MB590-012 Microbiome Analysis

Christine V. Hawkes

March 2, 2021

Contents

Topic: EXPLORATORY ANALYSIS - CORE MICROBIOMES

```
References:
Risley (2020) Applying the core microbiome to understand host-microbe systems.
J Animal Ecology 89: 1549-1558. DOI: 10.1111/1365-2656.13229

Shade & Stopnisek (2019)Abundance-occupancy distributions to prioritize plant core
microbiome membership. Curr Op Microbio 49: 50-58. DOI: 10.1016/j.mib.2019.09.008

Data: Oono et al. (2020) Species diversity of fungal endophytes across a stress gradient
for plants. New Phytologist 228: 210-225. DOI: 10.1111/nph.16709
```

SETUP

Load and install R packages

```
library(phyloseq)
library(microbiome)
library(ggplot2)
library(tidyverse)
library(compositions)
library(rmarkdown)
library(knitr)
library(Biostrings)
library(vegan)

#install.packages("RColorBrewer")
library(RColorBrewer)
#install.packages("reshape2")
library(reshape2)
#devtools::install_github("Russel88/MicEco")
library(MicEco)
```

Load and prepare data

All files are on GitHub, add the raw url path to the read commands.

OTU table

```
OTU_data <- read.csv("wk8_97OTU_table.csv", stringsAsFactors=FALSE, row.names=1, header=TRUE)
# str(OTU_data)
# anyNA(OTU_data)
colnames(OTU_data) # taxa are rows!
```

```
##  [1] "T1P10" "T1P1"  "T1P2"  "T1P3"  "T1P4"  "T1P5"  "T1P6"  "T1P7"  "T1P8"
## [10] "T1P9"  "T2P10" "T2P1"  "T2P2"  "T2P3"  "T2P5"  "T2P8"  "T2P9"  "T2V10"
## [19] "T2V1"  "T2V2"  "T2V3"  "T2V4"  "T2V5"  "T2V6"  "T2V7"  "T2V8"  "T2V9"
## [28] "T3P10" "T3P1"  "T3P2"  "T3P3"  "T3P4"  "T3P5"  "T3P6"  "T3P7"  "T3P8"
## [37] "T3P9"  "T3V10" "T3V1"  "T3V2"  "T3V3"  "T3V4"  "T3V5"  "T3V6"  "T3V7"
## [46] "T3V8"  "T3V9"  "T4P10" "T4P1"  "T4P2"  "T4P3"  "T4P4"  "T4P5"  "T4P6"
## [55] "T4P7"  "T4P8"  "T4P9"  "T4V10" "T4V1"  "T4V2"  "T4V3"  "T4V4"  "T4V6"
## [64] "T4V8"  "T4V9"  "T5P10" "T5P1"  "T5P2"  "T5P3"  "T5P4"  "T5P5"  "T5P6"
## [73] "T5P7"  "T5P8"  "T5P9"  "T5V10" "T5V1"  "T5V2"  "T5V3"  "T5V4"  "T5V5"
## [82] "T5V6"  "T5V7"  "T5V8"  "T5V9"
```

Sample/environmental data

```
SAM_data <- read.csv("wk8_EnvDataAll.csv", row.names=1, header=TRUE, sep=",")
# str(SAM_data) # check for treatment factors and continuous numeric vars
SAM_data[1,]
```

```
##      Terrace Species Replicate EcoType Carbon Nitrogen Phenolics Aluminum Boron
## T1P1       1   Pinus         1     T1P  43.69     1.08       1.8   595.98    21
##      Calcium Cadmium Copper Iron Potassium Magnesium Manganese Molybdenum
## T1P1 2666.47    0.09   6.42 54.1   7562.62   1776.28    292.93       0.71
##       Sodium Phosphorus Lead  Sulfur Silicon  Zinc Degrees_long Min_long
## T1P1 1444.15    1722.58 0.23 1309.67   30.26 10.89           39       22
##      Sec_long Decimals_Lat Degrees_long.1 Min_long.1 Sec_long.1 Decimals_Long
## T1P1       39      39.3775            123         48         54      -123.815
```

```
anyNA(SAM_data)# will need to keep an eye on these NAs
```

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

Taxonomy table

```
TAX_data <- read.csv("wk8_97Taxa.csv", row.names=1, header=TRUE, sep=",")
TAX_data <- as.matrix(TAX_data)
# str(TAX_data)
TAX_data[1,]
```

```
##            Phylum            ClassI             Class             Order
##      "Ascomycota" "Arthoniomycetes" "Arthoniomycetes"   " Roccellaceae"
##            Family
##       " Sigridea"
```

References sequences for OTUs

```
REF_data <- Biostrings::readDNAStringSet("wk8_Fungi_seq.fasta", format="fasta")
str(REF_data)
```

```
## Formal class ’DNAStringSet’ [package "Biostrings"] with 5 slots
##   ..@ pool           :Formal class ’SharedRaw_Pool’ [package "XVector"] with 2 slots
##   .. .. ..@ xp_list                    :List of 1
##   .. .. .. ..$ :<externalptr>
##   .. .. ..@ .link_to_cached_object_list:List of 1
##   .. .. .. ..$ :<environment: 0x0000000025878048>
##   ..@ ranges         :Formal class ’GroupedIRanges’ [package "XVector"] with 7 slots
##   .. .. ..@ group          : int [1:1193] 1 1 1 1 1 1 1 1 1 1 ...
##   .. .. ..@ start          : int [1:1193] 1 273 569 810 1084 1411 1648 2014 2280 2561 ...
##   .. .. ..@ width          : int [1:1193] 272 296 241 274 327 237 366 266 281 242 ...
##   .. .. ..@ NAMES          : chr [1:1193] "OTU1" "OTU2" "OTU3" "OTU4" ...
##   .. .. ..@ elementType    : chr "ANY"
##   .. .. ..@ elementMetadata: NULL
##   .. .. ..@ metadata       : list()
##   ..@ elementType    : chr "DNAString"
##   ..@ elementMetadata: NULL
##   ..@ metadata       : list()
```

Make phyloseq object

```
ASV <- phyloseq::otu_table(OTU_data, taxa_are_rows = TRUE)
SAM <- phyloseq::sample_data(SAM_data)
TAX <- phyloseq::tax_table(TAX_data)
REF <- phyloseq::refseq(REF_data)


ps <- phyloseq::phyloseq(ASV, SAM, TAX, REF)
ps
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1193 taxa and 85 samples ]
## sample_data() Sample Data:       [ 85 samples by 31 sample variables ]
## tax_table()   Taxonomy Table:    [ 1193 taxa by 5 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 1193 reference sequences ]
```

Remove singletons

```
# remove singletons
ps_nosing <- phyloseq::prune_taxa(phyloseq::taxa_sums(ps) > 1, ps)
phyloseq::ntaxa(ps_nosing)
```

```
## [1] 1192
```

DATA TRANSFORMATIONS

```
# microbiome::core requires ASVs to be in relative abundance
ps_ra <- microbiome::transform(ps_nosing, "compositional")

# Some approaches require a clr transformation
# ps_nosing_clr <- microbiome::transform(ps_nosing, transform="clr")
```

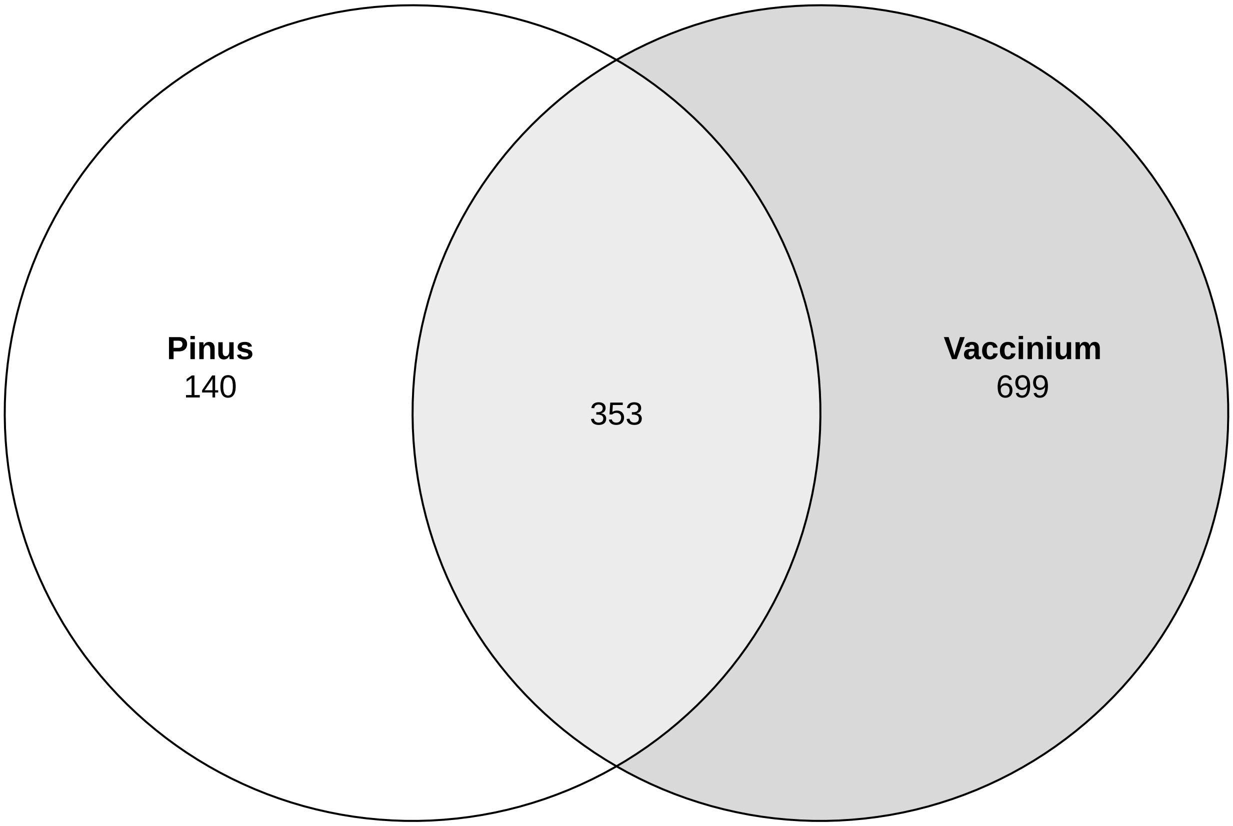
CORE MICROBIOME

We'll use `microbiome::core` to identify core taxa in the ps object.
Can adjust two parameters:

- `detection` = relative abundances of the ASVs
- `prevalence` = proportion of samples in which the ASVs are present

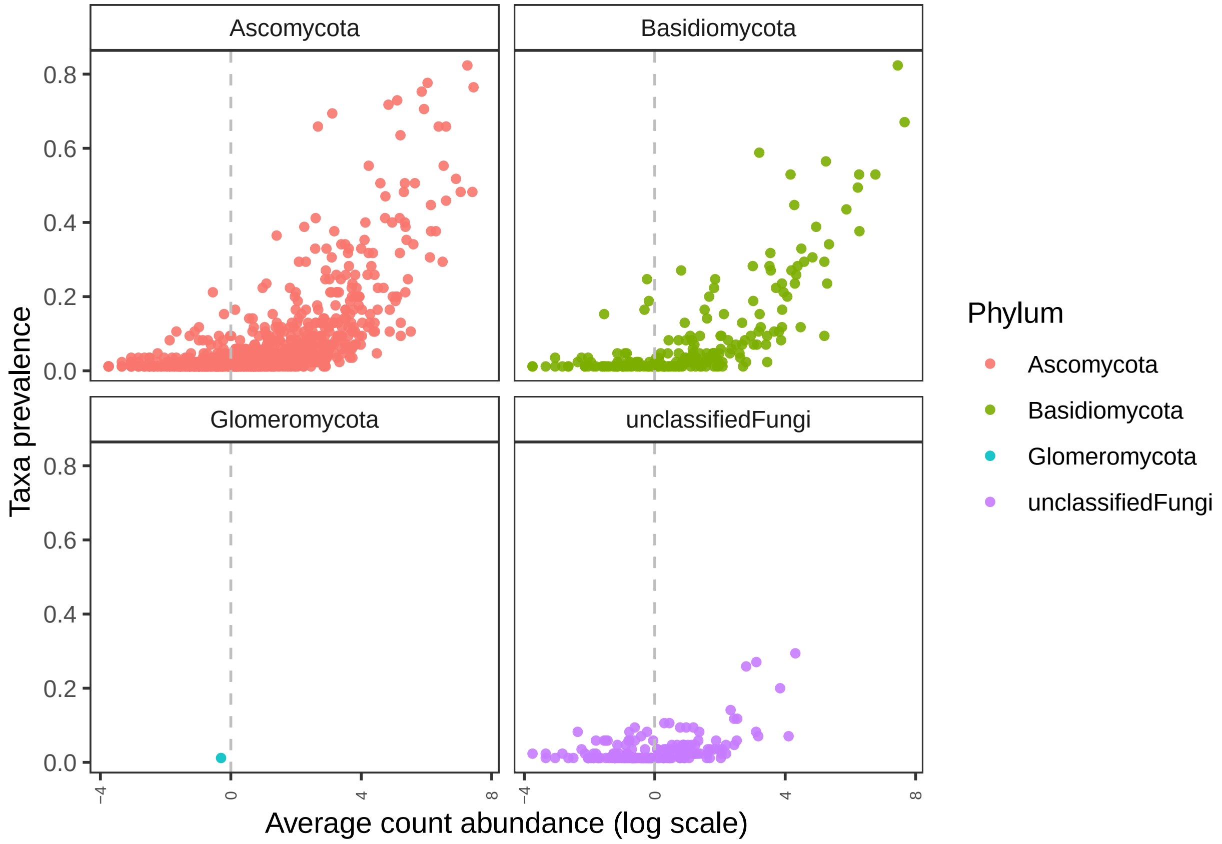
Explore the potential core with a Venn diagram

```
p1 <- MicEco::ps_venn(ps_ra, "Species", fraction=0, weight=FALSE,
                      type="counts", relative=FALSE, plot=TRUE)
# if you set plot=FALSE, can get a list of taxa
p1
```



Explore prevalence and abundance

```
# plot prevalence as a function of log(counts)
microbiome::plot_taxa_prevalence(ps_nosing, "Phylum", detection = 0/100)
```



```
# make table of taxa with >1% detection (rel abund) threshold
# use kable to get tabular format
# head limits to first 5 rows
taxa.prev <- knitr::kable(head(microbiome::prevalence(ps_ra, detection = 1/100, sort=TRUE)))
taxa.prev
```

	x
OTU1	0.5764706
OTU5	0.5294118
OTU9	0.5058824
OTU16	0.4588235
OTU3	0.3411765
OTU40	0.3176471

Identify core members based on detection and prevalence

Core taxa defined as present in >50% of samples at any rel abund (>0)
Use `core_members` on ps object to identify core taxa

```
core.taxa1 <- microbiome::core_members(ps_ra, detection = 0, prevalence = 50/100)
core.taxa1
```

```
##  [1] "OTU4"    "OTU5"    "OTU1"    "OTU16"   "OTU2"    "OTU215"  "OTU9"
##  [8] "OTU18"   "OTU12"   "OTU19"   "OTU442"  "OTU3008" "OTU40"   "OTU46"
## [15] "OTU21"   "OTU3828" "OTU3550" "OTU589"  "OTU675"  "OTU72"   "OTU183"
## [22] "OTU1537" "OTU1336" "OTU3747" "OTU3052"
```

Core taxa defined as present in >50% of samples with >1% rel abundance Use `core_members` on ps object to identify core taxa

```
core.taxa2 <- microbiome::core_members(ps_ra, detection = 1/100, prevalence = 50/100)
core.taxa2
```

```
## [1] "OTU5" "OTU1" "OTU9"
```

Core taxa defined as present in >50% of samples with >0.1% rel abundance Use `core` to generate new ps object with only core taxa

```
ps_core <- microbiome::core(ps_ra, detection = 0.1/100, prevalence = 50/100)
ps_core
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 10 taxa and 85 samples ]
## sample_data() Sample Data:       [ 85 samples by 31 sample variables ]
## tax_table()   Taxonomy Table:    [ 10 taxa by 5 taxonomic ranks ]
## refseq()      DNAStringSet:      [ 10 reference sequences ]
```

```
# retrieve core taxa and check match to core.taxa2
core.taxa3 <- phyloseq::taxa_names(ps_core)
core.taxa3
```

```
##  [1] "OTU5"    "OTU1"    "OTU16"   "OTU9"    "OTU12"   "OTU442"  "OTU3008"
##  [8] "OTU40"   "OTU3828" "OTU589"
```

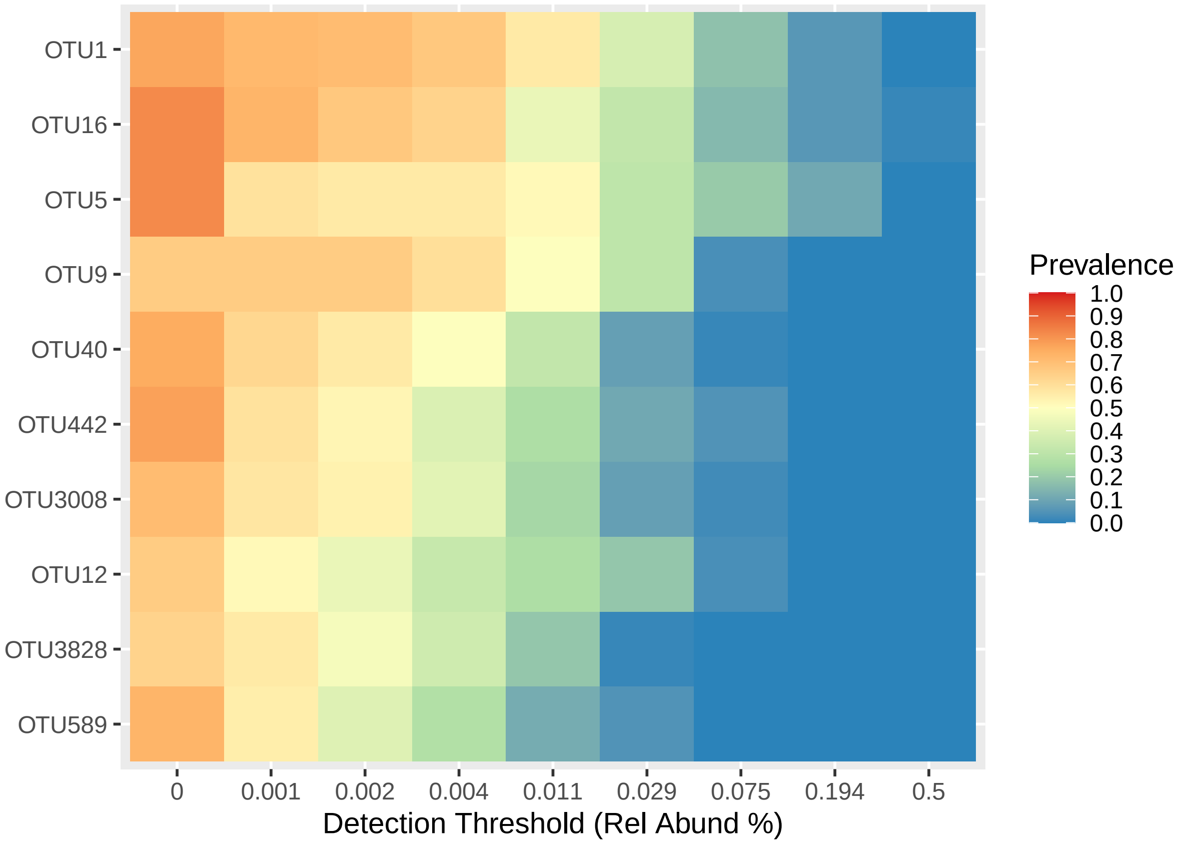
Link core OTUs to their taxonomic IDs

```
tax.core.id <- phyloseq::tax_table(ps_core) # get taxonomy table from ps object
tax.core.id <- as.data.frame(tax.core.id) # convert to dataframe
tax.core.id$OTU <- rownames(tax.core.id) # make OTU IDs the last column

core.taxa.class <- dplyr::filter(tax.core.id, rownames(tax.core.id) %in% core.taxa3)
knitr::kable(head(core.taxa.class))
```

	Phylum	ClassI	Class	Order	Family	OTU
OTU5	Basidiomycota	Malasseziomycetes	Malasseziomycetes	Malasseziaceae	Malassezia	OTU5
OTU1	Ascomycota	Dothideomycetes	Dothideomycetes	Capnodiales	Cladosporiaceae	OTU1
OTU16	Ascomycota	Dothideomycetes	Dothideomycetes	Capnodiales	Teratosphaeriaceae	OTU16
OTU9	Ascomycota	Dothideomycetes	Dothideomycetes	Pleosporales	Pleosporineae	OTU9
OTU12	Ascomycota	Dothideomycetes	Dothideomycetes	Capnodiales	Cladosporiaceae	OTU12
OTU442	Ascomycota	Dothideomycetes	Dothideomycetes	Capnodiales	Teratosphaeriaceae	OTU442

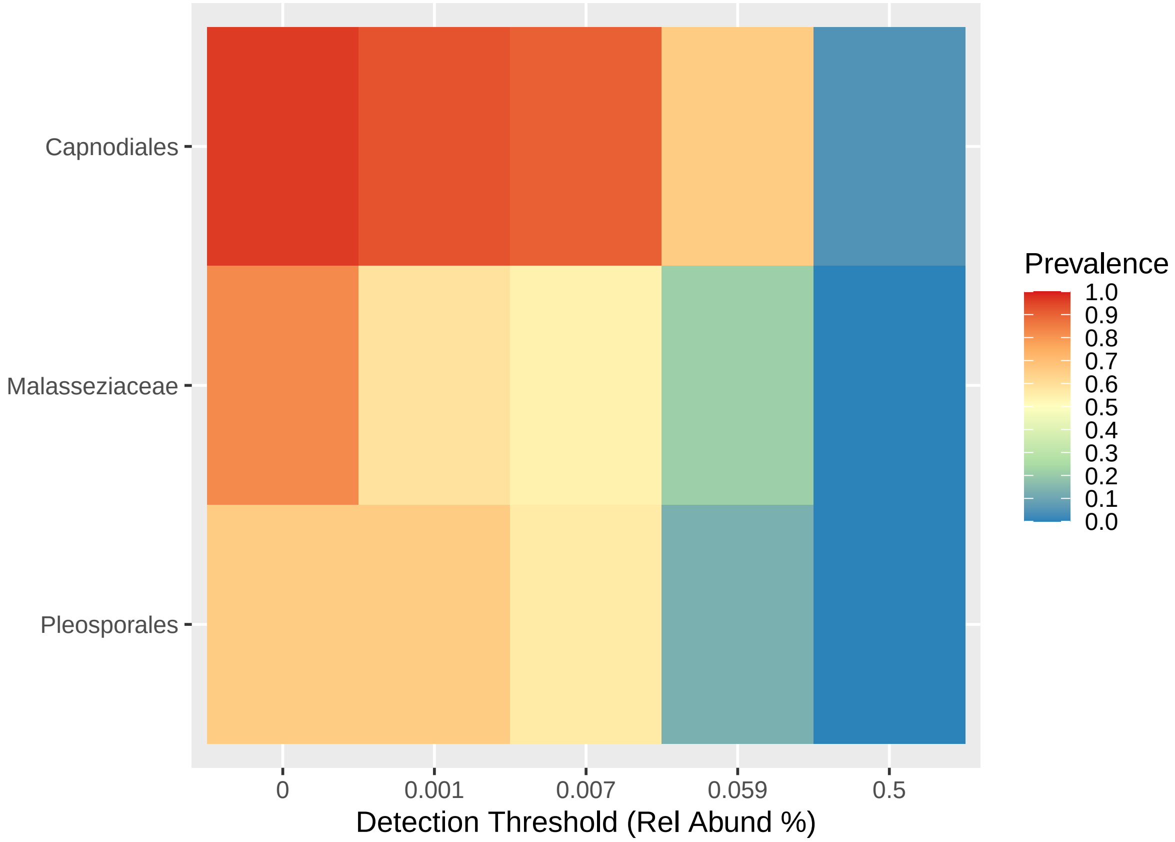
Visualize core microbiome

```
# heatmap of core taxa by detection and prevalence
prevalences <- seq(from=0, to=1, by=0.1)
detections <-  round(10^seq(log10(1e-4), log10(0.5), length = 10), 3)

microbiome::plot_core(ps_core, plot.type = "heatmap",
           colours = rev(RColorBrewer::brewer.pal(5, "Spectral")),
           prevalences = prevalences,
           detections = detections) +
           ggplot2::labs(x = "Detection Threshold (Rel Abund %)")
```

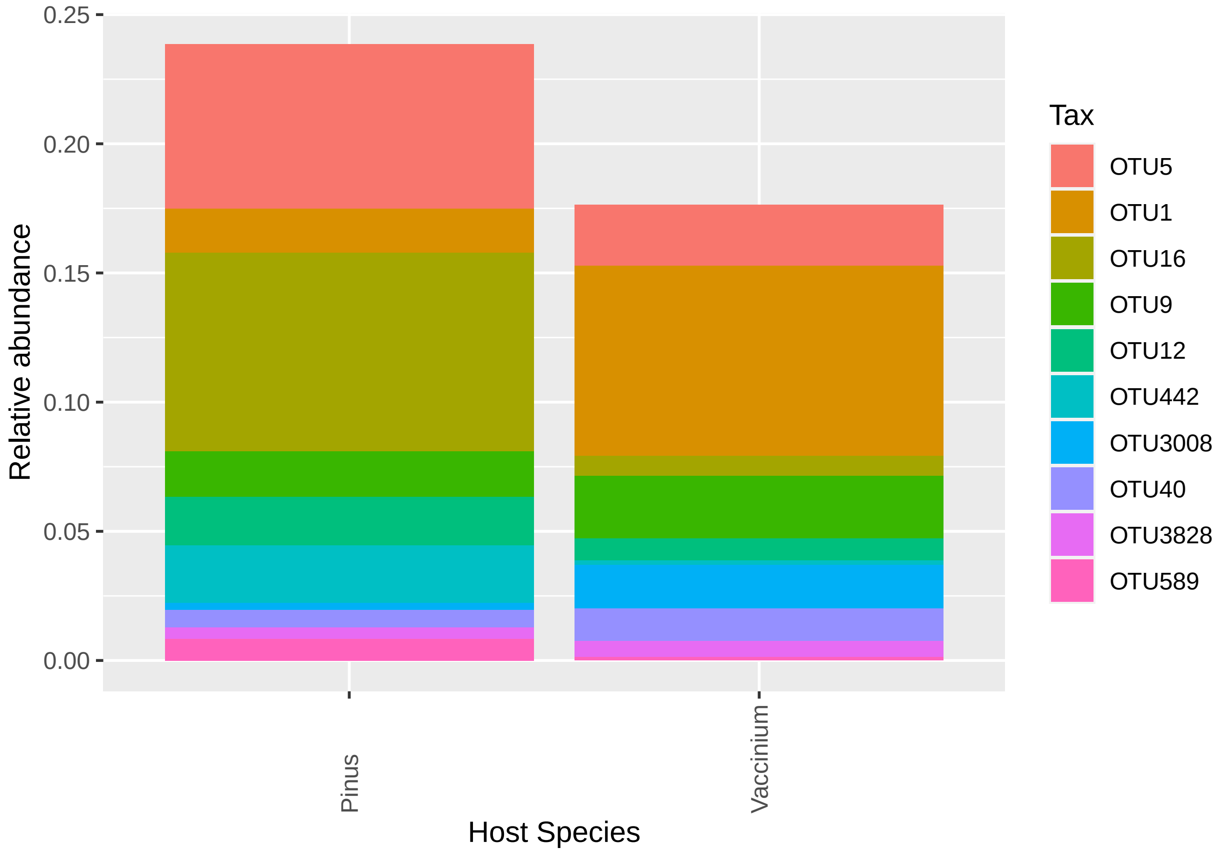


```
# heat map with core taxa aggregated by Order
ps_core_order <- microbiome::aggregate_taxa(ps_core, "Order")
```

```
prevalences <- seq(0.05, 1, 0.05)
detections <-  round(10^seq(log10(1e-4), log10(0.5), length = 5), 3)

microbiome::plot_core(ps_core_order, plot.type = "heatmap",
           colours = rev(RColorBrewer::brewer.pal(5, "Spectral")),
           prevalences = prevalences,
           detections = detections) +
           ggplot2::labs(x = "Detection Threshold (Rel Abund %)")
```

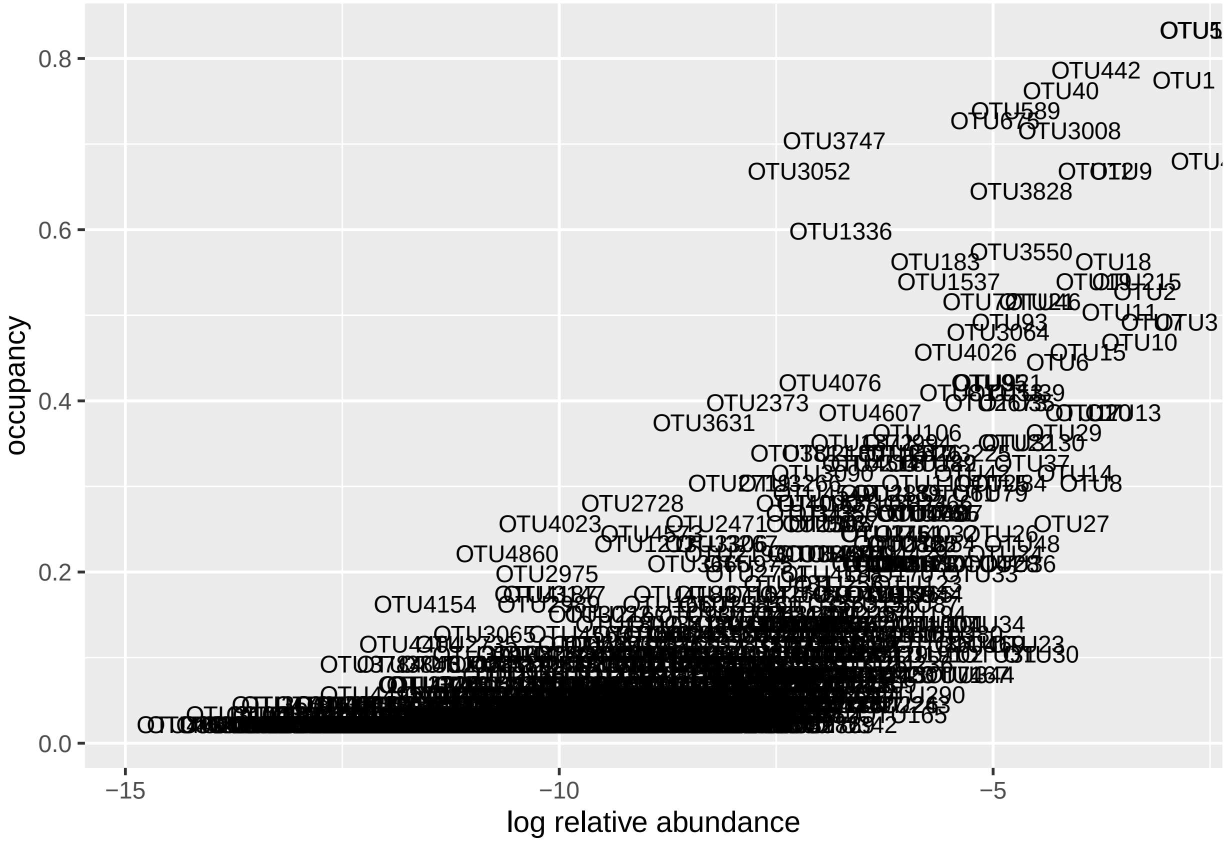


```
# barplot of core taxa by host plant species
microbiome::plot_composition(ps_core,
                             average_by="Species",
                             plot.type = "barplot",
                             sample.sort="Species") +
                 guides(fill = guide_legend(ncol = 1)) +
                 labs(x = "Host Species",
                 y = "Relative abundance")
```


0.25
0.20
0.15
0.10
0.05
0.00
Relative abundance
Pinus
Vaccinium
Host Species
Tax
OTU5
OTU1
OTU16
OTU9
OTU12
OTU442
OTU3008
OTU40
OTU3828
OTU589

Alternative approach using abundance-occupancy (Shade)

```
# full approach is too extensive for class
# below is the code to make abundance-occupancy curves from the data
# Shade lab has posted R code for full procedure:
# https://github.com/ShadeLab/PAPER_Shade_CurrOpinMicro/blob/master/script/Core_prioritizing_script.R

# obtain otu table from ps object and transpose
otu <- phyloseq::otu_table(ps_nosing)

# some approaches require you to rarefy data, if so:
  # min_r <- min(sample_sums(ps_nosing))
  # min_r
  # otu_r <- rrarefy(otu, min_r)

# calculate occupancy and relative abundance
otu_pa <- 1*((otu>0)==1) # convert to pres-abs
otu_occ <- rowSums(otu_pa)/ncol(otu_pa) # calculate occupancy
otu_rel <- apply(vegan::decostand(otu, method="total", MARGIN=2), 1, mean) # mean rel abund

# merge files and rank by relative abundance
occ_abun <- dplyr::add_rownames(as.data.frame(cbind(otu_occ, otu_rel)), "otu")
oa_rank <- dplyr::arrange(occ_abun, otu_rel)
oa_rank_log <- oa_rank %>%
  dplyr::mutate(log_rel = log(otu_rel))

# create occupancy abundance plot with OTU labels
ggplot2::ggplot(oa_rank_log, aes(x=log_rel, y=otu_occ, label=otu)) +
  xlab("log relative abundance") +
  ylab("occupancy") +
  geom_text(hjust=0, vjust=0, size=3)
```



```
# to better see in crowded plot, can try adding:
#  + geom_jitter()
# to read overlapping OTU names, can alternatively replace geom_text command with
#   geom_text_repel(size=3, min.segment.length=Inf, max.overlaps = Inf,  point.size=NA)
# but note this changes shape of the curve
```

CODING EXERCISES

Please submit as a knitted html or pdf markdown to GitHub due on 3/9

1. **Subset to Vaccinium unique OTUs and clr transform**

- Goal is to retain only fungal OTUs found uniquely associated with Vaccinium by removing Pinus OTUs
- p1 venn diagram can help you to confirm expected numbers
- use `phylosmith::unique_taxa` to identify taxa associated only with Pinus in ps_nosing
  - https://schuyler-smith.github.io/phylosmith/analytics.html#unique_taxa
  - `devtools::install_github("schuyler-smith/phylosmith")`
  - `library(phylosmith)`
    library(phylosmith)
- convert list to vector using base::unlist
  - https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/unlist
  - note that this gives you unique Pinus OTUs + OTUs shared with Pinus
- export list of all taxa with `phyloseq::taxa_names` from ps_nosing
  - make new ps object ps_vacc by subsetting the list by removing taxa from Pinus
  - hint: look back at code from lulu
- in new ps object, ps_vacc
  - use remaining taxa list to retain only truly unique taxa with `phyloseq::prune_taxa`
  - use `phyloseq::subset_samples` to limit to Species==“Vaccinium”
  - check for and remove new singletons
- create ps_vacc_clr with clr transformed otu_table using `microbiome::transform`
- include new Vaccinium venn diagram by EcoType
- optional: if you have time and want to practice more, repeat for Pinus

2. **Examine core microbiome for Vaccinium only**

- for one detection and prevalence level, compare clr and rel abund data transforms
- vary detection and prevalence for clr data
  - adjust only detection up and down (at least 3 levels)
  - adjust only prevalence up and down (at least 3 levels)
- describe the effects on the size and characteristics of the core community
- optional: if you want more practice, repeat for Pinus

3. **Identify the core microbiota of built-in `soilrep` data**

- load built-in `soilrep` data and examine
- remove clipped samples with `phyloseq::subset_samples`
- remove singletons with `phyloseq::prune_taxa`
- identify core with `microbiome::core`
  - indicate why you selected your specific prevalence and detection settings
- produce a table of core ASVs using `kable` and specify column name
- plot results as heatmap, barplot, or other plot of your choice

---

Session Info

```r
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
## 
## Matrix products: default
## 
## locale:
## [1] LC_COLLATE=English_United States.1252 
## [2] LC_CTYPE=English_United States.1252   
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C                          
## [5] LC_TIME=English_United States.1252    
## 
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods  
## [8] base     
## 
## other attached packages:
##  [1] MicEco_0.9.17       reshape2_1.4.4      RColorBrewer_1.1-2 
##  [4] vegan_2.5-7         lattice_0.20-45     permute_0.9-7      
##  [7] Biostrings_2.62.0   GenomeInfoDb_1.30.1 XVector_0.34.0     
## [10] IRanges_2.28.0      S4Vectors_0.32.3    BiocGenerics_0.40.0
## [13] knitr_1.37          rmarkdown_2.11      compositions_2.0-4 
## [16] forcats_0.5.1       stringr_1.4.0       dplyr_1.0.8        
## [19] purrr_0.3.4         readr_2.1.2         tidyr_1.2.0        
## [22] tibble_3.1.6        tidyverse_1.3.1     microbiome_1.16.0  
## [25] ggplot2_3.3.5       phyloseq_1.38.0    
## 
## loaded via a namespace (and not attached):
##   [1] Rtsne_0.15             colorspace_2.0-3       ellipsis_0.3.2        
##   [4] htmlTable_2.4.0        base64enc_0.1-3        fs_1.5.2              
##   [7] rstudioapi_0.13        farver_2.1.0           mvtnorm_1.1-3         
##  [10] fansi_1.0.2            lubridate_1.8.0        xml2_1.3.3            
##  [13] codetools_0.2-18       splines_4.1.2          robustbase_0.93-9     
##  [16] polyclip_1.10-0        ade4_1.7-18            Formula_1.2-4         
##  [19] jsonlite_1.8.0         broom_0.7.12           cluster_2.1.2         
##  [22] dbplyr_2.1.1           png_0.1-7              pheatmap_1.0.12       
##  [25] compiler_4.1.2         httr_1.4.2             backports_1.4.1       
##  [28] assertthat_0.2.1       Matrix_1.4-0           fastmap_1.1.0         
##  [31] cli_3.2.0              htmltools_0.5.2        tools_4.1.2           
##  [34] igraph_1.2.11          gtable_0.3.0           glue_1.6.2            
##  [37] GenomeInfoDbData_1.2.7 Rcpp_1.0.8             bbmle_1.0.24          
##  [40] Biobase_2.54.0         eulerr_6.1.1           cellranger_1.1.0      
##  [43] vctrs_0.3.8            rhdf5filters_1.6.0     multtest_2.50.0       
##  [46] ape_5.6-1              nlme_3.1-155           iterators_1.0.14      
##  [49] tensorA_0.36.2         polylabelr_0.2.0       xfun_0.29
##  [52] rvest_1.0.2            lifecycle_1.0.1        DEoptimR_1.0-10
##  [55] zlibbioc_1.40.0        MASS_7.3-54            scales_1.1.1
##  [58] doSNOW_1.0.20          hms_1.1.1              parallel_4.1.2
##  [61] biomformat_1.22.0      rhdf5_2.38.0           yaml_2.3.5
##  [64] gridExtra_2.3          bdsmatrix_1.3-4        rpart_4.1-15
##  [67] latticeExtra_0.6-29    stringi_1.7.6          highr_0.9
##  [70] foreach_1.5.2          checkmate_2.0.0        rlang_1.0.1
##  [73] pkgconfig_2.0.3        bitops_1.0-7           evaluate_0.15
##  [76] Rhdf5lib_1.16.0        labeling_0.4.2         htmlwidgets_1.5.4
##  [79] tidyselect_1.1.2       plyr_1.8.6             magrittr_2.0.2
##  [82] R6_2.5.1               snow_0.4-4             generics_0.1.2
##  [85] Hmisc_4.6-0            picante_1.8.2          DBI_1.1.2
##  [88] pillar_1.7.0           haven_2.4.3            foreign_0.8-81
##  [91] withr_2.4.3            mgcv_1.8-39            abind_1.4-5
##  [94] nnet_7.3-16            survival_3.2-13        RCurl_1.98-1.6
##  [97] bayesm_3.1-4           modelr_0.1.8           crayon_1.5.0
## [100] utf8_1.2.2             tzdb_0.2.0             jpeg_0.1-9
## [103] grid_4.1.2             readxl_1.3.1           data.table_1.14.2
## [106] reprex_2.0.1           digest_0.6.29          numDeriv_2016.8-1.1
## [109] munsell_0.5.0
```