coralMicrobiomes

Matthew J. Neave

April 22, 2015

This is an R Markdown document detailing the statistical and graphical steps for reproducting the results in:

Neave, M.J., Rachmawati, R., Xun, L., Michell, C.T., Barber, P.H., Bourne, D.G., McCulloch, M.T., Apprill, A., Voolstra, C.R. A global microbiome analysis reveals that closely related corals exhibit fine-scale differences in their association with Endozoicomonas symbionts.

Load required libraries

```
library("phyloseq"); packageVersion("phyloseq")
```

```
## [1] '1.9.15'
```

```
library("ggplot2"); packageVersion("ggplot2")
```

```
## [1] '1.0.0'
```

```
library("plyr"); packageVersion("plyr")
```

```
## [1] '1.8.1'
```

```
library("vegan"); packageVersion("vegan")
```

```
## [1] '2.0.10'
```

```
library("grid"); packageVersion("grid")
```

```
## [1] '3.1.1'
```

```
library("knitr"); packageVersion("knitr")
```

```
## [1] '1.6'
```

```
library("clustsig"); packageVersion("clustsig")
```

```
## Warning: package 'clustsig' was built under R version 3.1.2
```

```
## [1] '1.1'
```

```
library('ape'); packageVersion("ape")
```

```
## [1] '3.1.4'
```

```
library('RColorBrewer'); packageVersion("RColorBrewer")
```

```
## [1] '1.0.5'
```

```
library("dunn.test"); packageVersion("dunn.test")
```

```
## Warning: package 'dunn.test' was built under R version 3.1.2
```

```
## [1] '1.2.3'
```

```
library("DESeq2"); packageVersion("DESeq2")
```

```
## Warning: package 'RcppArmadillo' was built under R version 3.1.2
```

```
## [1] '1.4.5'
```

```
setwd("./data")
opts_knit$set(root.dir = "./data")
#opts_chunk$set(tidy.opts=list(width.cutoff=80))
```

Import data

First the matrix percent file and count file generated by the minimum entropy decomposition (MED) pipeline, subsampled to 7974 reads per sample, and the accociated taxonony file

```
allShared = read.table("all.7974.matrixPercent.txt", header = T, row.names = 1)
allCounts = read.table("all.7974.matrixCount.txt", header = T, row.names = 1)
allTax = read.table("all.7974.nodeReps.nr_v119.knn.taxonomy", header = T, sep = "\t",
    row.names = 1)
```

```
## Warning: number of items read is not a multiple of the number of columns
```

```
allTax = allTax[, 2:8]
allTax = as.matrix(allTax)
```

Import the shared and taxonomy files generated in mothur for 3% and 1% pairwise similarity, in order to calculate alpha diversity measures and to compare to the MED procedure. Also import the 3% OTU file without any subsampling for alpha diversity calculations.

```
all3OTUshared = read.table("all.7974.0.03.pick.shared", header=T, row.names=2)
all3OTUshared = all3OTUshared[,3:length(all3OTUshared)]

alpha3OTUshared = read.table("all.7974.0.03.shared", header=T)
rownames(alpha3OTUshared) = alpha3OTUshared[,2]
alpha3OTUshared = alpha3OTUshared[,4:length(alpha3OTUshared)]

all1OTUshared = read.table("all.7974.0.01.pick.shared", header=T, row.names=2)
all1OTUshared = all1OTUshared[,3:length(all1OTUshared)]

all3OTUtax = read.table('all.7974.0.03.taxonomy', header=T, sep='\t', row.names=1)
all3OTUtax = all3OTUtax[,2:8]
all3OTUtax = as.matrix(all3OTUtax)

all1OTUtax = read.table('all.7974.0.01.taxonomy', header=T, sep='\t', row.names=1)
all1OTUtax = all1OTUtax[,2:8]
all1OTUtax = as.matrix(all1OTUtax)
```

Import Endozoicomonas phylogenetic tree (exported from ARB) using the APE package (Fig. 3). Also import a MED percent matrix that is slightly modified to accomodate the tree

```
endoTreeFile = read.tree(file='MEDNJ5.tree')
allSharedTree = read.table("all.7974.matrixPercent.tree.txt", header=T, row.names=1)
```

Import meta data for the samples, including metaData3.txt, which is slightly modified to accomodate heatmap sample ordering, and metaDataChem which contains additional columns of physiochemical data

```
metaFile = read.table('metaData2.MED', header=T, sep='\t', row.names=1)
metaFile3 = read.table('metaData3.txt', header=T, sep='\t', row.names=1)
metaFileChem = read.table('metaDataChem.txt', header=T, sep='\t', row.names=1)
```

Create phyloseq objects and add consistent coloring for sites

```
OTU = otu_table(allShared, taxa_are_rows = FALSE)
OTUcounts = otu_table(allCounts, taxa_are_rows = FALSE)
OTUs3 = otu_table(all3OTUshared, taxa_are_rows = FALSE)
OTUs3alpha = otu_table(alpha3OTUshared, taxa_are_rows = FALSE)
OTUs1 = otu_table(all1OTUshared, taxa_are_rows = FALSE)
OTUtree = otu_table(allSharedTree, taxa_are_rows = FALSE)

TAX = tax_table(allTax)
TAX3 = tax_table(all3OTUtax)
TAX1 = tax_table(all1OTUtax)

META = sample_data(metaFile)
METAchem = sample_data(metaFileChem)
TREE = phy_tree(endoTreeFile)

allPhylo = phyloseq(OTU, TAX, META)
countPhylo = phyloseq(OTUcounts, TAX, META)
all3OTUphylo = phyloseq(OTUs3, TAX3, META)
```

```
alpha3OTUphylo = phyloseq(OTUs3alpha, META)
all1OTUphylo = phyloseq(OTUs1, TAX1, META)
allPhyloChem = phyloseq(OTU, TAX, METAchem)
endoTree = phyloseq(OTUtree, META, TREE)

cols <- c(AmericanSamoa = "#D95F02", Indonesia = "#A6761D", MaggieIs = "#666666",
    Maldives = "#E6AB02", Micronesia = "#66A61E", Ningaloo = "#7570B3", RedSea = "#E7298A",
    other = "black")
```

Ordinations to compare MED vs pairwise OTUs

Subset samples for the two corals, remove taxa with 0s, create relative abundance and square-root sample counts

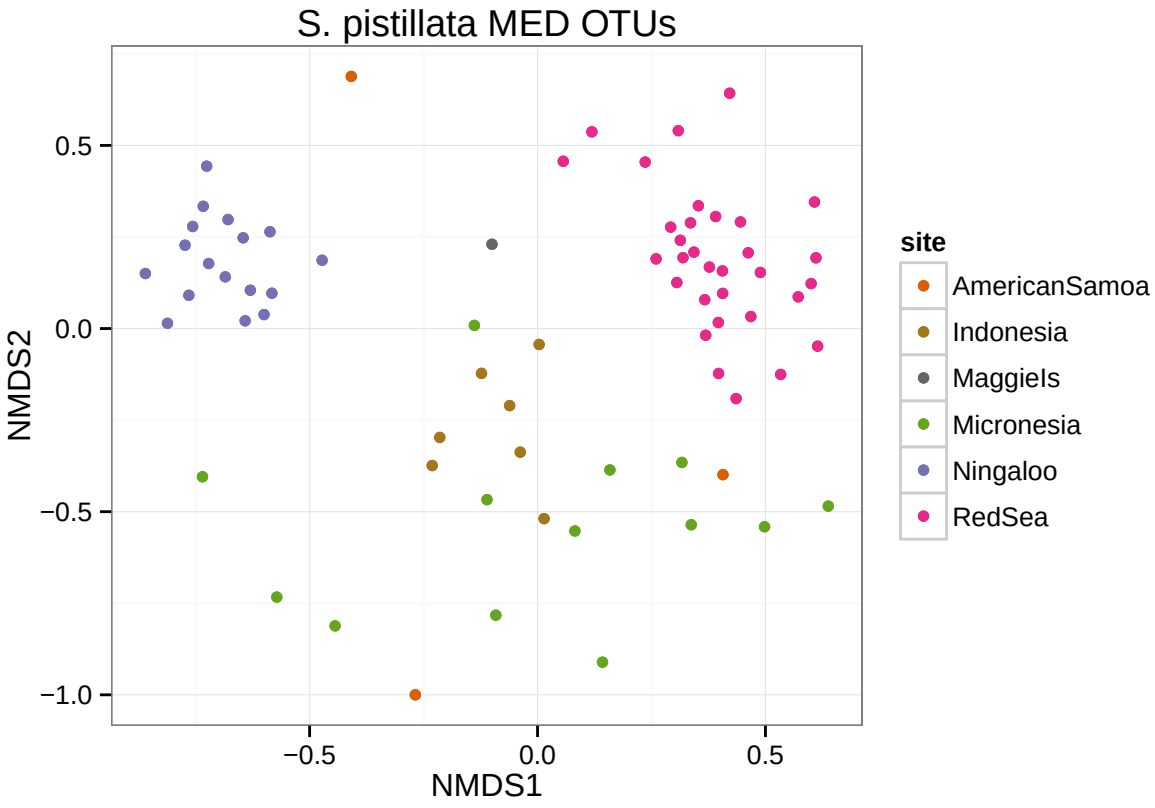
```
filter_stylo_data <- function(initial_matrix){
  initial_coral <- subset_samples(initial_matrix, species=="Stylophora pistillata")
  coral_filt = filter_taxa(initial_coral, function(x) mean(x) > 0, TRUE)
  coral_filt_rel = transform_sample_counts(coral_filt, function(x) x / sum(x) )
  coral_filt_rel_sqrt = transform_sample_counts(coral_filt_rel, function(x) sqrt(x) )
  return(coral_filt_rel_sqrt)
}

filter_pverr_data <- function(initial_matrix){
  initial_coral <- subset_samples(initial_matrix, species=="Pocillopora verrucosa")
  coral_filt = filter_taxa(initial_coral, function(x) mean(x) > 0, TRUE)
  coral_filt_rel = transform_sample_counts(coral_filt, function(x) x / sum(x) )
  coral_filt_rel_sqrt = transform_sample_counts(coral_filt_rel, function(x) sqrt(x) )
  return(coral_filt_rel_sqrt)
}

spistPhyloRelSqrt <- filter_stylo_data(allPhylo)
spist3OTUphyloRelSqrt <- filter_stylo_data(all3OTUphylo)
spist1OTUphyloRelSqrt <- filter_stylo_data(all1OTUphylo)

pverrPhyloRelSqrt <- filter_pverr_data(allPhylo)
pverr3OTUphyloRelSqrt <- filter_pverr_data(all3OTUphylo)
pverr1OTUphyloRelSqrt <- filter_pverr_data(all1OTUphylo)
```

Now do ordinations for each

```
compOrdinations <- function(sample_data, sample_name) {
    theme_set(theme_bw())
    sample_dataOrd <- ordinate(sample_data, "NMDS", "bray")
    plot_ordination(sample_data, sample_dataOrd, type = "samples", color = "site",
        title = sample_name) + geom_point(size = 2) + scale_color_manual(values = cols)
}

compOrdinations(spistPhyloRelSqrt, "S. pistillata MED OTUs")
```

```
## Run 0 stress 0.2254
## Run 1 stress 0.2459
## Run 2 stress 0.2413
```

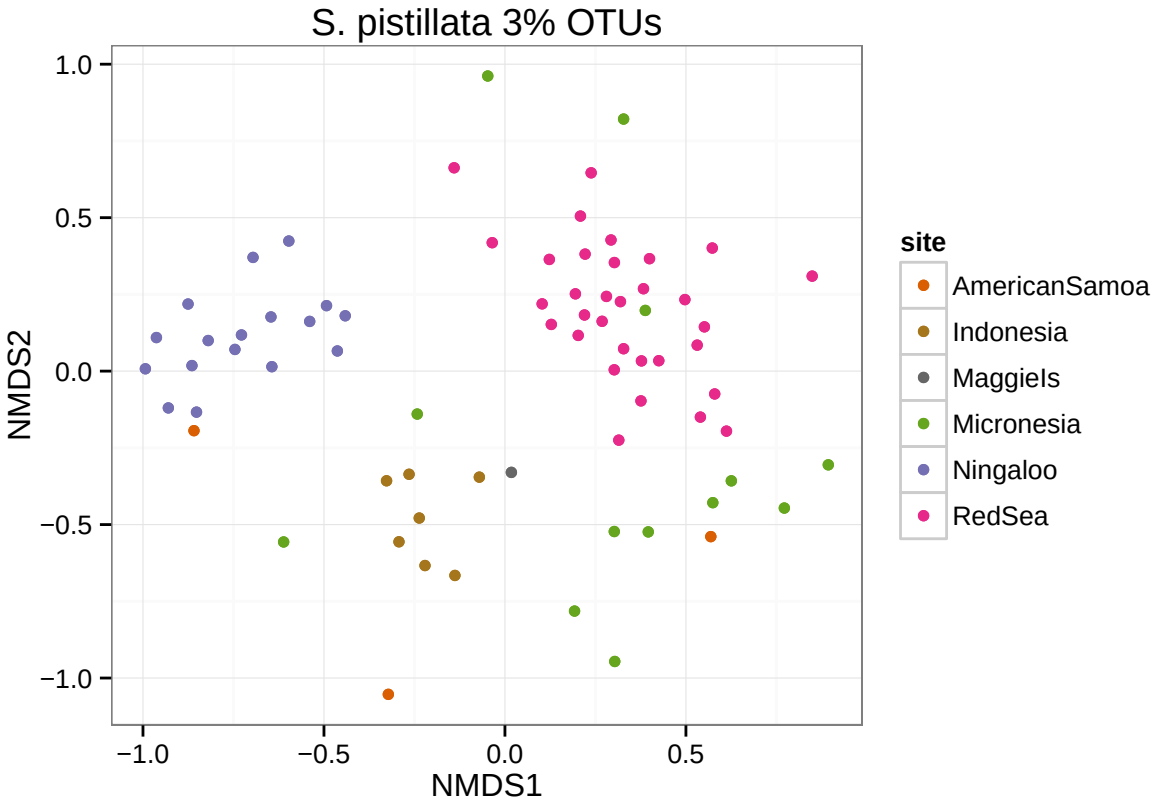
```
## Run 3 stress 0.2399
## Run 4 stress 0.2389
## Run 5 stress 0.2373
## Run 6 stress 0.2344
## Run 7 stress 0.2328
## Run 8 stress 0.2371
## Run 9 stress 0.229
## Run 10 stress 0.2273
## Run 11 stress 0.2341
## Run 12 stress 0.2302
## Run 13 stress 0.2352
## Run 14 stress 0.2288
## Run 15 stress 0.2503
## Run 16 stress 0.2381
## Run 17 stress 0.2486
## Run 18 stress 0.2409
## Run 19 stress 0.2242
## ... New best solution
## ... procrustes: rmse 0.02453  max resid 0.1974
## Run 20 stress 0.2292
```




```
compOrdinations(spist3OTUphyloRelSqrt, "S. pistillata 3% OTUs")
```

```
## Run 0 stress 0.2273
## Run 1 stress 0.235
## Run 2 stress 0.248
## Run 3 stress 0.2386
```

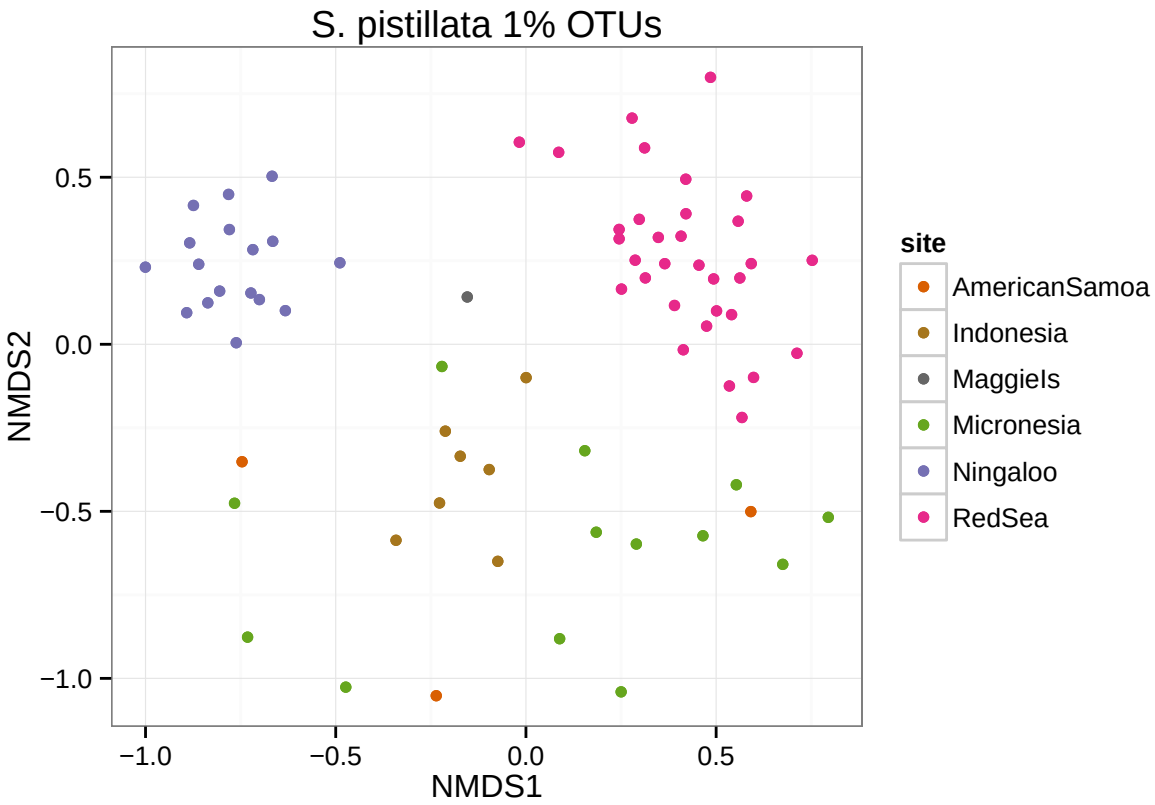
```
## Run 4 stress 0.2395
## Run 5 stress 0.2663
## Run 6 stress 0.2542
## Run 7 stress 0.2273
## ... New best solution
## ... procrustes: rmse 0.0004645  max resid 0.003106
## *** Solution reached
```




```
compOrdinations(spist1OTUphyloRelSqrt, "S. pistillata 1% OTUs")
```

```
## Run 0 stress 0.2225
## Run 1 stress 0.2575
## Run 2 stress 0.2512
## Run 3 stress 0.2274
## Run 4 stress 0.2645
## Run 5 stress 0.2512
## Run 6 stress 0.2366
## Run 7 stress 0.2274
## Run 8 stress 0.2351
## Run 9 stress 0.2532
## Run 10 stress 0.2304
## Run 11 stress 0.2582
## Run 12 stress 0.2338
## Run 13 stress 0.2318
## Run 14 stress 0.2331
## Run 15 stress 0.2399
## Run 16 stress 0.2586
```

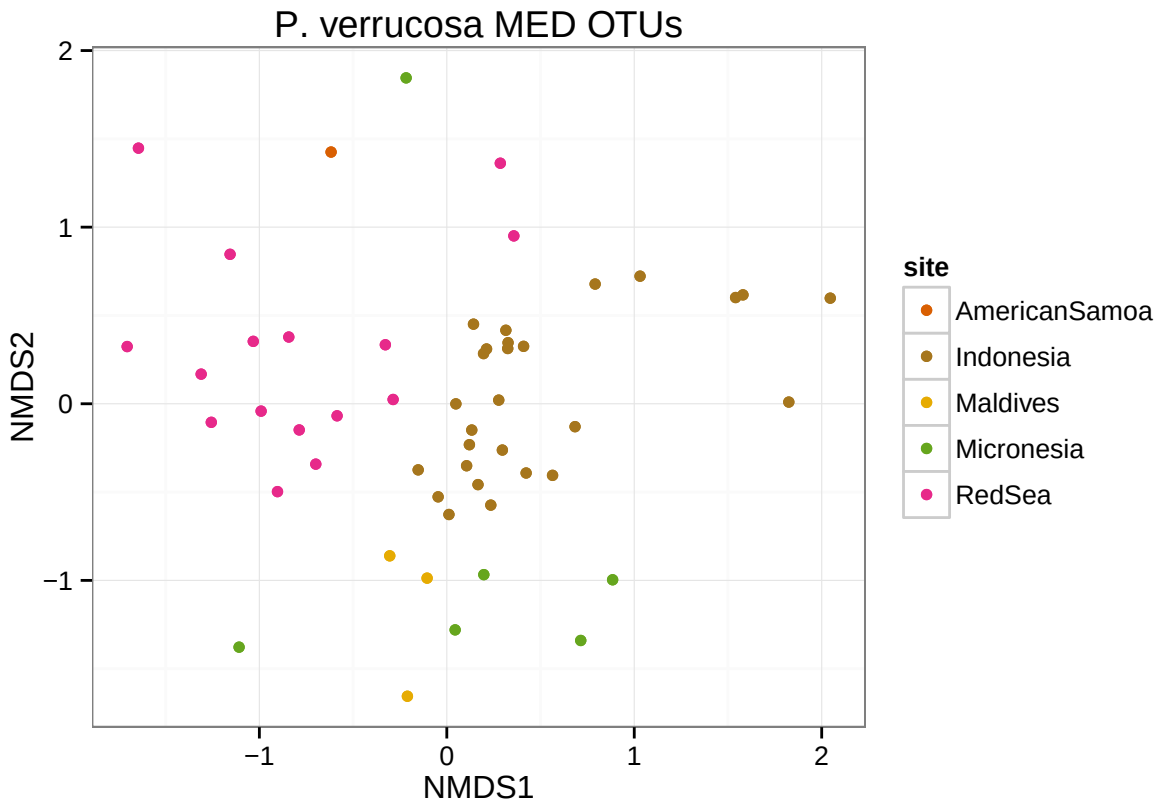
```
## Run 17 stress 0.2571
## Run 18 stress 0.2417
## Run 19 stress 0.2287
## Run 20 stress 0.2559
```




```
compOrdinations(pverrPhyloRelSqrt, "P. verrucosa MED OTUs")
```

```
## Run 0 stress 0.2437
## Run 1 stress 0.2485
## Run 2 stress 0.271
## Run 3 stress 0.2362
## ... New best solution
## ... procrustes: rmse 0.09863  max resid 0.3381
## Run 4 stress 0.2446
## Run 5 stress 0.2349
## ... New best solution
## ... procrustes: rmse 0.07606  max resid 0.353
## Run 6 stress 0.2133
## ... New best solution
## ... procrustes: rmse 0.08037  max resid 0.3474
## Run 7 stress 0.2259
## Run 8 stress 0.2227
## Run 9 stress 0.2438
## Run 10 stress 0.2136
## ... procrustes: rmse 0.02494  max resid 0.146
## Run 11 stress 0.2247
## Run 12 stress 0.2454
```

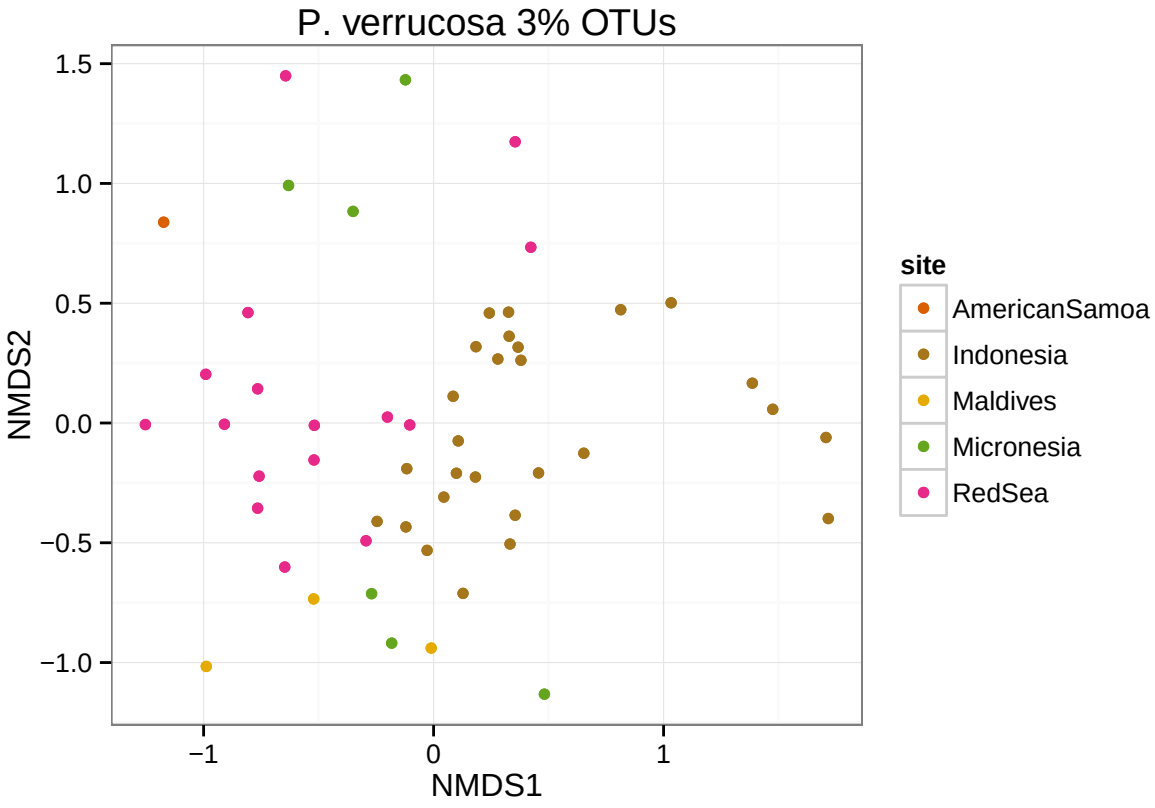
```
## Run 13 stress 0.2186
## Run 14 stress 0.2393
## Run 15 stress 0.2393
## Run 16 stress 0.2206
## Run 17 stress 0.2315
## Run 18 stress 0.2264
## Run 19 stress 0.2572
## Run 20 stress 0.239
```




```
compOrdinations(pverr3OTUphyloRelSqrt, "P. verrucosa 3% OTUs")
```

```
## Run 0 stress 0.2409
## Run 1 stress 0.2354
## ... New best solution
## ... procrustes: rmse 0.09003  max resid 0.3035
## Run 2 stress 0.2482
## Run 3 stress 0.2598
## Run 4 stress 0.2349
## ... New best solution
## ... procrustes: rmse 0.09954  max resid 0.4044
## Run 5 stress 0.23
## ... New best solution
## ... procrustes: rmse 0.1085  max resid 0.3136
## Run 6 stress 0.2283
## ... New best solution
## ... procrustes: rmse 0.09872  max resid 0.2776
## Run 7 stress 0.224
```

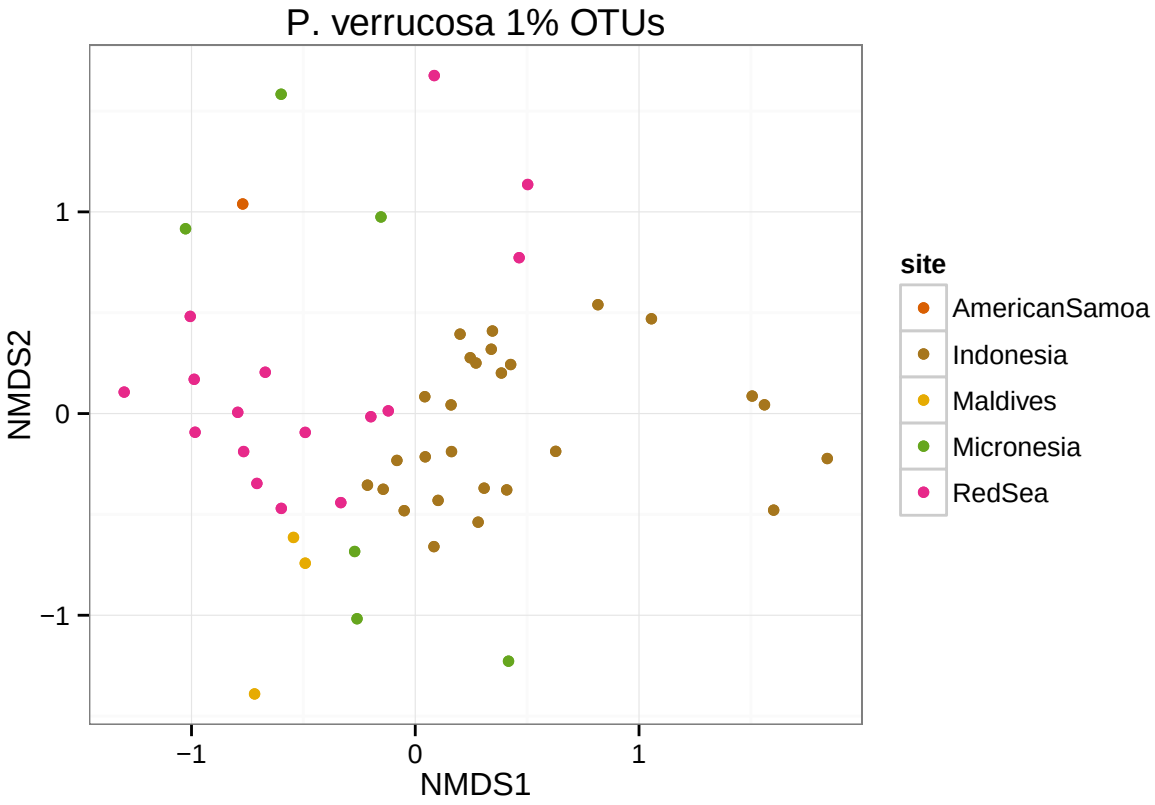
```
## ... New best solution
## ... procrustes: rmse 0.0909  max resid 0.2662
## Run 8 stress 0.2289
## Run 9 stress 0.2407
## Run 10 stress 0.2288
## Run 11 stress 0.2238
## ... New best solution
## ... procrustes: rmse 0.04183  max resid 0.2614
## Run 12 stress 0.2593
## Run 13 stress 0.2852
## Run 14 stress 0.2277
## Run 15 stress 0.227
## Run 16 stress 0.2358
## Run 17 stress 0.2468
## Run 18 stress 0.2353
## Run 19 stress 0.2316
## Run 20 stress 0.4043
```




```
compOrdinations(pverr1OTUphyloRelSqrt, "P. verrucosa 1% OTUs")
```

```
## Run 0 stress 0.234
## Run 1 stress 0.2289
## ... New best solution
## ... procrustes: rmse 0.08099  max resid 0.2296
## Run 2 stress 0.2458
## Run 3 stress 0.217
## ... New best solution
```

```
## ... procrustes: rmse 0.09259  max resid 0.4139
## Run 4 stress 0.2297
## Run 5 stress 0.2492
## Run 6 stress 0.2473
## Run 7 stress 0.228
## Run 8 stress 0.2341
## Run 9 stress 0.2263
## Run 10 stress 0.2306
## Run 11 stress 0.2187
## Run 12 stress 0.2263
## Run 13 stress 0.2331
## Run 14 stress 0.2356
## Run 15 stress 0.2355
## Run 16 stress 0.2172
## ... procrustes: rmse 0.04413  max resid 0.297
## Run 17 stress 0.2188
## Run 18 stress 0.2283
## Run 19 stress 0.2181
## Run 20 stress 0.2322
```



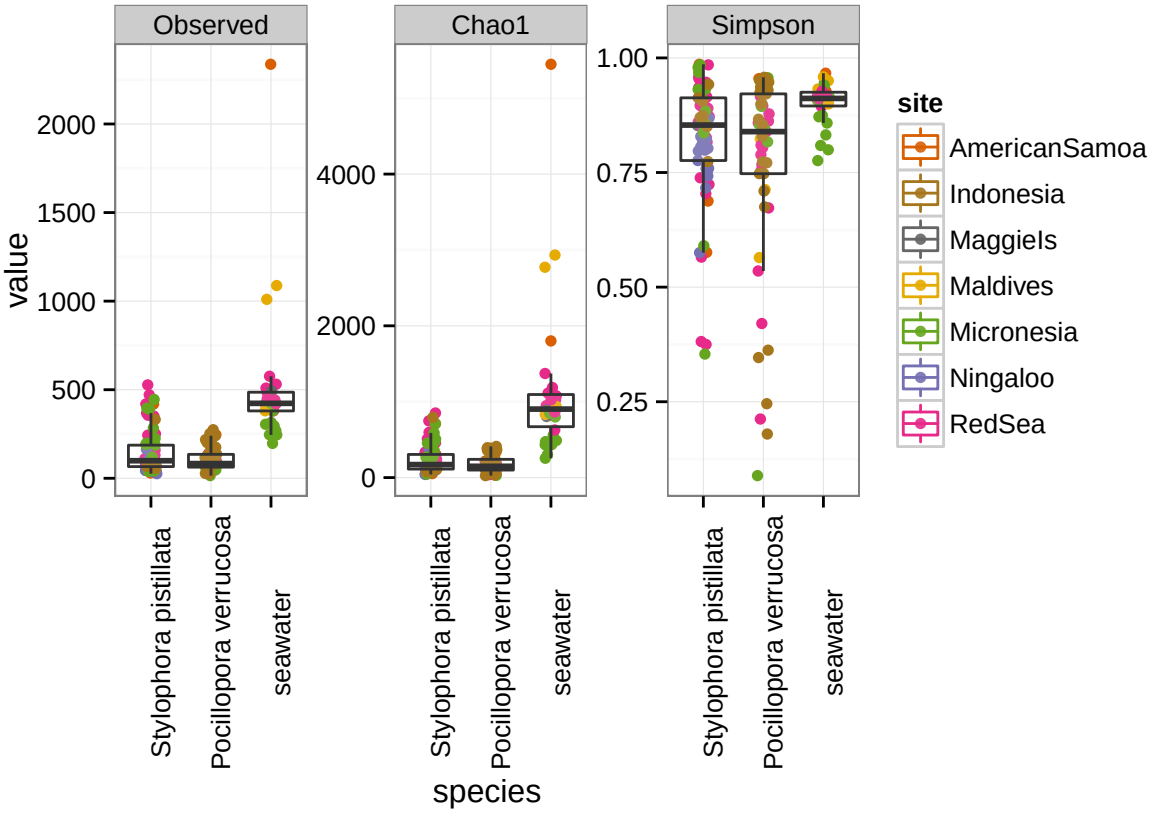
Alpha diversity measures

First subset the corals, then plot using phyloseq and ggplot2

Note: I'll use unsubampled 3% pairwise OTUs for calculation of alpha diversity measures as this will make them more comparable to other studies, plus the MED pipeline is has not yet implemented alpha diversity

```
allAlphaTmp <- subset_samples(alpha3OTUphylo, species == "seawater")
allAlphaTmp2 <- subset_samples(alpha3OTUphylo, species == "Stylophora pistillata")
allAlphaTmp3 <- subset_samples(alpha3OTUphylo, species == "Pocillopora verrucosa")
allAlpha2 <- merge_phyloseq(allAlphaTmp, allAlphaTmp2, allAlphaTmp3)

allAlphaPlot2 <- plot_richness(allAlpha2, x = "species", measures = c("Chao1", "Simpson",
    "observed"), color = "site", sortby = "Chao1")

ggplot(data = allAlphaPlot2$data) + geom_point(aes(x = species, y = value, color = site),
    position = position_jitter(width = 0.1, height = 0)) + geom_boxplot(aes(x = species,
    y = value, color = NULL), alpha = 0.1, outlier.shape = NA) + scale_color_manual(values = cols) +
    theme(axis.text.x = element_text(angle = 90)) + facet_wrap(~variable, scales = "free_y") +
    scale_x_discrete(limits = c("Stylophora pistillata", "Pocillopora verrucosa",
        "seawater"))
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
## Warning: Removed 6 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
## Warning: Removed 7 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
## Warning: Removed 4 rows containing missing values (geom_point).
```



Check for significant differences in the alpha diversity measures using a kruskal-wallis test and a dunn post-hoc test to check which specific groups were different

```
alphaObserved = estimate_richness(allAlpha2, measures="Observed")
alphaSimpson = estimate_richness(allAlpha2, measures="Simpson")
alphaChao = estimate_richness(allAlpha2, measures="Chao1")

alpha.stats <- cbind(alphaObserved, sample_data(allAlpha2))
alpha.stats2 <- cbind(alpha.stats, alphaSimpson)
alpha.stats3 <- cbind(alpha.stats2, alphaChao)

kruskal.test(Observed~species, data = alpha.stats3)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Observed by species
## Kruskal-Wallis chi-squared = 61.88, df = 2, p-value = 3.662e-14
```

```
dunn.test(alpha.stats3$Observed, alpha.stats3$species, method="bonferroni")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 61.8764, df = 2, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |   Pocillop   seawater
## ---------+----------------------
## seawater |   7.510384
##          |     0.0000
##          |
## Stylopho |   1.357184  -6.783011
##          |     0.2621     0.0000
```

```
kruskal.test(Simpson~species, data = alpha.stats3)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Simpson by species
## Kruskal-Wallis chi-squared = 12.25, df = 2, p-value = 0.002193
```

```
dunn.test(alpha.stats3$Simpson, alpha.stats3$species, method="bonferroni")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 12.2453, df = 2, p-value = 0
##
##
```

```
##                             Comparison of x by group
##                                   (Bonferroni)
## Col Mean-|
## Row Mean |   Pocillop   seawater
## ---------+----------------------
## seawater |   3.397898
##          |     0.0010
##          |
## Stylopho |   0.811204  -2.904738
##          |     0.6259     0.0055
```

```
kruskal.test(Chao1~species, data = alpha.stats3)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Chao1 by species
## Kruskal-Wallis chi-squared = 64.31, df = 2, p-value = 1.086e-14
```

```
dunn.test(alpha.stats3$Chao1, alpha.stats3$species, method="bonferroni")
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 64.3067, df = 2, p-value = 0
##
##
##                             Comparison of x by group
##                                   (Bonferroni)
## Col Mean-|
## Row Mean |   Pocillop   seawater
## ---------+----------------------
## seawater |   7.581725
##          |     0.0000
##          |
## Stylopho |   1.146749  -7.033279
##          |     0.3772     0.0000
```

In each case, the seawater was signficiantly different to the corals, while the corals were not different to each other. This suggests the corals have a more 'selective' community of microbes compared to the surrounding seawater.

Similarity Profile Analysis (SIMPROF)

This will show how the samples cluster without any a priori assumptions regarding sample origin

Need to import the shared file containing just spist OTUs, then calcualte the simprof clusters based on the braycurtis metric.

```
# spist <- subset_samples(allPhylo, species=='Stylophora pistillata') spistShared
# = otu_table(spist) class(spistShared) <- 'numeric' spistSIMPROF <-
# simprof(spistShared, num.expected=1000, num.simulated=99,
```

```
# method.cluster='average', method.distance='braycurtis',
# method.transform='squareroot', alpha=0.05, sample.orientation='row',
# silent=TRUE) simprof.plot(spistSIMPROF, leafcolors=NA, plot=TRUE, fill=TRUE,
# leaflab='perpendicular', siglinetype=1) pVerr <- subset_samples(allPhylo,
# species=='Pocillopora verrucosa') pVerrShared = otu_table(pVerr)
# class(pVerrShared) <- 'numeric' pVerrSIMPROF <- simprof(pVerrShared,
# num.expected=1000, num.simulated=99, method.cluster='average',
# method.distance='braycurtis', method.transform='squareroot', alpha=0.05,
# sample.orientation='row', silent=TRUE) simprof.plot(pVerrSIMPROF,
# leafcolors=NA, plot=TRUE, fill=TRUE, leaflab='perpendicular', siglinetype=1)
```

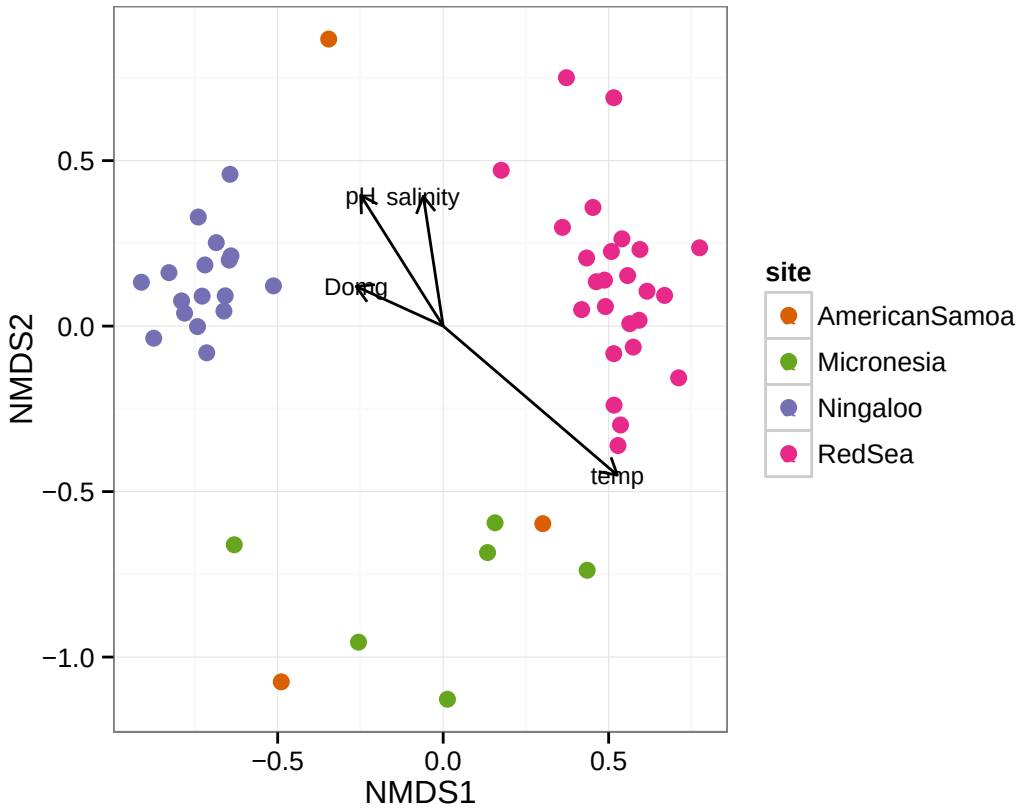
Chemical and biological correlations

Use the envfit function from the Vegan package to test if any environmental variables are significantly correlated with microbiome differences in the corals

```
draw_envfit_ord <- function(coral_chem, env_data) {
    chemNoNA <- na.omit(metaFileChem[sample_names(coral_chem), env_data])
    coralNoNA <- prune_samples(rownames(chemNoNA), coral_chem)

    theme_set(theme_bw())
    coralNoNAOrd <- ordinate(coralNoNA, "NMDS", "bray")
    coralNoNAOrdPlot <- plot_ordination(coralNoNA, coralNoNAOrd, type = "samples",
        color = "site") + geom_point(size = 3) + scale_color_manual(values = c(cols))

    # get points for ggplot
    pointsNoNA <- coralNoNAOrd$points[rownames(chemNoNA), ]
    chemFit <- envfit(pointsNoNA, env = chemNoNA, na.rm = TRUE)
    print(chemFit)
    chemFit.scores <- as.data.frame(scores(chemFit, display = "vectors"))
    chemFit.scores <- cbind(chemFit.scores, Species = rownames(chemFit.scores))

    # create arrow info
    chemNames <- rownames(chemFit.scores)
    arrowmap <- aes(xend = MDS1, yend = MDS2, x = 0, y = 0, shape = NULL, color = NULL)
    labelmap <- aes(x = MDS1, y = MDS2 + 0.04, shape = NULL, color = NULL, size = 1.5,
        label = chemNames)
    arrowhead = arrow(length = unit(0.25, "cm"))

    # note: had to use aes_string to get ggplot to recognize variables
    coralNoNAOrdPlot + coord_fixed() + geom_segment(arrowmap, size = 0.5, data = chemFit.scores,
        color = "black", arrow = arrowhead, show_guide = FALSE) + geom_text(aes_string(x = "MDS1",
        y = "MDS2", shape = NULL, color = NULL, size = 1.5, label = "Species"), size = 3,
        data = chemFit.scores)
}

waterQual <- c("temp", "salinity", "Domg", "pH")
nutrients <- c("PO4", "N.N", "silicate", "NO2", "NH4")
FCM <- c("prok", "syn", "peuk", "pe.peuk", "Hbact")

spistChem <- subset_samples(allPhyloChem, species == "Stylophora pistillata")
pverrChem <- subset_samples(allPhyloChem, species == "Pocillopora verrucosa")
```

```
draw_envfit_ord(spistChem, waterQual)
```

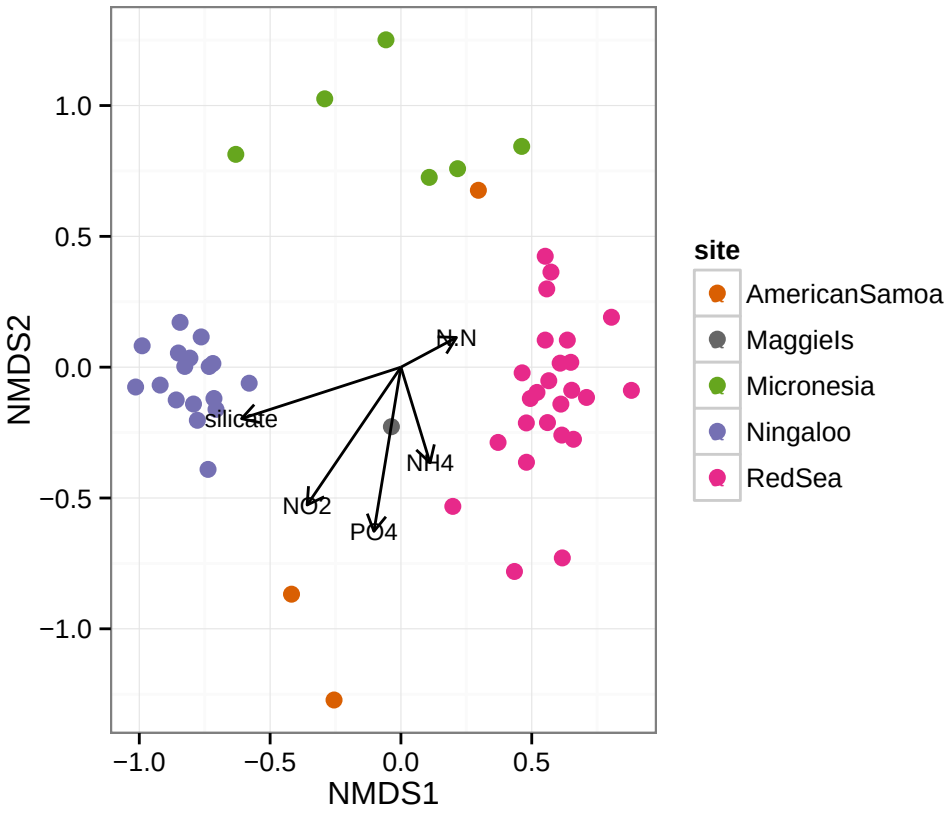
```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.189
## Run 1 stress 0.1729
## ... New best solution
## ... procrustes: rmse 0.04878  max resid 0.2672
## Run 2 stress 0.2191
## Run 3 stress 0.2
## Run 4 stress 0.2133
## Run 5 stress 0.2092
## Run 6 stress 0.2003
## Run 7 stress 0.1853
## Run 8 stress 0.1848
## Run 9 stress 0.2156
## Run 10 stress 0.2121
## Run 11 stress 0.1828
## Run 12 stress 0.2202
## Run 13 stress 0.2
## Run 14 stress 0.1931
## Run 15 stress 0.1848
## Run 16 stress 0.2178
## Run 17 stress 0.2018
## Run 18 stress 0.2141
## Run 19 stress 0.1885
## Run 20 stress 0.1941
##
## ***VECTORS
##
##            MDS1   MDS2   r2 Pr(>r)
## temp      0.760 -0.650 0.48  0.001 ***
## salinity -0.151  0.989 0.16  0.020 *
## Domg     -0.910  0.415 0.08  0.122
## pH       -0.533  0.846 0.22  0.004 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```




```
draw_envfit_ord(spistChem, nutrients)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1927
## Run 1 stress 0.1979
## Run 2 stress 0.1848
## ... New best solution
## ... procrustes: rmse 0.0305  max resid 0.1973
## Run 3 stress 0.1837
## ... New best solution
## ... procrustes: rmse 0.01026  max resid 0.06491
## Run 4 stress 0.2098
## Run 5 stress 0.2
## Run 6 stress 0.2092
## Run 7 stress 0.1963
## Run 8 stress 0.1874
## Run 9 stress 0.1957
## Run 10 stress 0.2218
## Run 11 stress 0.2114
## Run 12 stress 0.1888
## Run 13 stress 0.1955
## Run 14 stress 0.2078
## Run 15 stress 0.1821
## ... New best solution
## ... procrustes: rmse 0.07277  max resid 0.4077
## Run 16 stress 0.2004
```

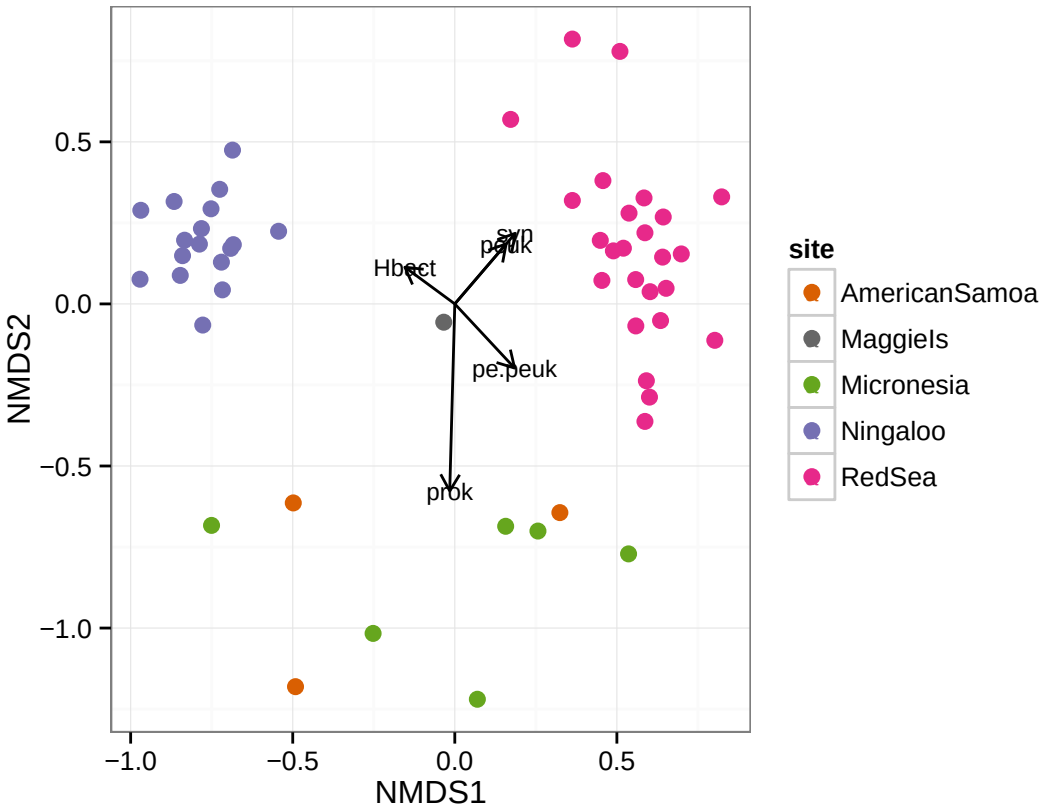
```
## Run 17 stress 0.1837
## Run 18 stress 0.2134
## Run 19 stress 0.2139
## Run 20 stress 0.1955
##
## ***VECTORS
##
##             MDS1   MDS2   r2 Pr(>r)
## PO4      -0.162 -0.987 0.40  0.001 ***
## N.N       0.882  0.472 0.06  0.197
## silicate -0.952 -0.307 0.41  0.001 ***
## NO2      -0.563 -0.827 0.41  0.001 ***
## NH4       0.292 -0.957 0.15  0.016 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```




```
draw_envfit_ord(spistChem, FCM)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.1927
## Run 1 stress 0.2191
## Run 2 stress 0.2193
## Run 3 stress 0.1971
## Run 4 stress 0.2141
## Run 5 stress 0.1848
```

```
## ... New best solution
## ... procrustes: rmse 0.03064  max resid 0.1973
## Run 6 stress 0.2075
## Run 7 stress 0.2075
## Run 8 stress 0.1936
## Run 9 stress 0.1922
## Run 10 stress 0.2043
## Run 11 stress 0.216
## Run 12 stress 0.196
## Run 13 stress 0.206
## Run 14 stress 0.2108
## Run 15 stress 0.1943
## Run 16 stress 0.1949
## Run 17 stress 0.2147
## Run 18 stress 0.2017
## Run 19 stress 0.2089
## Run 20 stress 0.2179
##
## ***VECTORS
##
##           MDS1   MDS2   r2 Pr(>r)
## prok    -0.027 -1.000 0.33  0.002 **
## syn      0.650  0.760 0.08  0.092 .
## peuk     0.650  0.760 0.06  0.214
## pe.peuk  0.682 -0.731 0.07  0.163
## Hbact   -0.805  0.593 0.04  0.426
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```



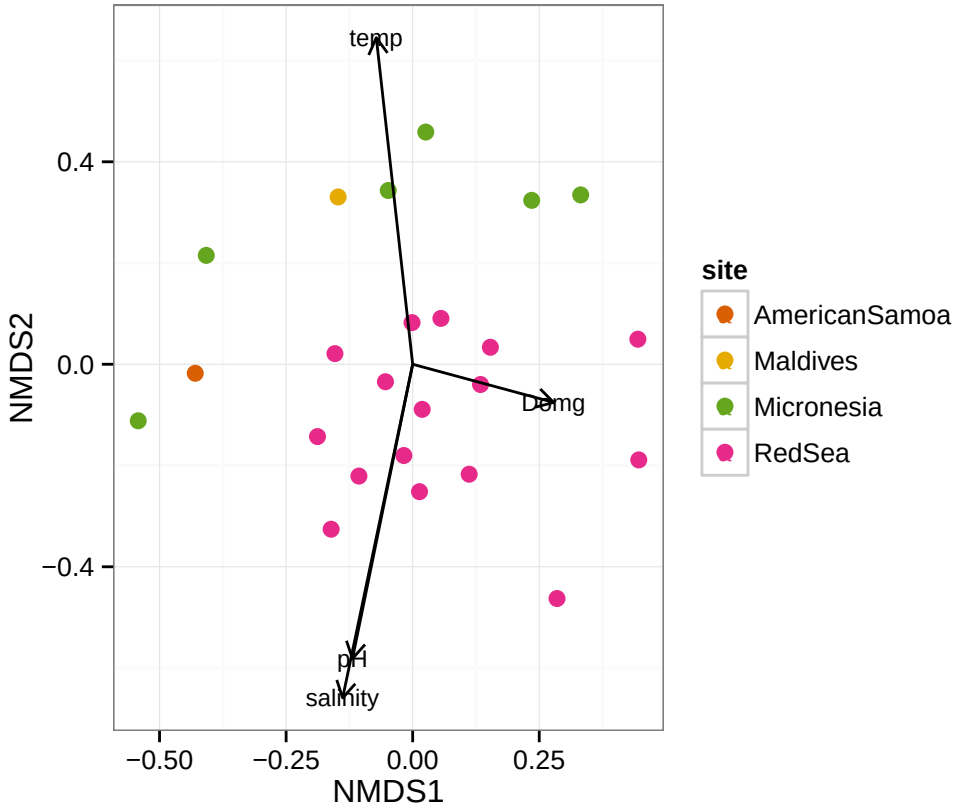

```
draw_envfit_ord(pverrChem, waterQual)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.2599
## Run 1 stress 0.2639
## Run 2 stress 0.2739
## Run 3 stress 0.2583
## ... New best solution
## ... procrustes: rmse 0.1098  max resid 0.2726
## Run 4 stress 0.263
## Run 5 stress 0.2699
## Run 6 stress 0.2583
## ... New best solution
## ... procrustes: rmse 0.1761  max resid 0.4596
## Run 7 stress 0.259
## Run 8 stress 0.2535
## ... New best solution
## ... procrustes: rmse 0.1793  max resid 0.436
## Run 9 stress 0.2579
## Run 10 stress 0.2545
## Run 11 stress 0.247
## ... New best solution
## ... procrustes: rmse 0.1713  max resid 0.384
## Run 12 stress 0.2708
## Run 13 stress 0.2762
## Run 14 stress 0.2536
```

```
## Run 15 stress 0.2559
## Run 16 stress 0.2555
## Run 17 stress 0.263
## Run 18 stress 0.2692
## Run 19 stress 0.2582
## Run 20 stress 0.2581
```

```
## Warning: skipping half-change scaling: too few points below threshold
```

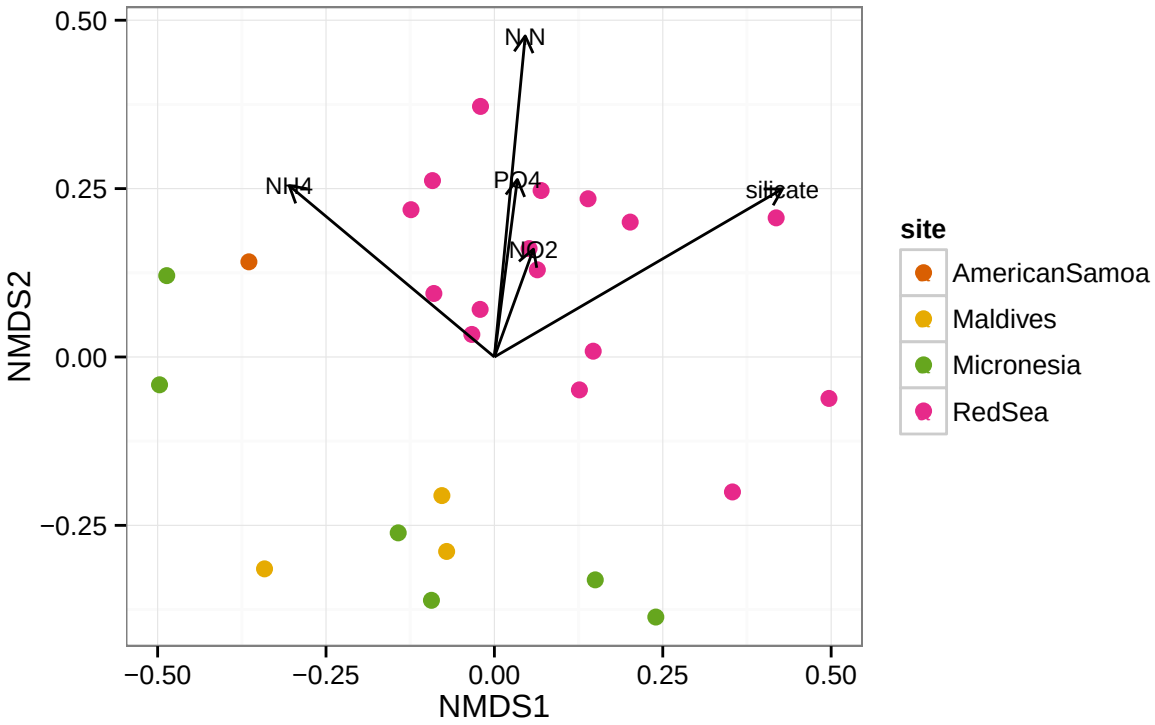
```
##
## ***VECTORS
##
##            MDS1   MDS2   r2 Pr(>r)
## temp     -0.111  0.994 0.42  0.006 **
## salinity -0.205 -0.979 0.45  0.007 **
## Domg      0.965 -0.261 0.08  0.369
## pH       -0.200 -0.980 0.35  0.010 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```




```
draw_envfit_ord(pverrChem, nutrients)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.262
```

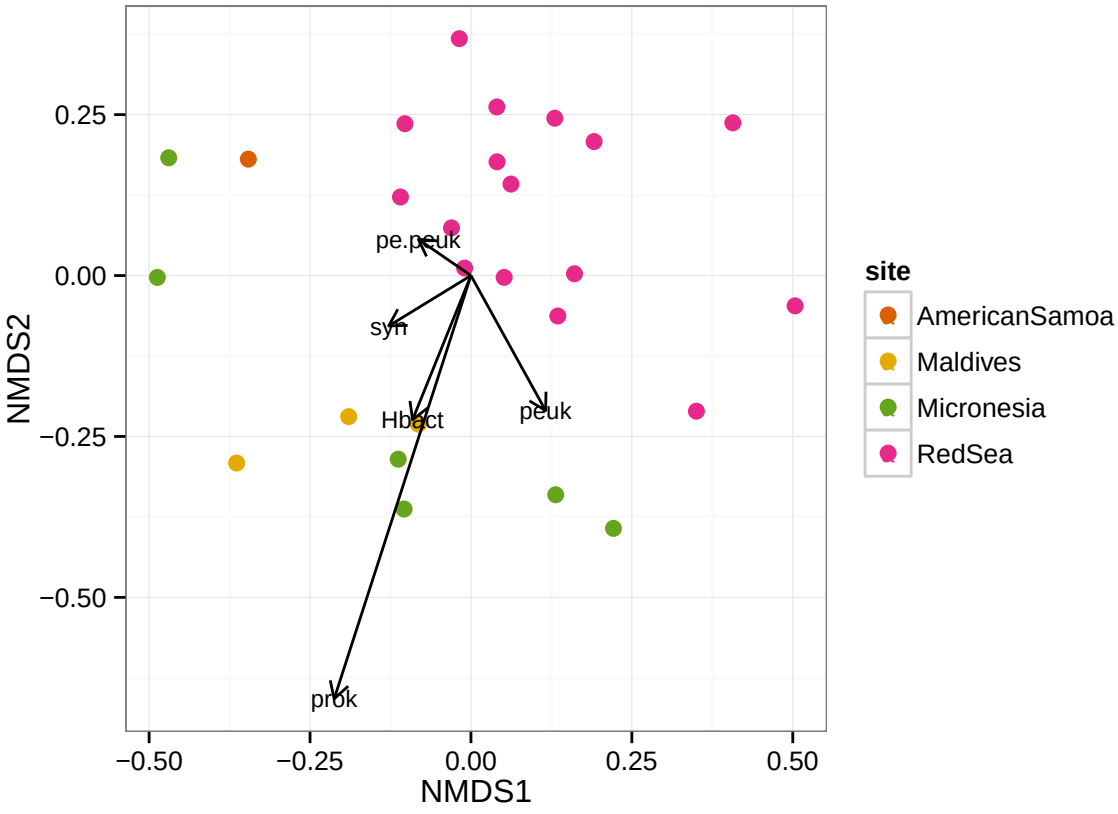
```
## Run 1 stress 0.2529
## ... New best solution
## ... procrustes: rmse 0.1795  max resid 0.3817
## Run 2 stress 0.2605
## Run 3 stress 0.276
## Run 4 stress 0.2585
## Run 5 stress 0.2703
## Run 6 stress 0.282
## Run 7 stress 0.2666
## Run 8 stress 0.2939
## Run 9 stress 0.2439
## ... New best solution
## ... procrustes: rmse 0.08096  max resid 0.2958
## Run 10 stress 0.2542
## Run 11 stress 0.2778
## Run 12 stress 0.2645
## Run 13 stress 0.2743
## Run 14 stress 0.2654
## Run 15 stress 0.2766
## Run 16 stress 0.2719
## Run 17 stress 0.2748
## Run 18 stress 0.2627
## Run 19 stress 0.2657
## Run 20 stress 0.2907

## Warning: skipping half-change scaling: too few points below threshold

##
## ***VECTORS
##
##            MDS1   MDS2   r2 Pr(>r)
## PO4       0.128  0.992 0.07  0.437
## N.N       0.096  0.995 0.23  0.038 *
## silicate  0.864  0.504 0.25  0.027 *
## NO2       0.340  0.940 0.03  0.711
## NH4      -0.768  0.641 0.16  0.135
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```




```
draw_envfit_ord(pverrChem, FCM)
```

```
## Square root transformation
## Wisconsin double standardization
## Run 0 stress 0.262
## Run 1 stress 0.2789
## Run 2 stress 0.2602
## ... New best solution
## ... procrustes: rmse 0.1728  max resid 0.4041
## Run 3 stress 0.2598
## ... New best solution
## ... procrustes: rmse 0.15  max resid 0.4096
## Run 4 stress 0.2633
## Run 5 stress 0.2603
## Run 6 stress 0.2684
## Run 7 stress 0.2588
## ... New best solution
## ... procrustes: rmse 0.1693  max resid 0.3954
## Run 8 stress 0.2437
## ... New best solution
## ... procrustes: rmse 0.1479  max resid 0.3598
## Run 9 stress 0.244
## ... procrustes: rmse 0.01929  max resid 0.08613
## Run 10 stress 0.2873
## Run 11 stress 0.2627
## Run 12 stress 0.2963
## Run 13 stress 0.255
## Run 14 stress 0.2608
## Run 15 stress 0.2957
## Run 16 stress 0.2736
```

```
## Run 17 stress 0.2652
## Run 18 stress 0.2667
## Run 19 stress 0.2906
## Run 20 stress 0.2621

## Warning: skipping half-change scaling: too few points below threshold

##
## ***VECTORS
##
##           MDS1   MDS2   r2 Pr(>r)
## prok    -0.308 -0.952 0.48  0.001 ***
## syn     -0.853 -0.522 0.02  0.785
## peuk     0.484 -0.875 0.06  0.511
## pe.peuk -0.823  0.568 0.01  0.886
## Hbact   -0.375 -0.927 0.06  0.492
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## P values based on 999 permutations.
```



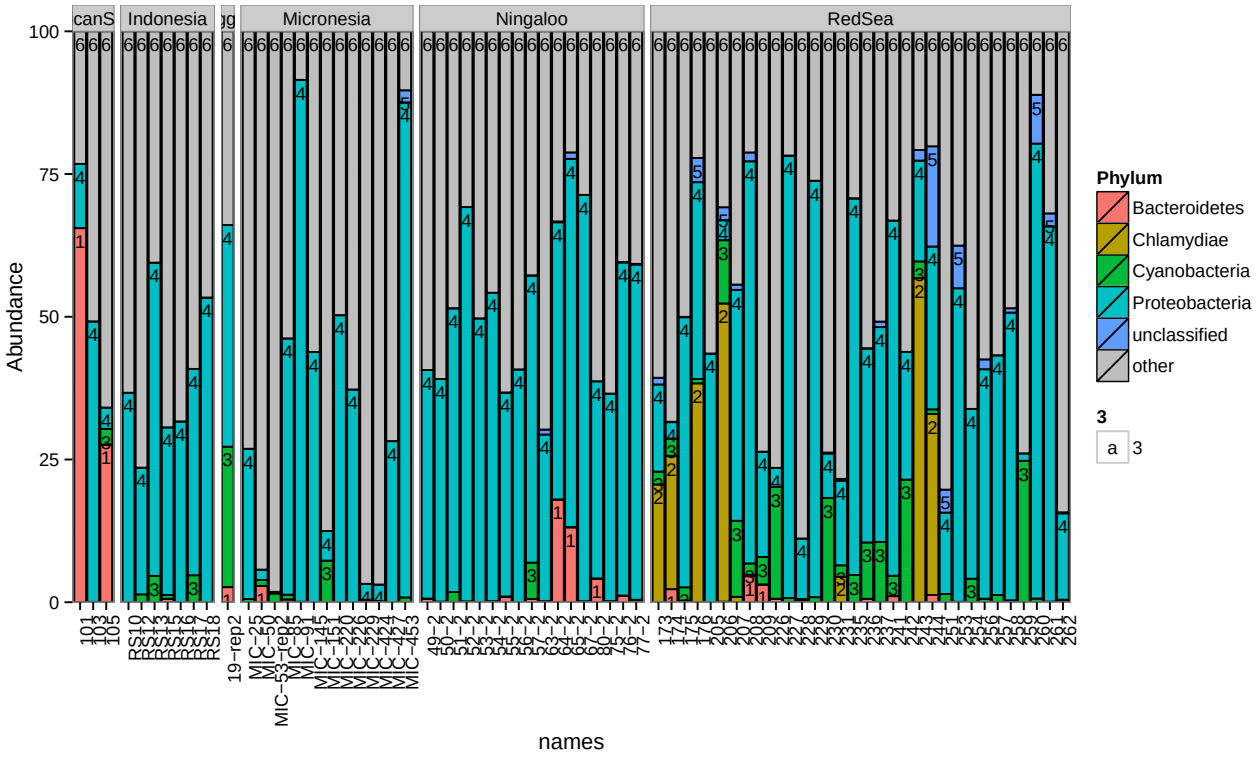
Taxonomic barcharts of bacteria in the corals and seawaters, and core microbiome members

```
# define a function to draw barcharts at a specific taxonomic level also need to
# create my own ggplot colors then replace the last one ('other' column) with
# gray
```

```
gg_color_hue <- function(n) {
    hues = seq(15, 375, length = n + 1)
    hcl(h = hues, l = 65, c = 100)[1:n]
}

draw_barcharts <- function(coral_species, tax_level) {

    coralFiltGlom <- tax_glom(coral_species, taxrank = tax_level)
    physeqdf <- psmelt(coralFiltGlom)

    # get total abundance so can make an 'other' column had to add ^ and $ characters
    # to make sure grep matches whole word

    physeqdfOther <- physeqdf

    for (j in unique(physeqdf$Sample)) {
        jFirst = paste("^", j, sep = "")
        jBoth = paste(jFirst, "$", sep = "")
        rowNumbers = grep(jBoth, physeqdf$Sample)
        otherValue = 100 - sum(physeqdf[rowNumbers, "Abundance"])
        newRow = (physeqdf[rowNumbers, ])[1, ]
        newRow[, tax_level] = "other"
        newRow[, "Abundance"] = otherValue
        physeqdfOther <- rbind(physeqdfOther, newRow)
    }

    ggCols <- gg_color_hue(length(unique(physeqdfOther[, tax_level])))
    ggCols <- head(ggCols, n = -1)

    # add names and numbers for easier referencing
    physeqdfOther$names <- factor(physeqdfOther$Sample, levels = rownames(metaFile),
        ordered = TRUE)
    physeqdfOther$tax_level_num <- as.numeric(physeqdfOther[, tax_level])

    theme_set(theme_bw())
    ggplot(physeqdfOther, aes_string(x = "names", y = "Abundance", fill = tax_level,
        order = tax_level)) + geom_bar(stat = "identity", colour = "black") + geom_text(position = "sta
        aes(label = ifelse(Abundance > 2, tax_level_num, ""), vjust = 1.5, size = 3)) +
        scale_fill_manual(values = c(ggCols, "gray")) + scale_y_continuous(expand = c(0,
        0), limits = c(0, 100)) + facet_grid(~site, scales = "free", space = "free_x") +
        theme(axis.text.x = element_text(angle = 90, hjust = 1))
}

# subset coral samples, create names factor for label ordering and filter so the
# graph isn't too full
spist <- subset_samples(allPhylo, species == "Stylophora pistillata")
sample_data(spist)$names <- factor(sample_names(spist), levels = rownames(metaFile),
    ordered = TRUE)
spistFilt = filter_taxa(spist, function(x) mean(x) > 0.8, TRUE)
draw_barcharts(spistFilt, "Phylum")  # 0.2
```

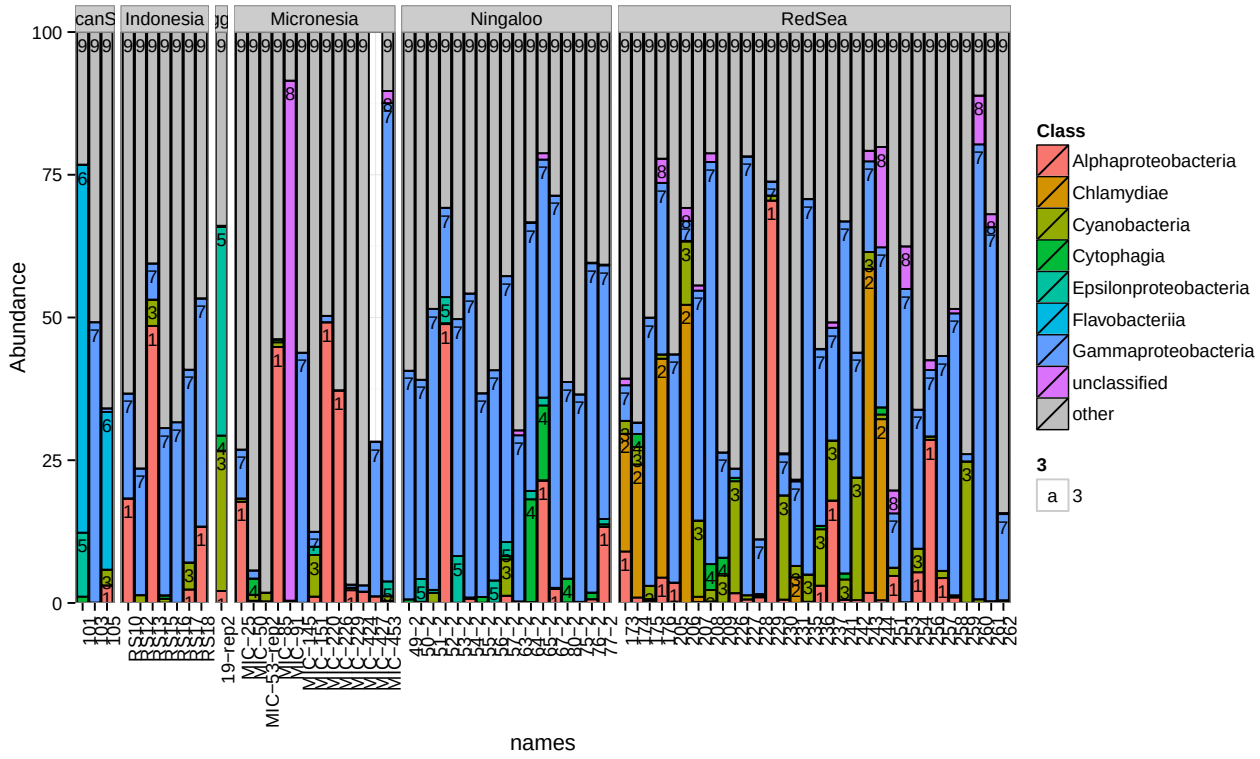
```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



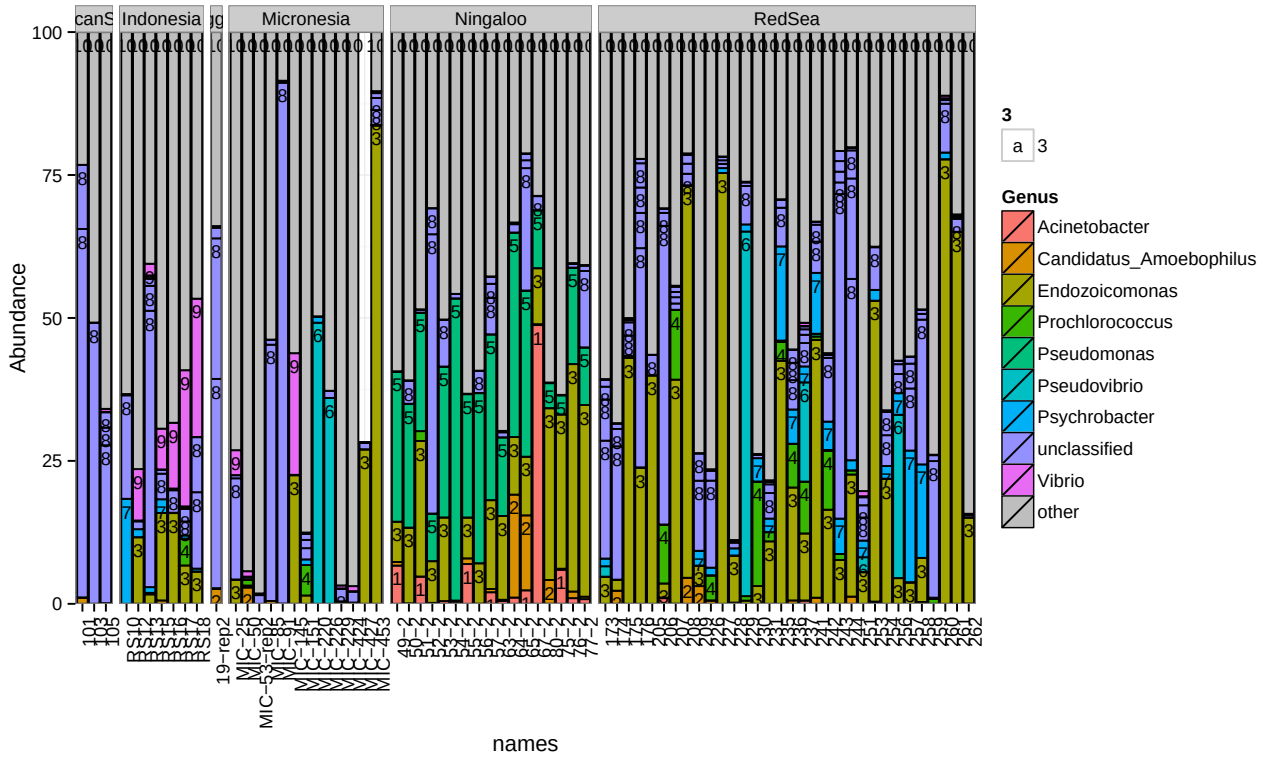

```
draw_barcharts(spistFilt, "Class")  # 0.5
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

## Warning: Removed 1 rows containing missing values (geom_text).
```



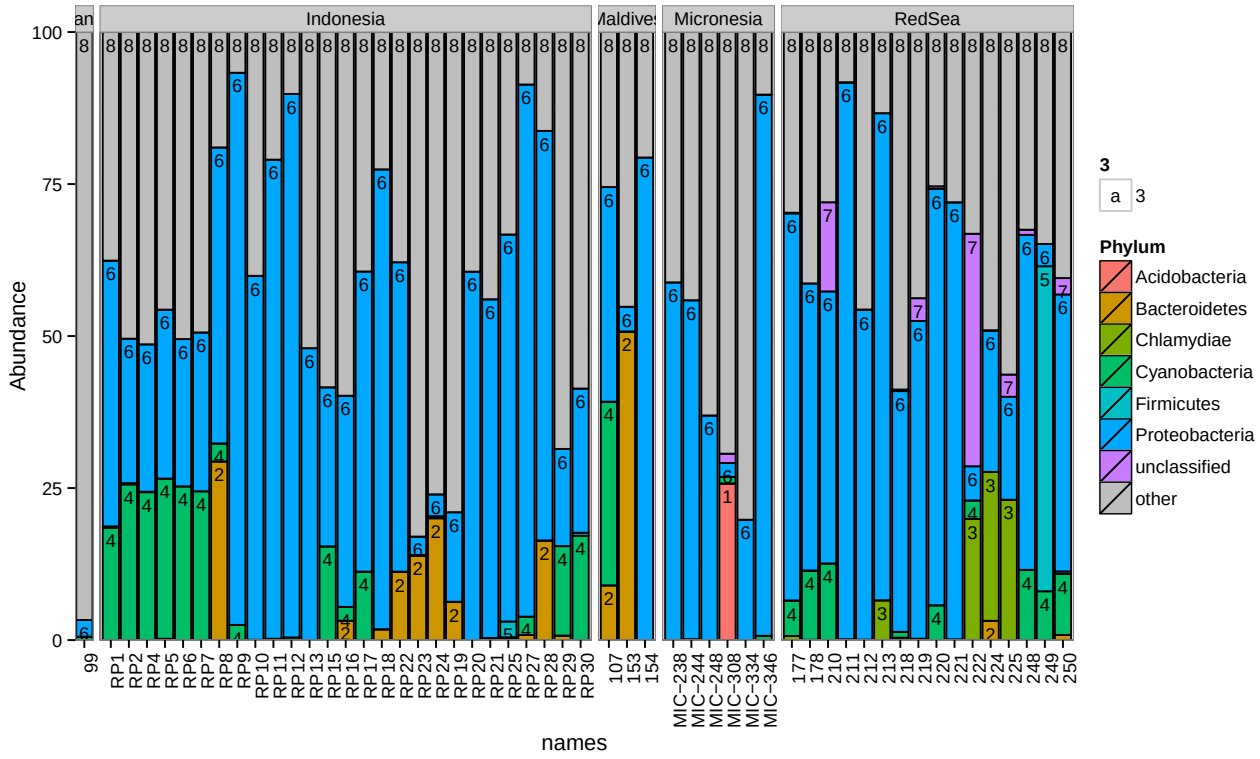

```
draw_barcharts(spistFilt, "Genus")  # 0.8    # 1500 x 700
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

## Warning: Removed 1 rows containing missing values (geom_text).
```



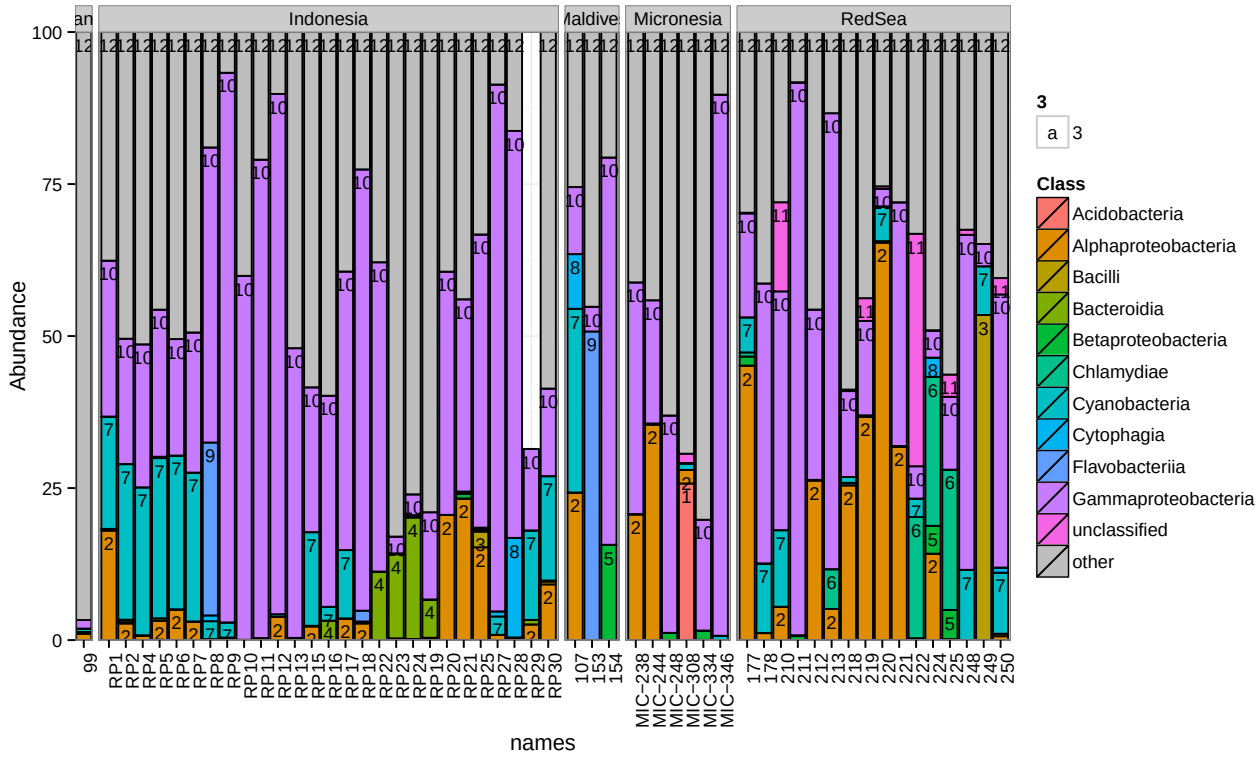

```
pVerr <- subset_samples(allPhylo, species == "Pocillopora verrucosa")
sample_data(pVerr)$names <- factor(sample_names(pVerr), levels = unique(sample_names(pVerr)))
pVerrFilt = filter_taxa(pVerr, function(x) mean(x) > 0.45, TRUE)
draw_barcharts(pVerrFilt, "Phylum")  # 0.3
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



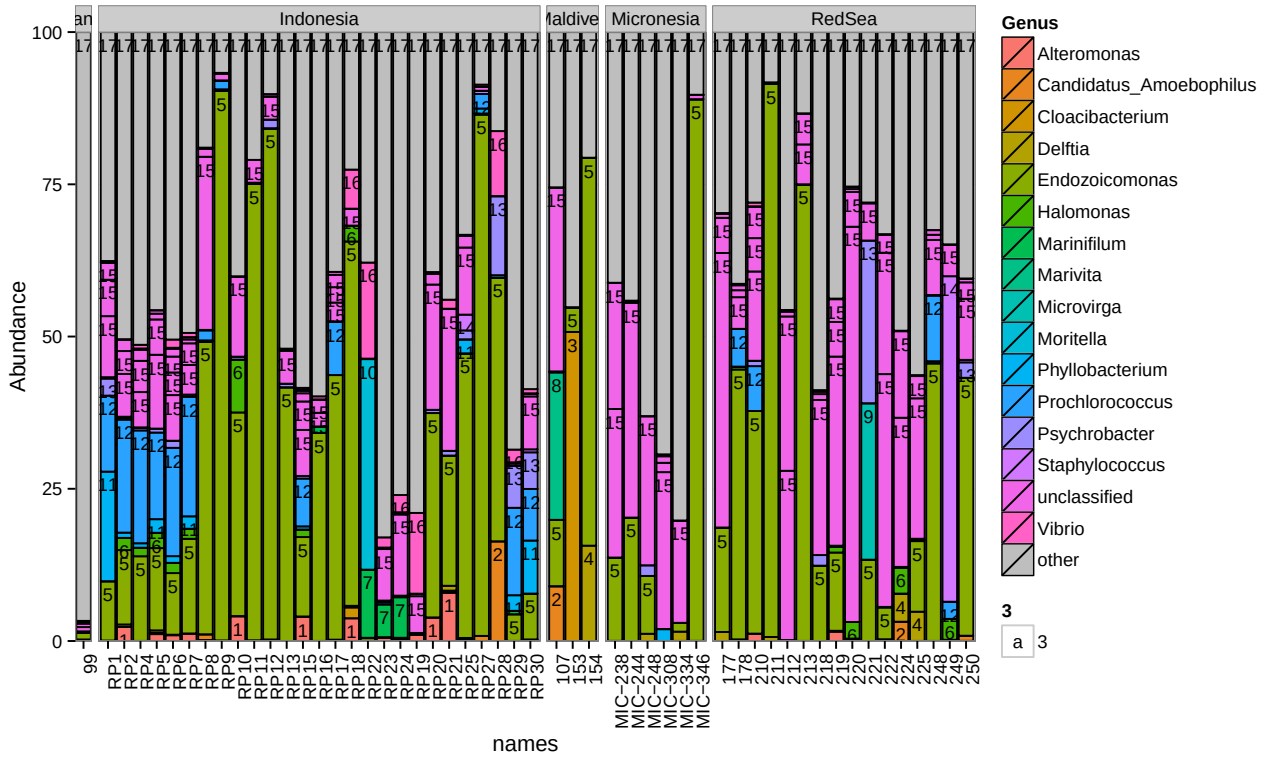

```
draw_barcharts(pVerrFilt, "Class")  # 0.45
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead

## Warning: Removed 1 rows containing missing values (geom_text).
```



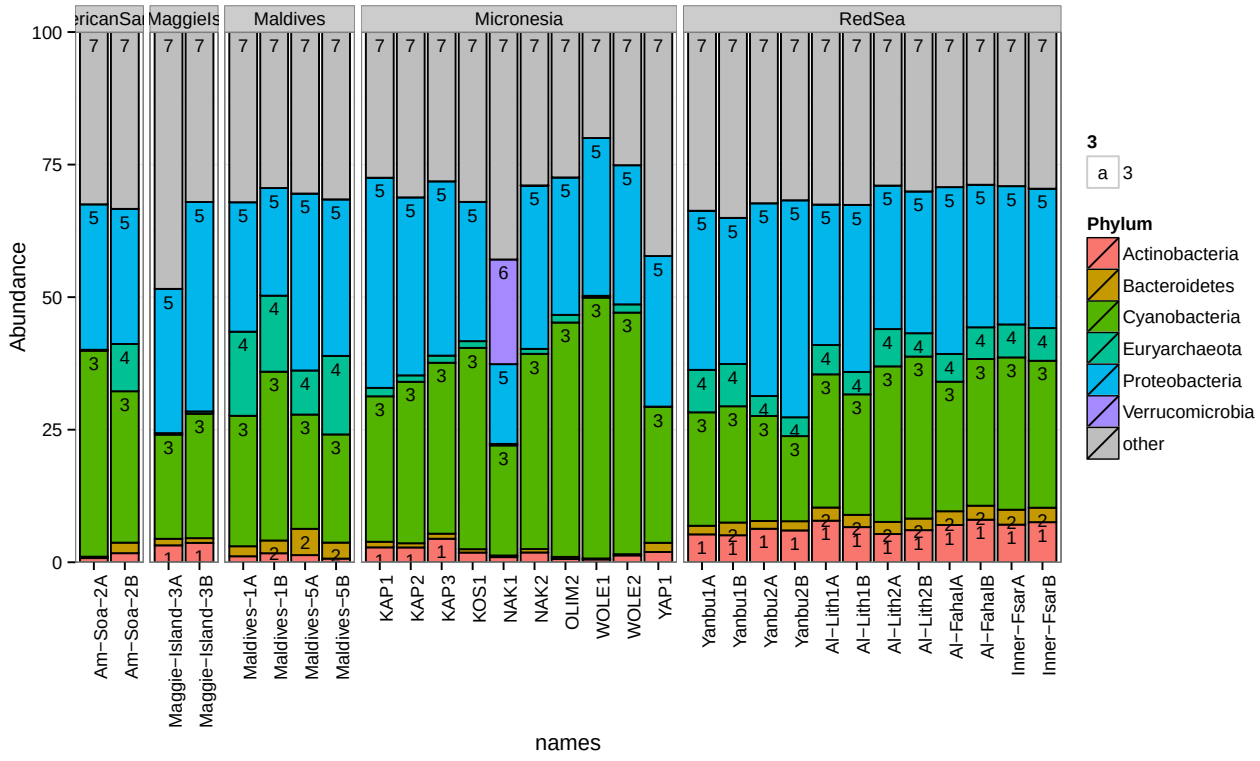

```
draw_barcharts(pVerrFilt, "Genus")  # 0.6    # 1500 x 600
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



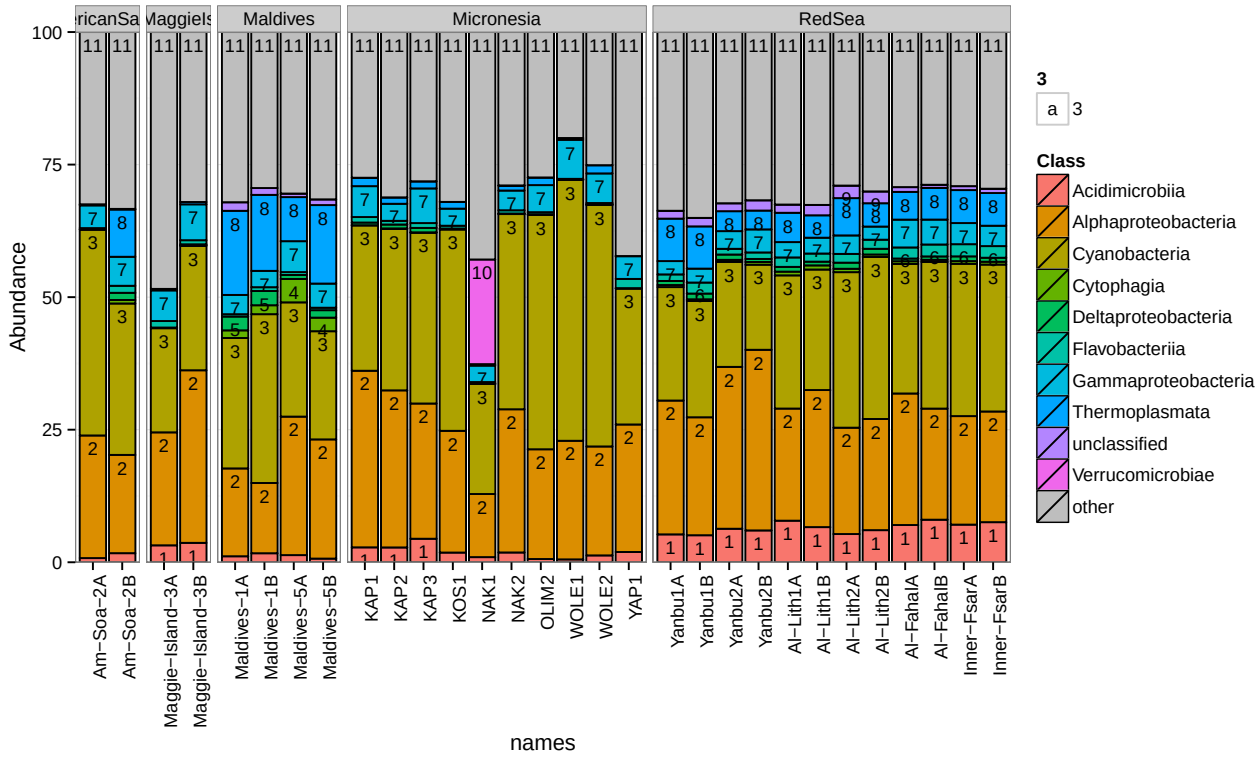

```
sea <- subset_samples(allPhylo, species == "seawater")
sample_data(sea)$names <- factor(sample_names(sea), levels = rownames(metaFile),
    ordered = TRUE)
seaFilt = filter_taxa(sea, function(x) mean(x) > 0.5, TRUE)
draw_barcharts(seaFilt, "Phylum")  # 0.1
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



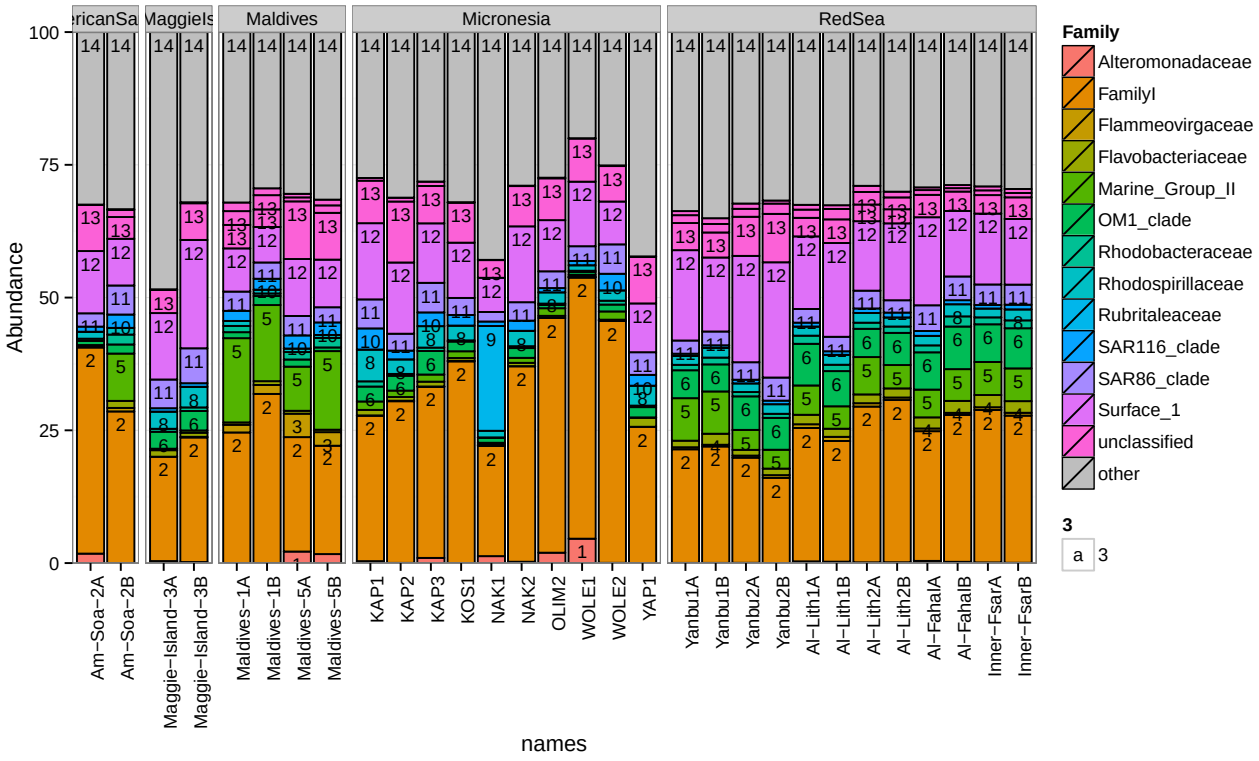

```
draw_barcharts(seaFilt, "Class")  # 0.1
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```




```
draw_barcharts(seaFilt, "Family")  # 1200 x 600
```

```
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
## ymax not defined: adjusting position using y instead
```



The two corals are both dominated by Gammaproteobacteria at the higher taxonomic levels. At the genus level, there is more variability but Endozoicomonas seem to be fairly prevalent. Let's check which bacterial genera are most consistently associated with the corals and may be considered a 'core' microbiome member.

Core coral microbiome members

```
# check for 'core' microbiome members at the genus level which taxa are present
# at 1% overall abundance and at least 50% of samples in Stylophora pistillata?
spistGenusGlom <- tax_glom(spistFilt, taxrank = "Genus")
coreTaxa = filter_taxa(spistGenusGlom, function(x) sum(x > 1) > (0.5 * length(x)),
    TRUE)
tax_table(coreTaxa)
```

```
## Taxonomy Table:     [1 taxa by 7 taxonomic ranks]:
##                Domain     Phylum           Class
## MED000008661 "Bacteria" "Proteobacteria" "Gammaproteobacteria"
##                Order                Family        Genus             Species
## MED000008661 "Oceanospirillales" "Hahellaceae" "Endozoicomonas" NA
```

```
sum(otu_table(coreTaxa) > 1)/nsamples(spist)
```

```
## [1] 0.7671
```

```
# which taxa are present at 1% overall abundance and at least 50% of samples in
# Pocillopora verrucosa?
pVerrGenusGlom <- tax_glom(pVerrFilt, taxrank = "Genus")
coreTaxa = filter_taxa(pVerrGenusGlom, function(x) sum(x > 1) > (0.5 * length(x)),
```

```
    TRUE)
tax_table(coreTaxa)
```

```
## Taxonomy Table:     [1 taxa by 7 taxonomic ranks]:
##              Domain     Phylum           Class
## MED000008683 "Bacteria" "Proteobacteria" "Gammaproteobacteria"
##              Order               Family        Genus            Species
## MED000008683 "Oceanospirillales" "Hahellaceae" "Endozoicomonas" NA
```

```
sum(otu_table(coreTaxa) > 1)/nsamples(pVerr)
```

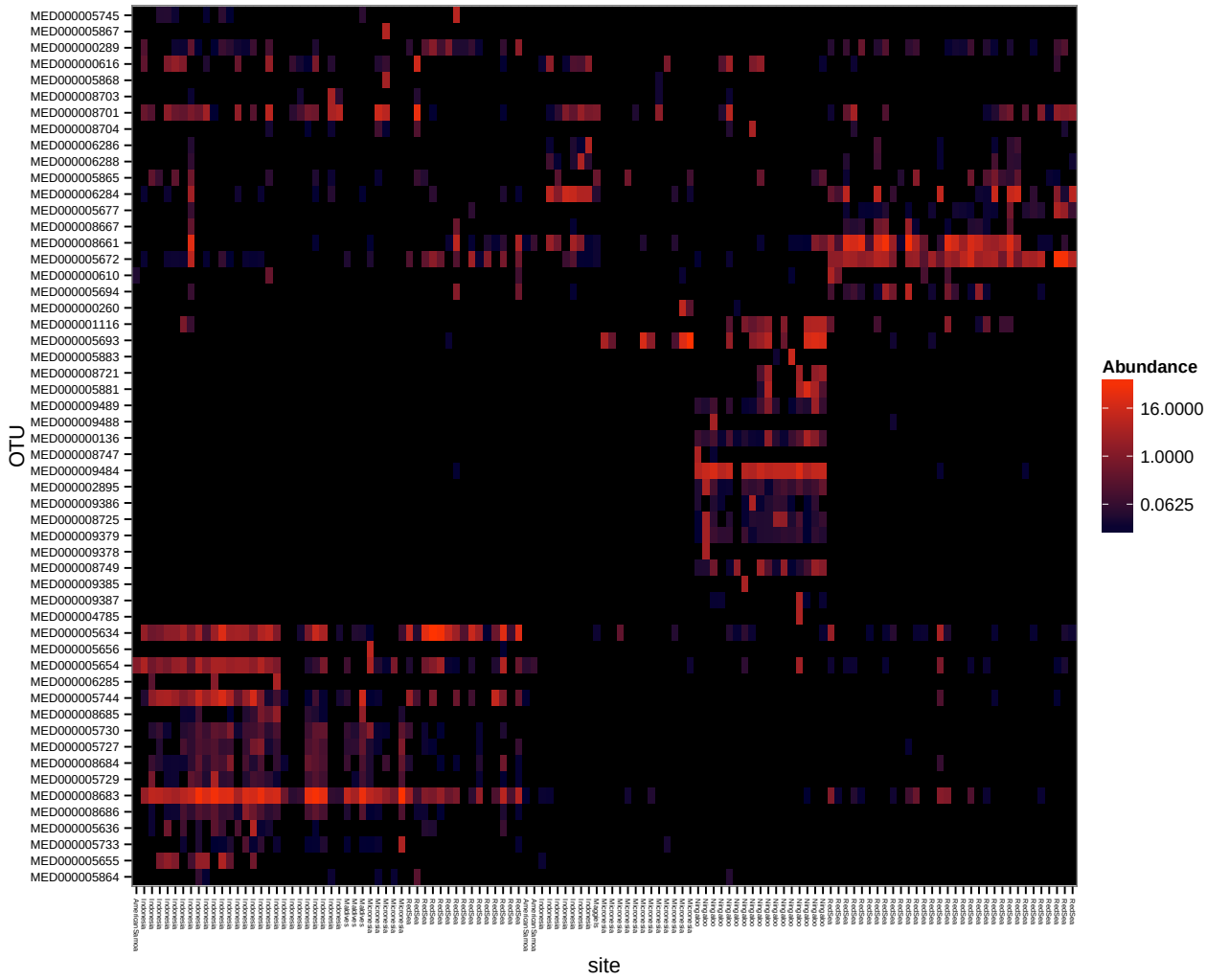
```
## [1] 0.8302
```

Indeed Endozoicomonas were the most prevalent bacteria in the corals and were they only bacterial genera to occur in more than 50% of the colonies sampled. In fact, for both coral species Endozoicomonas occurred in more than 75% of colonies.

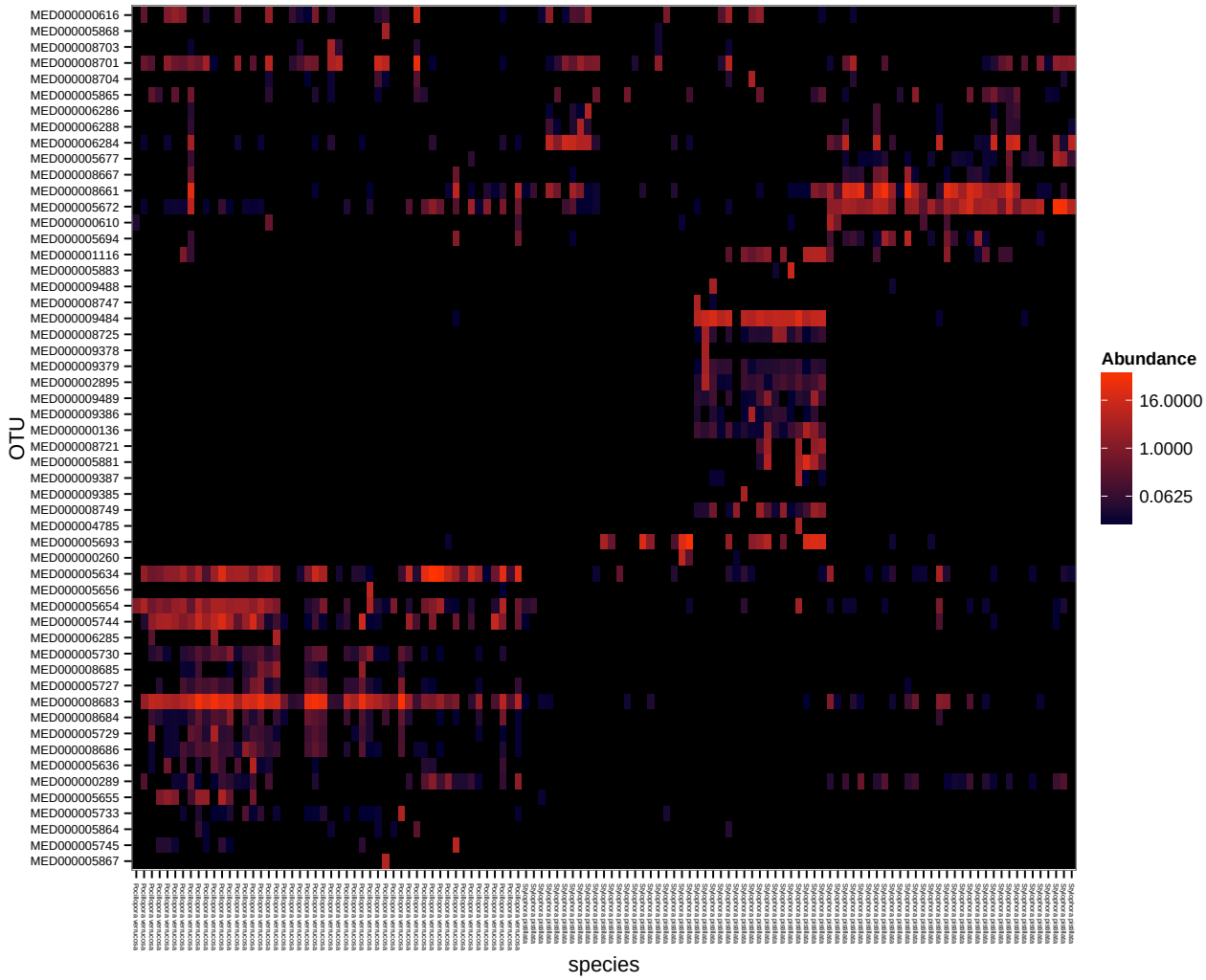
Let's check if these Endozoicomonas OTUs show any patterns across the coral species or at different geographic areas.

Heatmap of different Endozoicomonas MED OTUs across the coral species

```
allPhyloEndo <- subset_taxa(allPhylo, Genus == "Endozoicomonas")
spistPhyloEndo <- subset_samples(allPhyloEndo, species == "Stylophora pistillata")
pVerrPhyloEndo <- subset_samples(allPhyloEndo, species == "Pocillopora verrucosa")
spistPverrEndo <- merge_phyloseq(spistPhyloEndo, pVerrPhyloEndo)

spistPverrEndoFilt = filter_taxa(spistPverrEndo, function(x) mean(x) > 0, TRUE)
spistPverrEndoFiltPrune = prune_samples(sample_sums(spistPverrEndoFilt) > 0, spistPverrEndoFilt)

plot_heatmap(spistPverrEndoFiltPrune, "NMDS", "bray", "site", low = "#000033", high = "#FF3300",
    sample.order = rownames(metaFile3))
```




```
plot_heatmap(spistPverrEndoFiltPrune, "NMDS", "bray", "species", low = "#000033",
    high = "#FF3300", sample.order = rownames(metaFile3))
```



Pretty cool. Looks like the two corals have different Endozoicomonas types, and the types also seem to partition differently across sites for the coral species, i.e., Pocillopora verrucosa has similar Endozoicomonas types across large geographic areas but Stylophora pistillata seems to have different Endozoicomonas types at each area. Let's do some significance testing to see if more of the Endozoicomonas OTUs are different across sites for S. pistillata compared to P. verrucosa.

DESeq2 significance testing for Endozoicomonas MED OTUs

```
# subset endozoicomonas OTUs from the count data as required by DESeq2
countEndos <- subset_taxa(countPhylo, Genus=='Endozoicomonas')
spistCountEndo <- subset_samples(countEndos, species=='Stylophora pistillata')
pVerrCountEndo <- subset_samples(countEndos, species=='Pocillopora verrucosa')
coralCountEndo <- merge_phyloseq(spistCountEndo, pVerrCountEndo)

# do some filtering for Os
spistCountEndoFilt = filter_taxa(spistCountEndo, function(x) mean(x) > 0.0, TRUE)
spistCountEndoFiltPrune = prune_samples(sample_sums(spistCountEndoFilt) > 0, spistCountEndoFilt)
pVerrCountEndoFilt = filter_taxa(pVerrCountEndo, function(x) mean(x) > 0.0, TRUE)
pVerrCountEndoFiltPrune = prune_samples(sample_sums(pVerrCountEndoFilt) > 0, pVerrCountEndoFilt)
```

```r
# convert phyloseq object to DESeq object
spistDeseq <- phyloseq_to_deseq2(spistCountEndoFiltPrune, ~ site)
pverrDeseq <- phyloseq_to_deseq2(pVerrCountEndoFiltPrune, ~ site)

# need to calculate geometric means separately because there are zeros in the data
gm_mean = function(x, na.rm=TRUE){
  exp(sum(log(x[x > 0]), na.rm=na.rm) / length(x))
}
spistMeans <- apply(counts(spistDeseq), 1, gm_mean)
spistDeseq <- estimateSizeFactors(spistDeseq, geoMeans = spistMeans)
pverrMeans <- apply(counts(pverrDeseq), 1, gm_mean)
pverrDeseq <- estimateSizeFactors(pverrDeseq, geoMeans = pverrMeans)

# now can run the DESeq tests
spistDeseq <- DESeq(spistDeseq, fitType="local")
```

```
## using pre-existing size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## -- replacing outliers and refitting for 38 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds))
## estimating dispersions
## fitting model and testing
```

```r
pverrDeseq <- DESeq(pverrDeseq, fitType="local")
```

```
## using pre-existing size factors
## estimating dispersions
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
## -- replacing outliers and refitting for 29 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds))
## estimating dispersions
## fitting model and testing
```

```r
# create a function to check the results for each of the site comparisons
# output significant p-value (FDR adjusted) ordered OTUs

check_site_results <- function(coral_deseq, site_comparison){
  res <- results(coral_deseq, contrast = site_comparison)
  res = res[order(res$padj, na.last=NA), ]
  sigtab = as.data.frame(res[(res$padj < 0.05), ])
  print(sigtab)
}
```

```
# Stylophora pistillata
check_site_results(spistDeseq, c("site", "AmericanSamoa", "RedSea"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000000289   0.4453         -6.767 1.515 -4.466 7.983e-06 0.0002954
## MED000005672  45.9003         -8.229 1.987 -4.140 3.468e-05 0.0006417
## MED000005677   0.2435         -5.288 1.521 -3.478 5.060e-04 0.0062405
## MED000005694   0.8396         -5.769 1.825 -3.160 1.576e-03 0.0145776
## MED000008667   0.2736         -5.025 1.663 -3.021 2.518e-03 0.0186321
## MED000008661  23.8289         -4.508 1.696 -2.658 7.855e-03 0.0484376
```

```
check_site_results(spistDeseq, c("site", "AmericanSamoa", "Ningaloo"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000002895  0.33262         -7.194 1.520 -4.732 2.227e-06 6.905e-05
## MED000009484 24.74728         -9.485 2.124 -4.466 7.965e-06 8.231e-05
## MED000009379  0.22244         -6.706 1.493 -4.490 7.119e-06 8.231e-05
## MED000008725  0.30197         -6.708 1.614 -4.157 3.230e-05 2.503e-04
## MED000009489  0.17044         -6.297 1.539 -4.090 4.309e-05 2.671e-04
## MED000008749  0.58961         -6.953 1.738 -4.000 6.340e-05 3.275e-04
## MED000000136  0.31981         -6.343 1.618 -3.921 8.816e-05 3.904e-04
## MED000009386  0.09981         -5.449 1.566 -3.480 5.006e-04 1.940e-03
## MED000005881  0.46854         -5.531 1.915 -2.888 3.879e-03 1.336e-02
## MED000008721  0.10896         -4.545 1.797 -2.529 1.142e-02 3.541e-02
## MED000005693  1.79057         -5.378 2.162 -2.487 1.287e-02 3.628e-02
```

```
check_site_results(spistDeseq, c("site", "AmericanSamoa", "Micronesia"))
```

```
## [1] baseMean       log2FoldChange lfcSE          stat
## [5] pvalue         padj
## <0 rows> (or 0-length row.names)
```

```
check_site_results(spistDeseq, c("site", "AmericanSamoa", "Indonesia"))
```

```
## [1] baseMean       log2FoldChange lfcSE          stat
## [5] pvalue         padj
## <0 rows> (or 0-length row.names)
```

```
check_site_results(spistDeseq, c("site", "RedSea", "Ningaloo"))
```

```
##              baseMean log2FoldChange  lfcSE   stat    pvalue      padj
## MED000005672 45.90025         11.032 1.0441 10.566 4.273e-26 1.795e-24
## MED000009484 24.74728        -11.528 1.3365 -8.625 6.399e-18 1.344e-16
## MED000008661 23.82895          6.504 0.8044  8.085 6.212e-16 8.696e-15
## MED000006284  4.66332          9.652 1.6475  5.858 4.671e-09 4.905e-08
## MED000005694  0.83957          8.958 1.8933  4.731 2.231e-06 1.561e-05
## MED000000289  0.44531          9.426 1.9822  4.755 1.982e-06 1.561e-05
## MED000005693  1.79057         -6.795 1.4752 -4.606 4.102e-06 2.154e-05
## MED000002895  0.33262         -9.394 2.0396 -4.606 4.103e-06 2.154e-05
## MED000000136  0.31981         -8.944 1.9658 -4.550 5.368e-06 2.255e-05
```

```
## MED000008749  0.58961         -9.471 2.0788 -4.556 5.213e-06 2.255e-05
## MED000005677  0.24346          8.322 1.8948  4.392 1.124e-05 4.293e-05
## MED000008725  0.30197         -9.002 2.0704 -4.348 1.373e-05 4.805e-05
## MED000009379  0.22244         -8.767 2.0752 -4.224 2.396e-05 7.739e-05
## MED000008667  0.27364          7.956 1.9383  4.105 4.047e-05 1.214e-04
## MED000009489  0.17044         -8.376 2.1012 -3.986 6.715e-05 1.880e-04
## MED000008701  4.18358          4.573 1.1844  3.861 1.128e-04 2.961e-04
## MED000006288  0.18381          6.537 1.9231  3.399 6.762e-04 1.590e-03
## MED000009386  0.09981         -7.387 2.1746 -3.397 6.816e-04 1.590e-03
## MED000005881  0.46854         -7.859 2.3362 -3.364 7.689e-04 1.700e-03
## MED000008683  0.32075          4.732 1.4268  3.316 9.126e-04 1.916e-03
## MED000000616  0.94550         -5.296 1.6137 -3.282 1.031e-03 2.061e-03
## MED000008721  0.10896         -6.484 2.3768 -2.728 6.373e-03 1.217e-02
## MED000006286  0.07270          5.374 2.0396  2.635 8.416e-03 1.537e-02
```

```
check_site_results(spistDeseq, c("site", "RedSea", "Micronesia"))
```

```
##              baseMean log2FoldChange  lfcSE  stat    pvalue      padj
## MED000008661  23.8289          9.196 1.0638 8.644 5.405e-18 2.270e-16
## MED000005672  45.9003          6.468 0.9063 7.137 9.569e-13 2.009e-11
## MED000000289   0.4453          9.307 2.0013 4.650 3.312e-06 4.637e-05
## MED000005694   0.8396          7.736 1.8558 4.168 3.069e-05 3.222e-04
## MED000006284   4.6633          5.816 1.4129 4.116 3.849e-05 3.233e-04
## MED000005677   0.2435          5.543 1.4749 3.758 1.714e-04 1.028e-03
## MED000008667   0.2736          7.529 1.9983 3.768 1.649e-04 1.028e-03
## MED000001116   1.2828          6.022 1.8659 3.228 1.248e-03 6.555e-03
## MED000008701   4.1836          4.117 1.3038 3.158 1.589e-03 7.417e-03
## MED000006288   0.1838          6.060 1.9856 3.052 2.272e-03 9.541e-03
```

```
check_site_results(spistDeseq, c("site", "RedSea", "Indonesia"))
```

```
##              baseMean log2FoldChange lfcSE  stat    pvalue      padj
## MED000005672  45.9003          6.244 1.084 5.761 8.368e-09 2.092e-07
## MED000000289   0.4453          7.879 2.219 3.551 3.834e-04 4.793e-03
## MED000005677   0.2435          6.634 2.146 3.092 1.990e-03 1.658e-02
## MED000005694   0.8396          4.940 1.766 2.797 5.163e-03 3.227e-02
## MED000001116   1.2828          5.369 2.028 2.647 8.115e-03 4.058e-02
## MED000005634   0.4262          4.922 1.936 2.542 1.102e-02 4.591e-02
```

```
check_site_results(spistDeseq, c("site", "Ningaloo", "Micronesia"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000009484 24.74728         10.236 1.473  6.949 3.672e-12 1.359e-10
## MED000002895  0.33262         10.466 1.870  5.596 2.193e-08 4.056e-07
## MED000009379  0.22244          9.810 1.906  5.148 2.632e-07 3.246e-06
## MED000000136  0.31981          7.785 1.580  4.926 8.387e-07 7.758e-06
## MED000008725  0.30197          9.297 2.025  4.592 4.388e-06 3.247e-05
## MED000009489  0.17044          8.964 2.006  4.469 7.867e-06 4.851e-05
## MED000008749  0.58961          9.274 2.107  4.401 1.076e-05 5.685e-05
## MED000005672 45.90025         -4.565 1.209 -3.775 1.603e-04 6.699e-04
## MED000001116  1.28283          7.406 1.964  3.770 1.630e-04 6.699e-04
## MED000000616  0.94550          7.190 2.040  3.524 4.253e-04 1.431e-03
```

```
## MED000009386  0.09981          7.593 2.142  3.545 3.925e-04 1.431e-03
## MED000005881  0.46854          7.028 2.416  2.909 3.623e-03 1.117e-02
## MED000008721  0.10896          5.866 2.434  2.410 1.596e-02 4.543e-02
```

```
check_site_results(spistDeseq, c("site", "Ningaloo", "Indonesia"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000009484 24.74728         12.009 1.936  6.203 5.529e-10 1.493e-08
## MED000006284  4.66332         -9.180 1.960 -4.683 2.822e-06 3.810e-05
## MED000008661 23.82895         -4.134 1.134 -3.645 2.674e-04 1.031e-03
## MED000005693  1.79057          7.649 2.050  3.731 1.905e-04 1.031e-03
## MED000000136  0.31981          7.841 2.134  3.675 2.380e-04 1.031e-03
## MED000008749  0.58961          8.285 2.246  3.689 2.250e-04 1.031e-03
## MED000002895  0.33262          8.414 2.192  3.839 1.237e-04 1.031e-03
## MED000005672 45.90025         -4.788 1.341 -3.570 3.576e-04 1.073e-03
## MED000008725  0.30197          7.950 2.227  3.570 3.573e-04 1.073e-03
## MED000009379  0.22244          7.823 2.215  3.532 4.130e-04 1.115e-03
## MED000009489  0.17044          7.405 2.241  3.305 9.495e-04 2.330e-03
## MED000001116  1.28283          6.753 2.109  3.201 1.368e-03 3.079e-03
## MED000006288  0.18381         -6.595 2.160 -3.054 2.260e-03 4.694e-03
## MED000009386  0.09981          6.434 2.293  2.806 5.016e-03 9.675e-03
## MED000005881  0.46854          6.451 2.442  2.642 8.245e-03 1.484e-02
## MED000008701  4.18358         -4.162 1.609 -2.586 9.710e-03 1.639e-02
## MED000006286  0.07270         -5.611 2.284 -2.457 1.401e-02 2.225e-02
## MED000008721  0.10896          5.337 2.439  2.188 2.864e-02 4.297e-02
```

```
check_site_results(spistDeseq, c("site", "Micronesia", "Indonesia"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000008661  23.8289         -6.826 1.323 -5.161 2.463e-07 2.709e-06
## MED000006284   4.6633         -5.344 1.793 -2.981 2.875e-03 1.581e-02
## MED000000616   0.9455         -6.324 2.259 -2.799 5.126e-03 1.879e-02
```

```
# Pocillopora verrucosa
check_site_results(pverrDeseq, c("site", "Maldives", "RedSea"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000005634  255.716         -9.037 1.267 -7.132 9.904e-13 1.783e-11
## MED000000289    1.384         -5.780 1.929 -2.995 2.740e-03 2.466e-02
## MED000005672    3.650         -4.019 1.514 -2.654 7.949e-03 4.769e-02
```

```
check_site_results(pverrDeseq, c("site", "Maldives", "Micronesia"))
```

```
## [1] baseMean       log2FoldChange lfcSE          stat
## [5] pvalue         padj
## <0 rows> (or 0-length row.names)
```

```
check_site_results(pverrDeseq, c("site", "Maldives", "Indonesia"))
```

```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000005634    255.7         -5.488 1.231 -4.459 8.238e-06 0.0001977
```

```
check_site_results(pverrDeseq, c("site", "RedSea", "Micronesia"))
```

```
##              baseMean log2FoldChange  lfcSE   stat    pvalue      padj
## MED000005634  255.716         10.107 1.1443  8.832 1.028e-18 1.542e-17
## MED000005672    3.650          4.675 1.3549  3.450 5.603e-04 3.279e-03
## MED000000289    1.384          6.330 1.8576  3.407 6.557e-04 3.279e-03
## MED000008683  359.232         -2.339 0.9037 -2.588 9.660e-03 3.623e-02
```

```
check_site_results(pverrDeseq, c("site", "RedSea", "Indonesia"))
```

```
##              baseMean log2FoldChange  lfcSE   stat    pvalue      padj
## MED000005672    3.650          5.503 0.8747  6.291 3.157e-10 7.578e-09
## MED000005634  255.716          3.549 0.6308  5.626 1.846e-08 2.215e-07
## MED000008683  359.232         -2.506 0.5847 -4.286 1.817e-05 1.453e-04
## MED000008686    1.164         -2.844 0.7519 -3.782 1.557e-04 9.344e-04
## MED000005655    3.205         -7.518 2.0199 -3.722 1.975e-04 9.482e-04
## MED000005729    1.074         -3.505 0.9725 -3.604 3.130e-04 1.252e-03
## MED000008684    1.428         -2.267 0.7219 -3.141 1.685e-03 5.779e-03
## MED000000289    1.384          2.103 0.8030  2.618 8.833e-03 2.650e-02
```

```
check_site_results(pverrDeseq, c("site", "Micronesia", "Indonesia"))
```

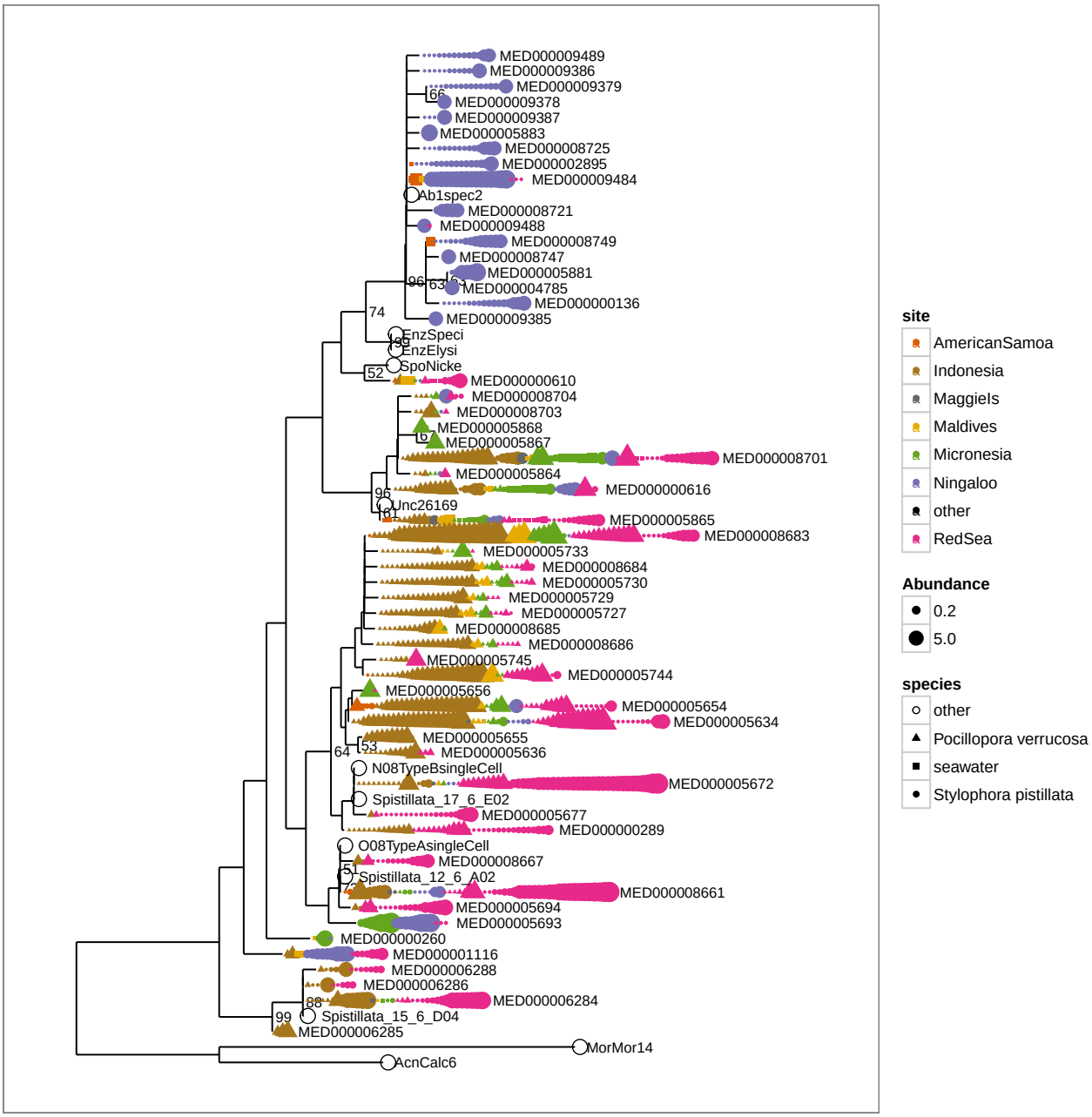
```
##              baseMean log2FoldChange lfcSE   stat    pvalue      padj
## MED000005634  255.716         -6.558 1.103 -5.947 2.730e-09 4.914e-08
## MED000005655    3.205         -6.374 2.164 -2.945 3.225e-03 2.902e-02
```

There is quite a few more significant differences across sites for Stylophora pistillata (90) compared to Pocillopora verrucosa (18), supporting what is shown in the heatmaps. I'll add an asterisk next to each of the significantly different OTUs in the heatmap to visually display these results.

Endozoicomonas seem to be displaying strain-specific relationships with the corals and across sites. I'll do a phylogenetic analysis of the Endozoicomonas sequences to further explore these relationships. The sequences were aligned using the SINA web service and imported into ARB for manual refinement, before being exported for use here.

Endozoicomonas phylogenetic tree with meta-data

```
# subset out our corals / seawater of interest
endoTreeSpist <- subset_samples(endoTree, species == "Stylophora pistillata")
endoTreePverr <- subset_samples(endoTree, species == "Pocillopora verrucosa")
endoTreeSea <- subset_samples(endoTree, species == "seawater")
endoTreeOther <- subset_samples(endoTree, species == "other")
endoTreeCorals <- merge_phyloseq(endoTreeSpist, endoTreePverr, endoTreeSea, endoTreeOther)

# plot tree - phyloseq makes this easy

plot_tree(endoTreeCorals, label.tips = "taxa_names", color = "site", shape = "species",
    size = "abundance", nodelabf = nodeplotboot(100, 50, 3), ladderize = "left",
    base.spacing = 0.01) + scale_color_manual(values = cols) + scale_shape_manual(values = c(other = 1,
    `Pocillopora verrucosa` = 17, `Stylophora pistillata` = 16, seawater = 15))
```



Some interesting things coming up here. Several abundant but phylogenetically diverse Endozoicomonas OTUs appear to co-inhabit the same coral colonies. There are also clear host and site groupings of Endozoicomonas strains. I also included single cell 16S sequences in the tree and they are identical to the abundant MED OTUs from the Red Sea, suggesting that the MED procedure produces biologically relevant OTUs. Also in the tree are sequences from an earlier study of Red Sea Stylophora pistillata, named 'Spistillata_17_6_E02, Spistillata_12_6_A02, Spistillata_15_6_D04', and these are also the same as the abundant MED nodes in this study, suggesting that the most abundant MED OTUs in Red Sea S. pistillata have not changed for ~ 4 years.