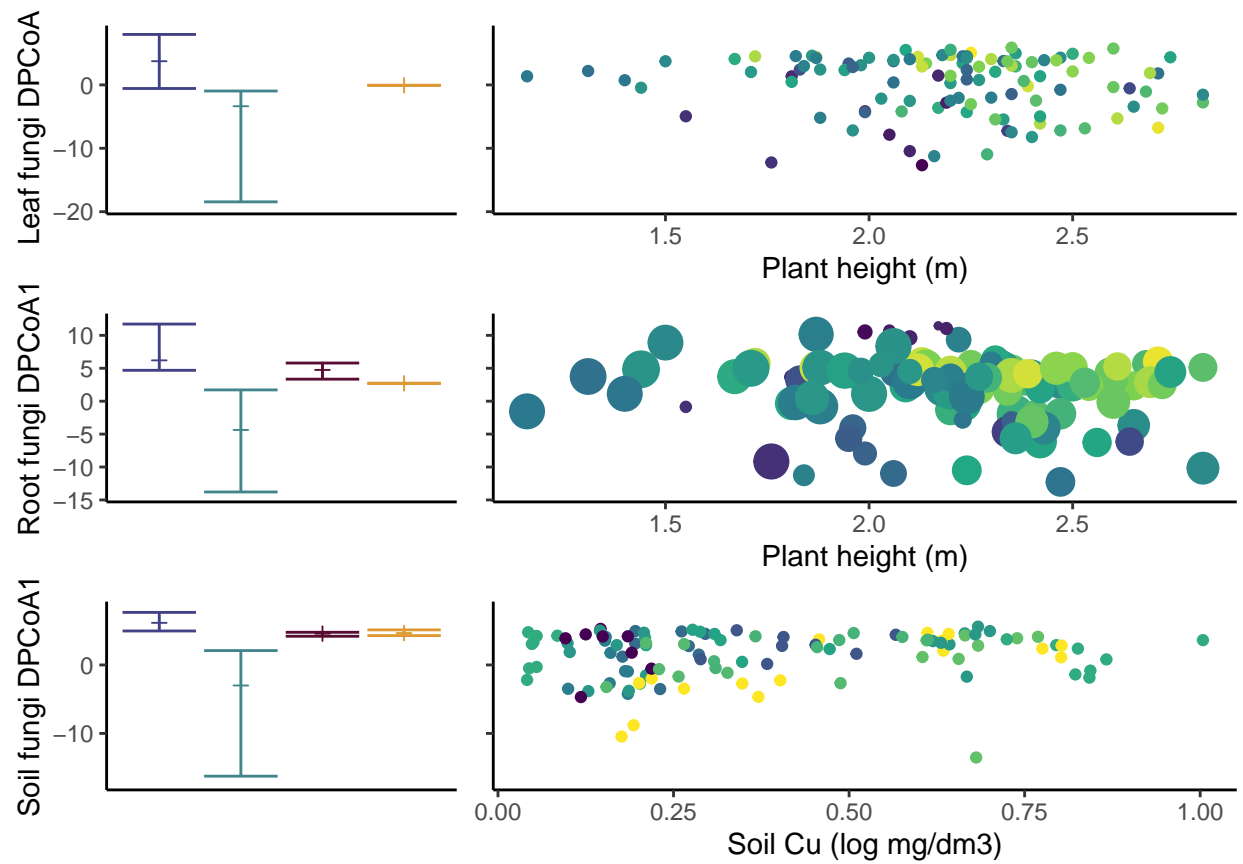
5. Plot – Indirect only

Leaf

Root

Soil

save the plots



add plots that show relevant correlations between predictor vars

