MB590-012 Microbiome Analysis

Christine Hawkes

2/16/2022

Contents

Topic: EXPLORATORY ANALYSIS - BETA DIVERSITY

```
References:
Anderson et al. (2011) Navigating the multiple meanings of Beta diversity: a roadmap for the practicing

Legendre & DeCaceres (2013) Beta diverstity as the variance of community data: dissimilarity coefficien

Data reference:
Lozupone & Knight (2007) PNAS 104:11436-11440 doi.org/10.1073/pnas.0611525104]
```

SETUP

Load and install R packages

```
library(phyloseq)
library(tidyverse)
library(DESeq2)
library(vegan)
library(ape)

# BiocManager::install("philr")
library(philr)
library(microbiome)
library(compositions)

library(ggplot2)
library(ggpubr)
# devtools::install_github("jfq3/ggordiplots", force=TRUE)
library(ggordiplots) #  may have to check the box to load this package
```

Load data and subset

If you saved `ps_gp_bact` from last week you can load that ps object
Otherwise, re-load and subset the GlobalPatterns data
https://www.rdocumentation.org/packages/phyloseq/versions/1.16.2/topics/data-GlobalPatterns

```
# load data - ps object with otu table, taxa table, sample data, & tree
data(GlobalPatterns)
ps_gp <- GlobalPatterns
# remove Mocks
ps_gp <-phyloseq::subset_samples(ps_gp, SampleType != "Mock")
# subset to only Bacteria
ps_gp_bact <- phyloseq::subset_taxa(ps_gp, Kingdom=="Bacteria")
# filter taxa not seen at least 5 times in at least 20% of samples
# filtering will make the computations feasible in class
ps_gp_bact <- phyloseq::filter_taxa(ps_gp_bact, function(x) sum(x>5) > (0.2*length(x)), TRUE)
ps_gp_bact # should have 23 samples and 2640 taxa
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2640 taxa and 23 samples ]
## sample_data() Sample Data:       [ 23 samples by 7 sample variables ]
## tax_table()   Taxonomy Table:    [ 2640 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2640 tips and 2639 internal nodes ]
```

Prepare data in Phyloseq

For phyloseq help, see http://joey711.github.io/phyloseq/

Add new sample data to the existing file

```
# add pH and salinity data columns to the sample data file
# retrieve the current sample data file from the ps object
sam.old <- sample_data(ps_gp_bact)
sam.old[1]
```

```
##     X.SampleID  Primer Final_Barcode Barcode_truncated_plus_T
## CL3        CL3 ILBC_01        AACGCA                   TGCGTT
##     Barcode_full_length SampleType                              Description
## CL3         CTAGCGTGCGT       Soil Calhoun South Carolina Pine soil, pH 4.9
```

```
str(sam.old)
```

```
## 'data.frame':    23 obs. of  7 variables:
## Formal class 'sample_data' [package "phyloseq"] with 4 slots
##   ..@ .Data    :List of 7
##   .. ..$ : Factor w/ 23 levels "AQC1cm","AQC4cm",..: 5 4 18 11 8 12 9 6 13 10 ...
##   .. ..$ : Factor w/ 23 levels "ILBC_01","ILBC_02",..: 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ : Factor w/ 23 levels "AACGCA","AACTCG",..: 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ : Factor w/ 23 levels "AACTGT","ACAGTT",..: 21 11 2 18 8 4 15 6 17 10 ...
##   .. ..$ : Factor w/ 23 levels "AGCCGACTCTG",..: 9 5 17 20 7 8 15 18 23 19 ...
##   .. ..$ : Factor w/ 8 levels "Feces","Freshwater",..: 7 7 7 1 1 6 6 6 8 8 ...
##   .. ..$ : Factor w/ 22 levels "Allequash Creek, 0-1cm depth",..: 4 5 17 11 8 12 9 6 13 10 ...
##   ..@ names    : chr  "X.SampleID" "Primer" "Final_Barcode" "Barcode_truncated_plus_T" ...
##   ..@ row.names: chr  "CL3" "CC1" "SV1" "M31Fcsw" ...
##   ..@ .S3Class : chr "data.frame"
```

```
# load the new data to add
# modify with path to GitHub file
sam.new <- read.csv("wk6_sam_new.csv", row.names = 1)
sam.new[1,]
```

```
##      pH salinity
## CL3 4.9        4
```

```
str(sam.new)
```

```
## 'data.frame':    23 obs. of  2 variables:
##  $ pH      : num  4.9 6.1 8.3 6.6 6.4 5.4 5.5 5.6 7.1 6.9 ...
##  $ salinity: num  4 2 5 19 18 70 68 72 10 8 ...
```

```
# check that rownames match
all(rownames(sam.old) == rownames(sam.new))
```

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

```
# merge the two data frames and set up the resulting file for phyloseq
sam.all<-merge(sam.old, sam.new, by="row.names")
sam.all[1,]
```

```
##   Row.names X.SampleID  Primer Final_Barcode Barcode_truncated_plus_T
## 1    AQC1cm     AQC1cm ILBC_16        ACAGCA                   TGCTGT
##   Barcode_full_length         SampleType                  Description pH
## 1         GACCACTGCTG Freshwater (creek) Allequash Creek, 0-1cm depth  9
##   salinity
## 1      0.1
```

```
# merge removes row names - fix this
sam.all <- tibble::column_to_rownames(sam.all, var = "Row.names")
sam.all[1,]
```

```
##        X.SampleID  Primer Final_Barcode Barcode_truncated_plus_T
## AQC1cm     AQC1cm ILBC_16        ACAGCA                   TGCTGT
##        Barcode_full_length         SampleType                  Description pH
## AQC1cm         GACCACTGCTG Freshwater (creek) Allequash Creek, 0-1cm depth  9
##        salinity
## AQC1cm      0.1
```

```
str(sam.all)
```

```
## 'data.frame':    23 obs. of  9 variables:
##  $ X.SampleID             : Factor w/ 23 levels "AQC1cm","AQC4cm",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Primer                 : Factor w/ 23 levels "ILBC_01","ILBC_02",..: 13 14 15 2 1 8 11 5 7 10 ..
##  $ Final_Barcode          : Factor w/ 23 levels "AACGCA","AACTCG",..: 13 14 15 2 1 8 11 5 7 10 ...
##  $ Barcode_truncated_plus_T: Factor w/ 23 levels "AACTGT","ACAGTT",..: 22 5 7 11 21 6 9 8 15 10 ...
##  $ Barcode_full_length    : Factor w/ 23 levels "AGCCGACTCTG",..: 12 3 2 5 9 18 6 7 15 19 ...
##  $ SampleType             : Factor w/ 8 levels "Feces","Freshwater",..: 3 3 3 7 7 6 2 1 6 8 ...
##  $ Description            : Factor w/ 22 levels "Allequash Creek, 0-1cm depth",..: 1 2 3 5 4 6 7 8 9
##  $ pH                     : num  9 9.1 9.2 6.1 4.9 5.6 9.2 6.4 5.5 6.9 ...
##  $ salinity               : num  0.1 0.2 0.3 2 4 72 0.5 18 68 8 ...
```

```
# replace sample_data in ps object
SAM <- phyloseq::sample_data(sam.all)
phyloseq::sample_data(ps_gp_bact) <- SAM

# confirm replacement
sample_data(ps_gp_bact)
```

```
##          X.SampleID  Primer Final_Barcode Barcode_truncated_plus_T
## CL3             CL3 ILBC_01        AACGCA                   TGCGTT
## CC1             CC1 ILBC_02        AACTCG                   CGAGTT
## SV1             SV1 ILBC_03        AACTGT                   ACAGTT
## M31Fcsw     M31Fcsw ILBC_04        AAGAGA                   TCTCTT
## M11Fcsw     M11Fcsw ILBC_05        AAGCTG                   CAGCTT
## M31Plmr     M31Plmr ILBC_07        AATCGT                   ACGATT
## M11Plmr     M11Plmr ILBC_08        ACACAC                   GTGTGT
## F21Plmr     F21Plmr ILBC_09        ACACAT                   ATGTGT
```

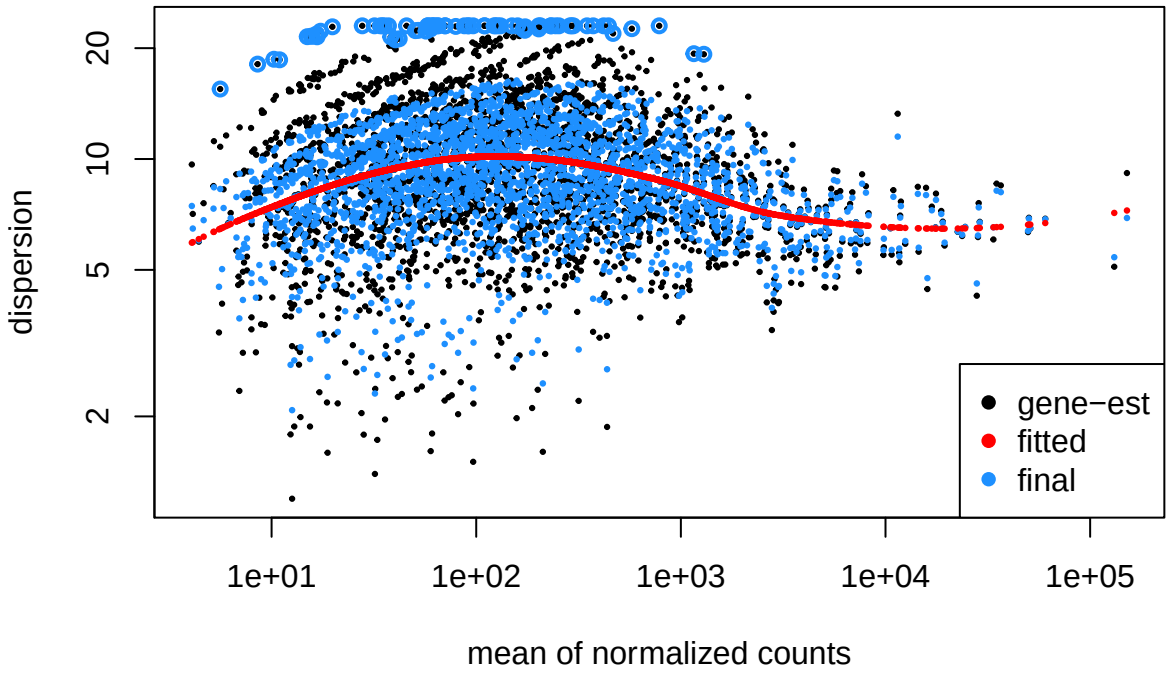
```
## M31Tong     M31Tong ILBC_10        ACACGA                  TCGTGT
## M11Tong     M11Tong ILBC_11        ACACGG                  CCGTGT
## LMEpi24M   LMEpi24M ILBC_13        ACACTG                  CAGTGT
## SLEpi20M   SLEpi20M ILBC_15        ACAGAG                  CTCTGT
## AQC1cm       AQC1cm ILBC_16        ACAGCA                  TGCTGT
## AQC4cm       AQC4cm ILBC_17        ACAGCT                  AGCTGT
## AQC7cm       AQC7cm ILBC_18        ACAGTG                  CACTGT
## NP2             NP2 ILBC_19        ACAGTT                  AACTGT
## NP3             NP3 ILBC_20        ACATCA                  TGATGT
## NP5             NP5 ILBC_21        ACATGA                  TCATGT
## TRRsed1     TRRsed1 ILBC_22        ACATGT                  ACATGT
## TRRsed2     TRRsed2 ILBC_23        ACATTC                  GAATGT
## TRRsed3     TRRsed3 ILBC_24        ACCACA                  TGTGGT
## TS28           TS28 ILBC_25        ACCAGA                  TCTGGT
## TS29           TS29 ILBC_26        ACCAGC                  GCTGGT
##          Barcode_full_length         SampleType
## CL3              CTAGCGTGCGT               Soil
## CC1              CATCGACGAGT               Soil
## SV1              GTACGCACAGT               Soil
## M31Fcsw          TCGACATCTCT              Feces
## M11Fcsw          CGACTGCAGCT              Feces
## M31Plmr          CGAGTCACGAT               Skin
## M11Plmr          GCCATAGTGTG               Skin
## F21Plmr          GTAGACATGTG               Skin
## M31Tong          TGTGGCTCGTG             Tongue
## M11Tong          TAGACACCGTG             Tongue
## LMEpi24M         CATGAACAGTG         Freshwater
## SLEpi20M         AGCCGACTCTG         Freshwater
## AQC1cm           GACCACTGCTG Freshwater (creek)
## AQC4cm           CAAGCTAGCTG Freshwater (creek)
## AQC7cm           ATGAAGCACTG Freshwater (creek)
## NP2              TCGCGCAACTG              Ocean
## NP3              GCTAAGTGATG              Ocean
## NP5              GAACGATCATG              Ocean
## TRRsed1          CACGTGACATG Sediment (estuary)
## TRRsed2          TGCGCTGAATG Sediment (estuary)
## TRRsed3          GATGTATGTGG Sediment (estuary)
## TS28             GCATCGTCTGG              Feces
## TS29             CTAGTCGCTGG              Feces
##                                          Description  pH salinity
## CL3          Calhoun South Carolina Pine soil, pH 4.9 4.9      4.0
## CC1          Cedar Creek Minnesota, grassland, pH 6.1 6.1      2.0
## SV1        Sevilleta new Mexico, desert scrub, pH 8.3 8.3      5.0
## M31Fcsw       M3, Day 1, fecal swab, whole body study 6.6     19.0
## M11Fcsw      M1, Day 1, fecal swab, whole body study  6.4     18.0
## M31Plmr       M3, Day 1, right palm, whole body study 5.4     70.0
## M11Plmr      M1, Day 1, right palm, whole body study  5.5     68.0
## F21Plmr     F1, Day 1,  right palm, whole body study  5.6     72.0
## M31Tong          M3, Day 1, tongue, whole body study  7.1     10.0
## M11Tong          M1, Day 1, tongue, whole body study  6.9      8.0
## LMEpi24M Lake Mendota Minnesota, 24 meter epilimnion  9.2      0.5
## SLEpi20M Sparkling Lake Wisconsin, 20 meter eplimnion 9.3      0.4
## AQC1cm                   Allequash Creek, 0-1cm depth 9.0      0.1
## AQC4cm                  Allequash Creek, 3-4 cm depth 9.1      0.2
```

```
## AQC7cm                  Allequash Creek, 6-7 cm depth 9.2      0.3
## NP2           Newport Pier, CA surface water, Time 1 8.1     35.0
## NP3           Newport Pier, CA surface water, Time 2 8.2     36.0
## NP5           Newport Pier, CA surface water, Time 3 8.0     38.0
## TRRsed1              Tijuana River Reserve, depth 1 8.6     30.0
## TRRsed2              Tijuana River Reserve, depth 2 8.7     29.0
## TRRsed3              Tijuana River Reserve, depth 2 8.5     27.0
## TS28                                      Twin #1 6.6     20.0
## TS29                                      Twin #2 6.7     21.0
```

VST DATA TRANSFORMATION in DESEQ2

Microbiome data should be transformed prior to analysis of beta-diversity

```
ps_ds <- phyloseq::phyloseq_to_deseq2(ps_gp_bact, ~1)
ps_ds = DESeq2::estimateSizeFactors(ps_ds)
ps_ds = DESeq2::estimateDispersions(ps_ds, fitType = "parametric")
DESeq2::plotDispEsts(ps_ds)
```




```
# make a copy of the ps object with vst-transformed otu_table
ps_vst <- ps_gp_bact
vst<-DESeq2::getVarianceStabilizedData(ps_ds)
phyloseq::otu_table(ps_vst) <- phyloseq::otu_table(vst, taxa_are_rows = TRUE)
ps_vst
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2640 taxa and 23 samples ]
## sample_data() Sample Data:       [ 23 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 2640 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2640 tips and 2639 internal nodes ]
```

ORDINATIONS

We'll run two examples of ordinations on vst-transformed data:

- NMDS in phyloseq
- NMDS in vegan with environmental data

Check available distances/dissimilarity metrics

```
help(distanceMethodList, package="phyloseq")
help(betadiver, package="vegan")
help(vegdist, package="vegan")
help(decostand, package = "vegan")
```

Non-metric Multidimensional Scaling (NMDS)

For Bray-Curtis distances, all values in otu table must be positive

```
min(otu_table(ps_vst))
```

```
## [1] -2.222172
```

```
ps_vst_pos <- transform_sample_counts(ps_vst, function(x) x+2.23)
```

NMDS with Bray-Curtis distances in phyloseq::ordinate

See manual https://bioconductor.org/packages/devel/bioc/manuals/phyloseq/man/phyloseq.pdf
`phyloseq` does not allow `envfit` and no specifications can be adjusted, but easy to explore samples/species data
Available methods include: DCA, CCA, RDA, CAP, DPCoA, NMDS, MDS/PCoA

Run NMDS with phyloseq::ordinate

```
# specify file, method, distance, and iterations
# default is 20 iterations, can increase trymax to get convergence
ord1 <- phyloseq::ordinate(ps_vst_pos, "NMDS", "bray", trymax=100)
```

```
## Wisconsin double standardization
## Run 0 stress 0.1200766
## Run 1 stress 0.1200766
## ... New best solution
## ... Procrustes: rmse 1.070266e-06  max resid 3.378199e-06
## ... Similar to previous best
## Run 2 stress 0.1200766
```

```
## ... Procrustes: rmse 1.214815e-05  max resid 4.167747e-05
## ... Similar to previous best
## Run 3 stress 0.1200766
## ... New best solution
## ... Procrustes: rmse 1.133051e-06  max resid 2.934571e-06
## ... Similar to previous best
## Run 4 stress 0.1200766
## ... New best solution
## ... Procrustes: rmse 6.329707e-06  max resid 2.128409e-05
## ... Similar to previous best
## Run 5 stress 0.1200766
## ... New best solution
## ... Procrustes: rmse 4.25763e-06  max resid 1.443057e-05
## ... Similar to previous best
## Run 6 stress 0.1200766
## ... Procrustes: rmse 1.19913e-06  max resid 3.363577e-06
## ... Similar to previous best
## Run 7 stress 0.1604021
## Run 8 stress 0.1200766
## ... Procrustes: rmse 7.251092e-06  max resid 2.128864e-05
## ... Similar to previous best
## Run 9 stress 0.1200766
## ... Procrustes: rmse 9.375244e-06  max resid 3.217513e-05
## ... Similar to previous best
## Run 10 stress 0.1200766
## ... Procrustes: rmse 5.182307e-06  max resid 1.575221e-05
## ... Similar to previous best
## Run 11 stress 0.2038901
## Run 12 stress 0.1976849
## Run 13 stress 0.1200766
## ... Procrustes: rmse 6.39098e-07  max resid 2.307583e-06
## ... Similar to previous best
## Run 14 stress 0.1200766
## ... Procrustes: rmse 5.705836e-07  max resid 1.285638e-06
## ... Similar to previous best
## Run 15 stress 0.1200766
## ... Procrustes: rmse 2.313998e-06  max resid 5.56658e-06
## ... Similar to previous best
## Run 16 stress 0.1200766
## ... Procrustes: rmse 5.734169e-06  max resid 1.961336e-05
## ... Similar to previous best
## Run 17 stress 0.2061679
## Run 18 stress 0.1604021
## Run 19 stress 0.1200766
## ... Procrustes: rmse 1.167875e-06  max resid 3.08696e-06
## ... Similar to previous best
## Run 20 stress 0.1200766
## ... Procrustes: rmse 3.377951e-06  max resid 1.000983e-05
## ... Similar to previous best
## *** Solution reached
```

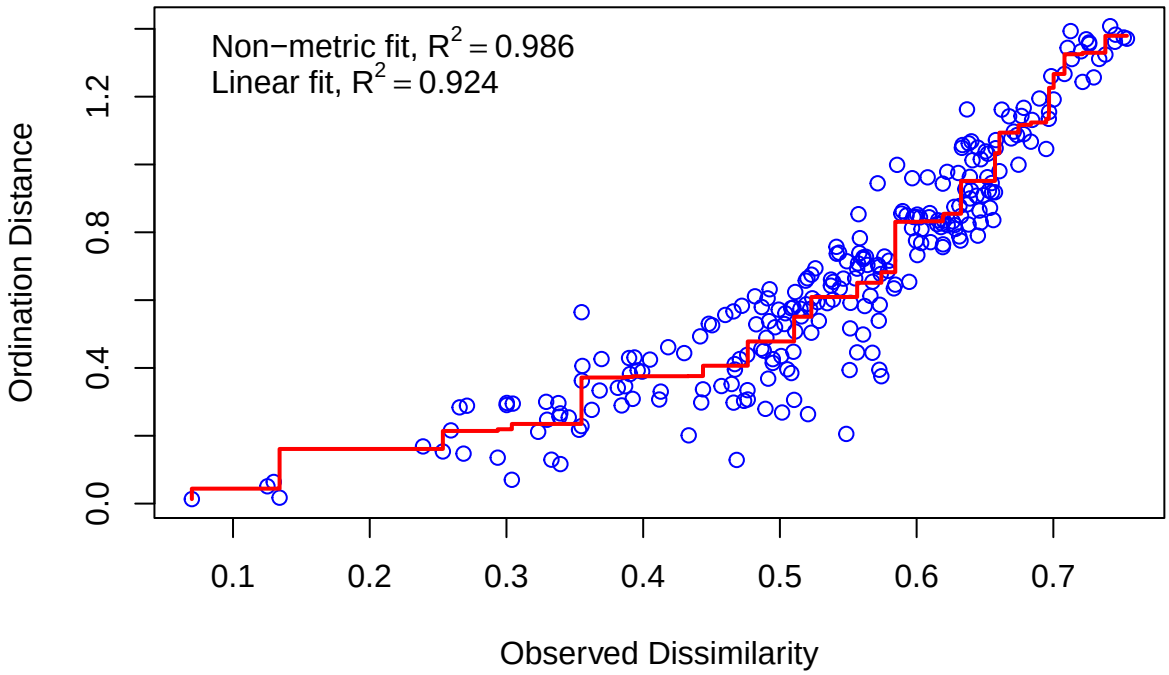
```
ord1
```

```
##
```

```
## Call:
## metaMDS(comm = veganifyOTU(physeq), distance = distance, trymax = 100)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     wisconsin(veganifyOTU(physeq))
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1200766
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'wisconsin(veganifyOTU(physeq))'
```

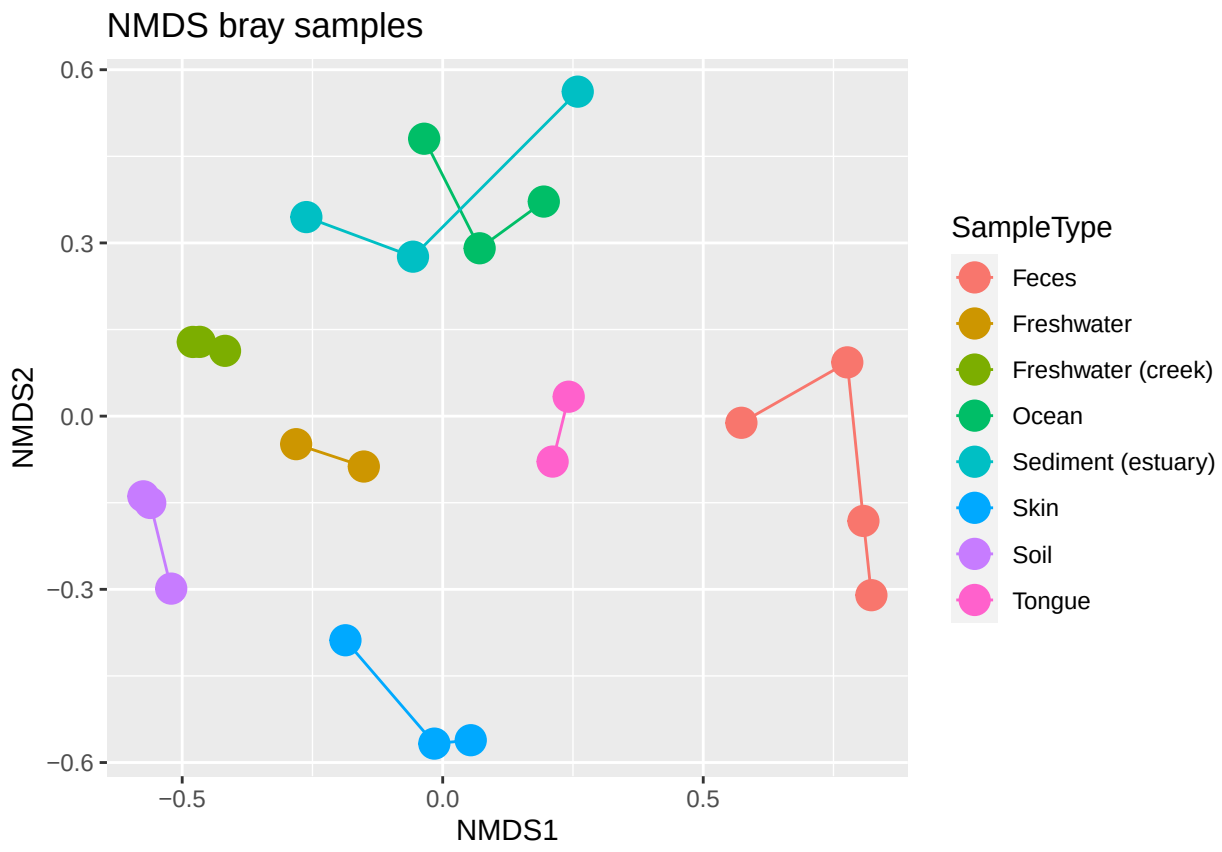
Evaluate NMDS run with stress

```
# shows relationship between actual dissimilarities and ordination distances
# if highly correlated, stress is low and ordination is a good representation of data
# if poorly correlated (large scatter), ordination is not representative of original distances
vegan::stressplot(ord1)
```



Basic ordination plot by samples

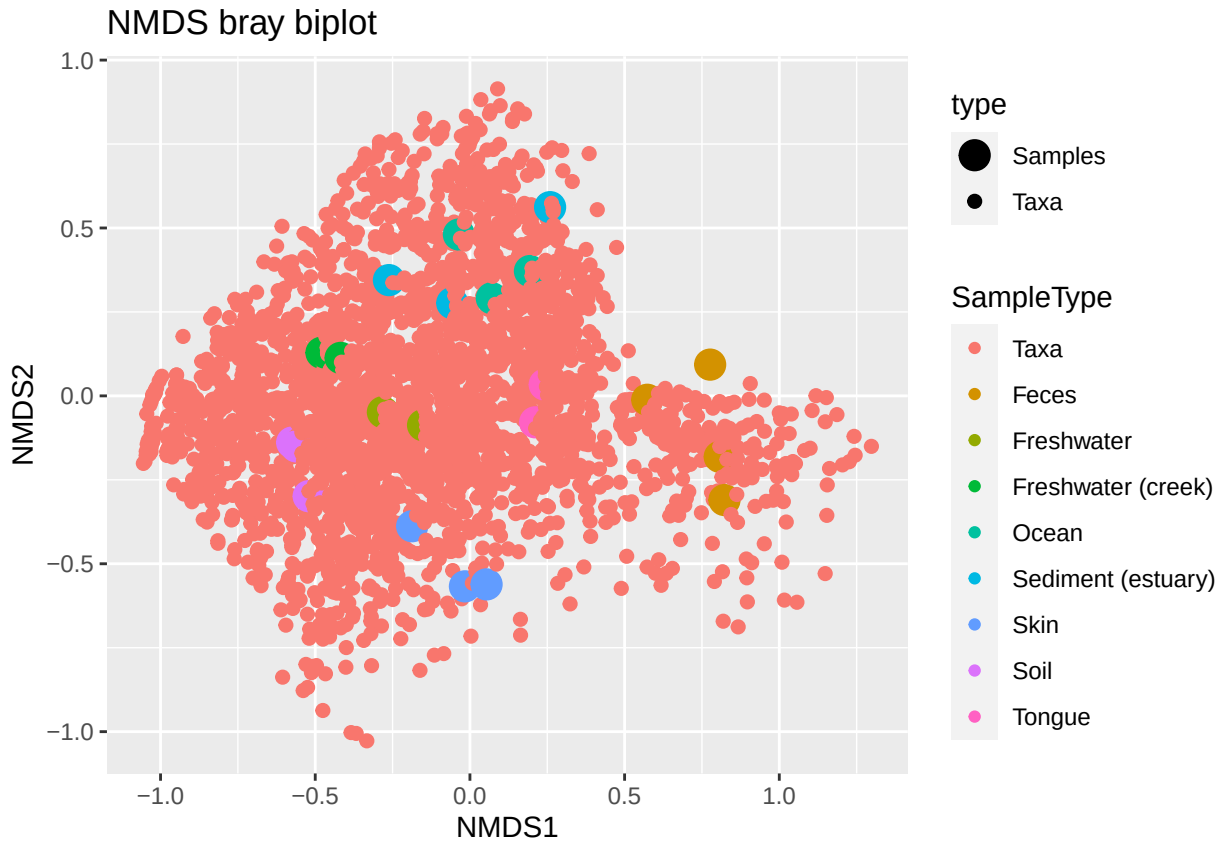
```
phyloseq::plot_ordination(ps_vst_pos, ord1,
                          type="samples",
                          color="SampleType",
                          title="NMDS bray samples") +
                          ggplot2::geom_line() +
                          ggplot2::geom_point(size=5)
```




```
# If you want to draw confidence ellipses around the treatments
# add the following to the plot_ordination code above:
#    + ggplot2::stat_ellipse(type="norm", linetype = 2)
# not done here because not enough replicate points for calculation
```

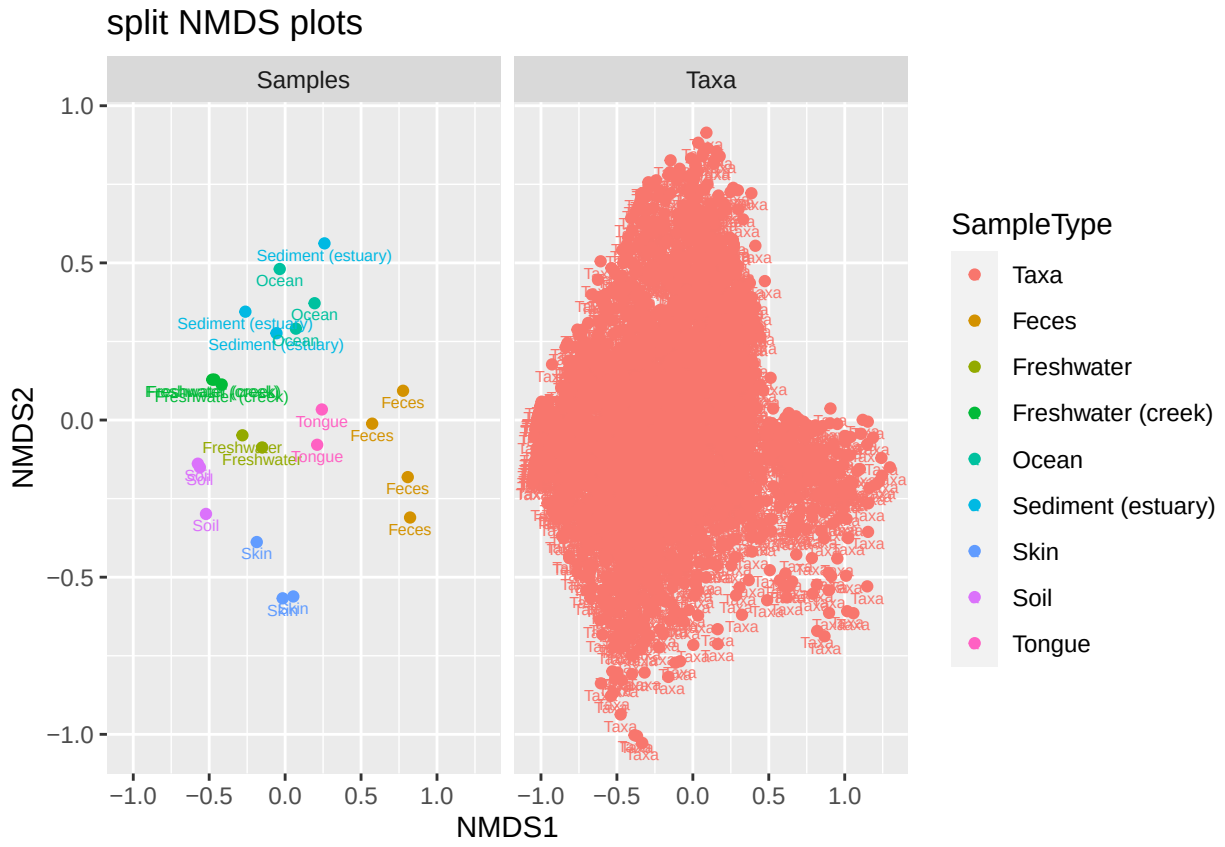
Biplot of samples + taxa

```
phyloseq::plot_ordination(ps_vst_pos, ord1,
                          type="biplot",
                          color="SampleType",
                          title = "NMDS bray biplot")
```



Split plot for samples and taxa

```
phyloseq::plot_ordination(ps_vst_pos, ord1,
                      type="split", color="SampleType",
                      label="SampleType",
                      title="split NMDS plots")
```



Extract NMDS data for further analysis or export

```
# two ways to download the data for further analyses or graphics
# export just the sample names + xy coordinates
write.csv(ord1$points, "wk6_NMDS_sample_xy.csv")
write.csv(ord1$species, "wk6_NMDS_species_xy.csv")
# alternatively export sample names, xy coordinates, and sample data
nmds.bray.xy <- plot_ordination(ps_vst_pos, ord1, justDF = TRUE)
write.csv(nmds.bray.xy, "Wk6_NMDS_xy.csv")
```

NMDS with Bray-Curtis distances + environmental data in vegan::metaMDS

see vegan manual https://cran.r-project.org/web/packages/vegan/vegan.pdf

Prepare otu_table for vegan

```
# make sure otu file has samples as rows and ASVs as columns; transpose if needed
# add constant to remove negative values
# alternatively, could access the otu_table slot in ps_vst_pos
otu_vst<- t(vst)
min(otu_vst)
```

```
## [1] -2.222172
```

```
otu_vst <- otu_vst+2.23
```

Run NMDS with vegan::metaMDS
https://www.rdocumentation.org/packages/vegan/versions/2.4-2/topics/metaMDS

```
# specify community file, distance, and autotransform = FALSE to avoid automated vegan transformations
ord2 <- vegan::metaMDS(otu_vst, distance = "bray", autotransform = FALSE, trymax=20)
```

```
## Run 0 stress 0.1327717
## Run 1 stress 0.1327717
## ... New best solution
## ... Procrustes: rmse 1.582796e-06  max resid 4.220893e-06
## ... Similar to previous best
## Run 2 stress 0.1327717
## ... Procrustes: rmse 5.280501e-06  max resid 1.561165e-05
## ... Similar to previous best
## Run 3 stress 0.1456786
## Run 4 stress 0.2343173
## Run 5 stress 0.1327717
## ... New best solution
## ... Procrustes: rmse 2.500777e-06  max resid 7.671826e-06
## ... Similar to previous best
## Run 6 stress 0.1327717
## ... Procrustes: rmse 2.59698e-06  max resid 8.538269e-06
## ... Similar to previous best
## Run 7 stress 0.1327717
## ... Procrustes: rmse 2.898936e-06  max resid 7.988699e-06
## ... Similar to previous best
## Run 8 stress 0.2142169
## Run 9 stress 0.1327717
## ... Procrustes: rmse 3.351694e-06  max resid 9.846088e-06
## ... Similar to previous best
## Run 10 stress 0.1327717
## ... Procrustes: rmse 3.139279e-06  max resid 9.583544e-06
## ... Similar to previous best
## Run 11 stress 0.1327717
```

```
## ... Procrustes: rmse 1.075795e-06  max resid 2.379346e-06
## ... Similar to previous best
## Run 12 stress 0.1327717
## ... Procrustes: rmse 1.708688e-06  max resid 5.01533e-06
## ... Similar to previous best
## Run 13 stress 0.1327717
## ... Procrustes: rmse 2.186547e-06  max resid 7.827268e-06
## ... Similar to previous best
## Run 14 stress 0.1327717
## ... Procrustes: rmse 6.823054e-07  max resid 2.011973e-06
## ... Similar to previous best
## Run 15 stress 0.1456787
## Run 16 stress 0.1327717
## ... Procrustes: rmse 8.026205e-07  max resid 2.015886e-06
## ... Similar to previous best
## Run 17 stress 0.2095181
## Run 18 stress 0.1890761
## Run 19 stress 0.1327717
## ... Procrustes: rmse 1.286507e-06  max resid 3.161932e-06
## ... Similar to previous best
## Run 20 stress 0.2182847
## *** Solution reached
```

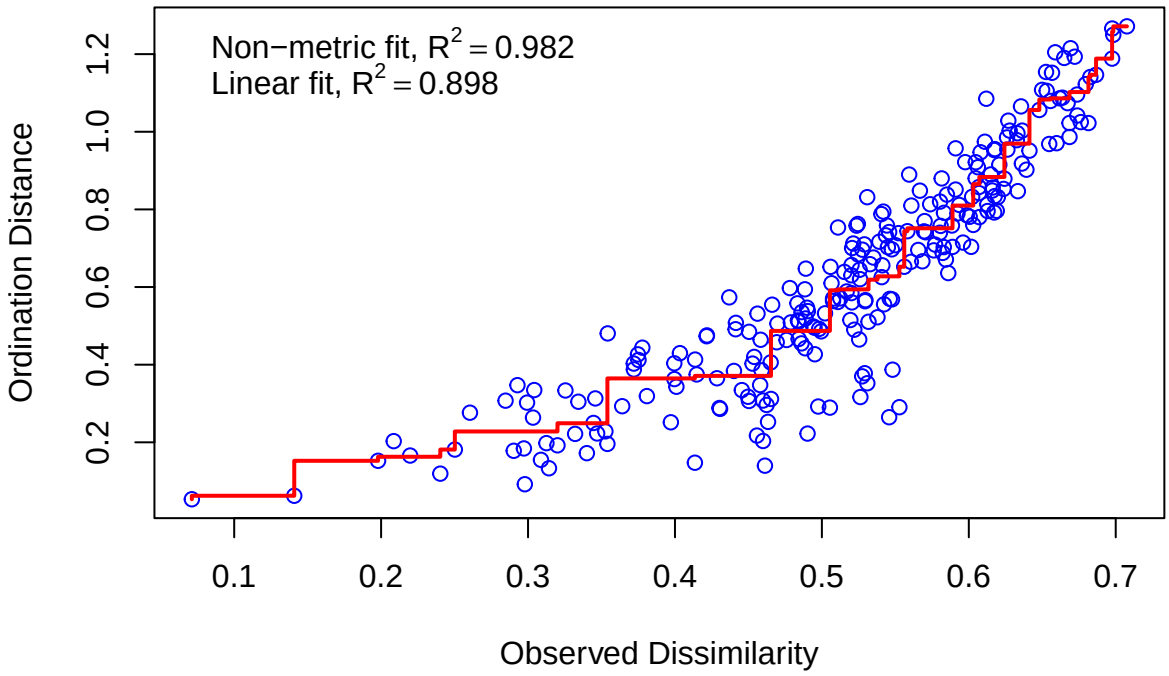
```
ord2
```

```
##
## Call:
## vegan::metaMDS(comm = otu_vst, distance = "bray", trymax = 20,      autotransform = FALSE)
##
## global Multidimensional Scaling using monoMDS
##
## Data:     otu_vst
## Distance: bray
##
## Dimensions: 2
## Stress:     0.1327717
## Stress type 1, weak ties
## Two convergent solutions found after 20 tries
## Scaling: centring, PC rotation, halfchange scaling
## Species: expanded scores based on 'otu_vst'
```

```
# what to do if you don't have convergence?
# extend number of random starts by specifying "trymax" > default = 20
# start a new ordination using the previous run as the start with "previous.best" to avoid local optima
# increase max iterations with "maxit"
# consider a different data transformation
```

Evaluate ordination fit with stress

```
vegan::stressplot(ord2)
```



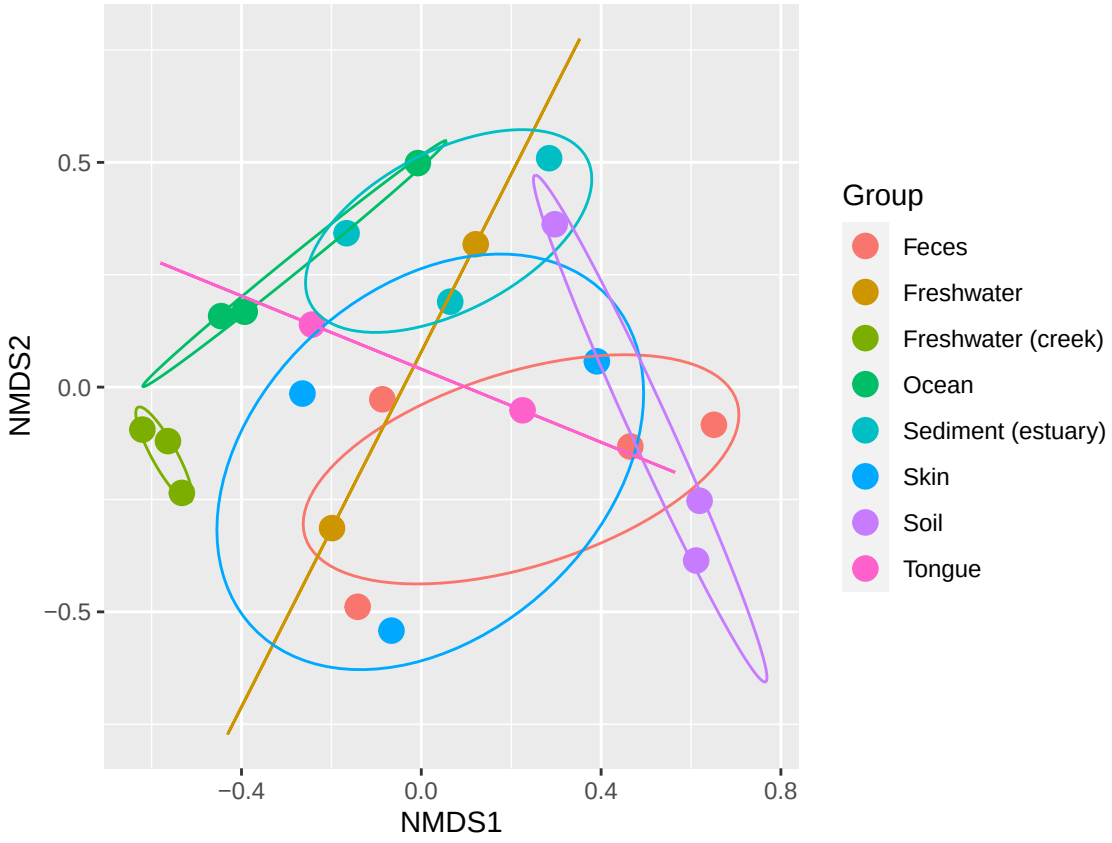
Evaluate environmental correlations to NMDS with vegan::envfit
https://www.rdocumentation.org/packages/vegan/versions/2.4-2/topics/envfit

```
# using the sam.new file because it is limited to the quantitative vars
# alternatively could specify the variables in the file to use
# note that this acts on ord object, not original otu file
ord2_env <- vegan::envfit(ord2, sam.new, permutations = 99, strata = NULL, choices=c(1,2))
ord2_env
```

```
##
## ***VECTORS
##
##             NMDS1    NMDS2     r2 Pr(>r)
## pH       -0.15554  0.98783 0.4913   0.01 **
## salinity  0.47522 -0.87987 0.1070   0.37
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Permutation: free
## Number of permutations: 99
```

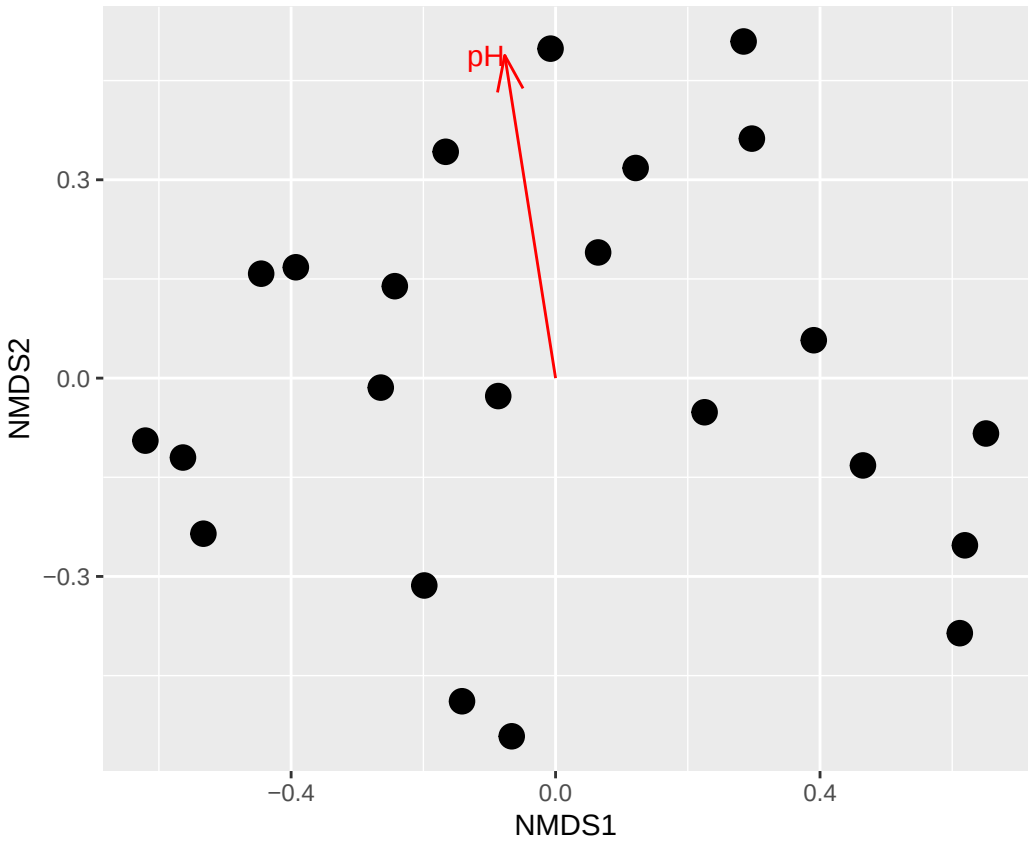
Plot with confidence ellipses around treatments
https://rdrr.io/github/jfq3/ggordiplots/man/gg_ordiplot.html

```
# groups = defines the point groupings based on column in sample data file
# choice = axes to plot
# kind = "sd", "se", or "ehull"
  # sd = standard deviation of points
  # se = standard deviations of averages
  # ehull = ellipsoid hull, minimum boundary around the points
# conf = confidence limits for ellipses, multiplies sd or se by appropriate value
ggordiplots::gg_ordiplot(ord2, groups=sam.all$SampleType, choices = c(1,2), kind = "se", conf = 0.95, p
```



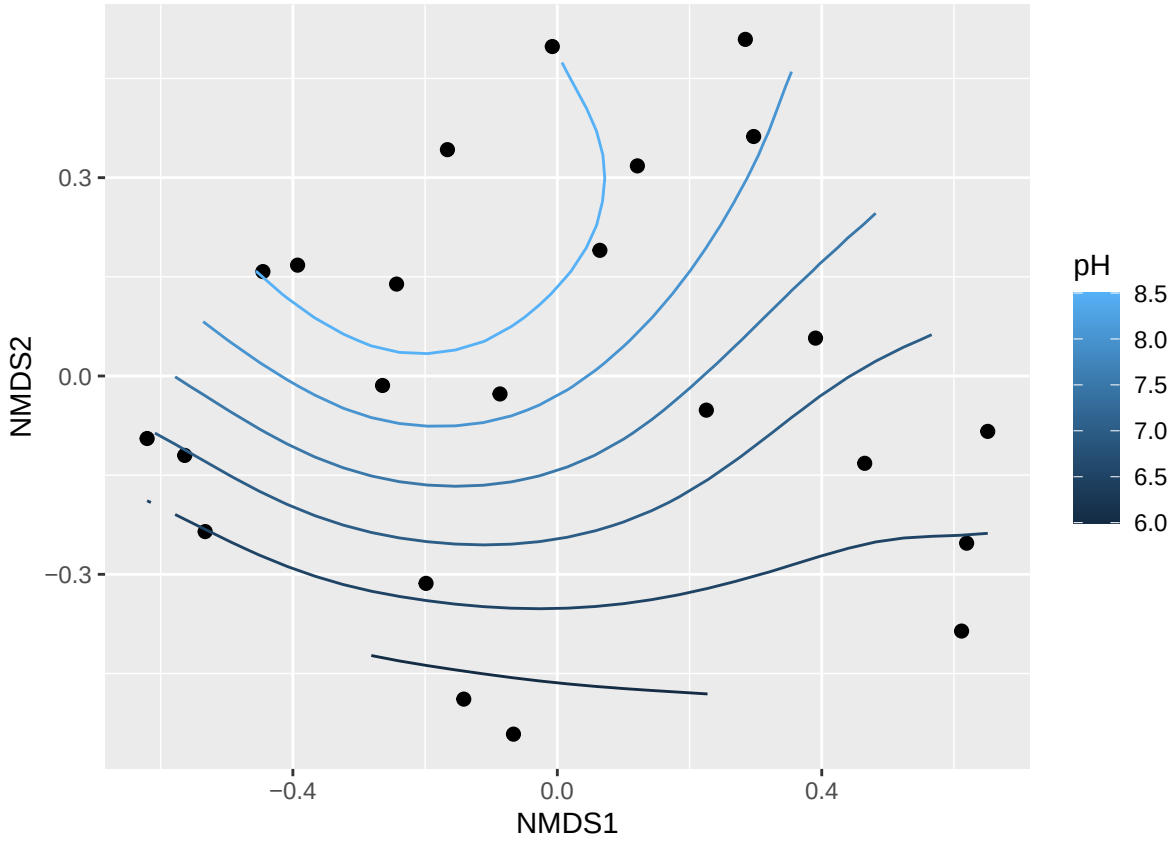
Plot with vectors of environmental variables to the ordination plot
https://rdrr.io/github/jfq3/ggordiplots/man/gg_envfit.html

```
# alpha controls what vectors show up based on sig
ggordiplots::gg_envfit(ord=ord2, env=sam.new, perm=99, pt.size=4, alpha= 0.05)
```



Create contour plots for important environmental factors
https://rdrr.io/github/jfq3/ggordiplots/man/gg_ordisurf.html

```
# define the countours using env.var to select a continuous variable from the sample data file
# change bin width to adjust size of contours
ggordiplots::gg_ordisurf(ord=ord2, env.var=sam.new$pH, choices = c(1,2), binwidth=0.5, pt.size=2, var.la
```



Extract the data for use elsewhere as needed

```
sampleScores <- ord2$points
otuScores <- ord2$species
envScores <- vegan::scores(ord2_env, "vectors")
envCorrels <- data.frame(r=ord2_env$vectors$r, p=ord2_env$vectors$pvals)
```

PHYLOGENETIC BETA DIVERSITY

Phylogenetic Isometric Log Ratio Transformation for Compositional Data

We'll use the philr package. For more info:
https://bioconductor.org/packages/release/bioc/vignettes/philr/inst/doc/philr-intro.html
https://bioconductor.org/packages/devel/bioc/vignettes/philr/inst/doc/philr-intro.html#transform-data-using-philr

Prepare ASV matrix and tree file for PhILR transformation

```
# check min value and use pseudocount to avoid log-ratios of zero counts
ASV <- otu_table(ps_gp_bact)
min(ASV)
```

```
## [1] 0
```

```
ASV <- ASV+1
min(ASV)
```

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

```
ASV <- as.matrix(ASV)

TREE <- phy_tree(ps_gp_bact)
TAX <- tax_table(ps_gp_bact)
SAM <- sample_data(ps_gp_bact)
# check that phylogenetic tree is rooted and binary
ape::is.rooted(TREE)
```

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

```
ape::is.binary(TREE)
```

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

```
# name tree internal nodes
TREE <- ape::makeNodeLabel(TREE, method="number", prefix="n")

# resolve consensus names
philr::name.balance(TREE, TAX, "n1")
```

```
## [1] "Kingdom_Bacteria/Phylum_Firmicutes"
```

PhILR transform ASV matrix

```
# philr requires taxa as columns and samples as rows
# colnames(ASV)
ASV <- t(ASV)
# row.names(ASV)

# philr transform - unweighted
ASV_ilr <- philr(ASV, TREE)
ASV_ilr[1:4,1:4]
```

```
##                n1       n2          n3         n4
## CL3      13.21092 4.033208  -6.6685031 -35.066980
## CC1      27.49177 3.117465  -8.4287634 -29.430452
## SV1      13.81909 1.476817 -12.7331482  -7.277954
## M31Fcsw -64.83814 1.994201   0.9462501   2.676352
```

```
# philr transform - weighted
# part.weights = weights for ASVs; here by geom mean of counts across samples * Euclidean norm of relat
# ilr.weights = weights for branch lengths; here uniform = no weight
ASV_ilrw <- philr(ASV, TREE,
                  part.weights="enorm.x.gm.counts",
                  ilr.weights="uniform")
ASV_ilrw[1:4,1:4]
```

```
##                 n1        n2          n3          n4
## CL3       31.22759 -6.288647 -23.4255874 -102.987493
## CC1       58.11957 -9.153316 -28.1552306 -104.610556
## SV1       40.04463 -7.284178 -30.5433647  -52.776944
## M31Fcsw -177.81364  2.571159  -0.1658274    2.858056
```

Use PhILR-transformed data for ordination with Euclidean distance

```
# here using phyloseq (could also use vegan etc.)

# with unweighted philr transform
dist_euc <- dist(ASV_ilr, method="euclidean")
ord3 <- phyloseq::ordinate(ps_gp_bact, 'PCoA', distance=dist_euc)
p3 <- phyloseq::plot_ordination(ps_gp_bact, ord3, type="samples", color="SampleType") +
  geom_point(size=4) +
  theme(legend.title = element_blank()) +
  ggtitle("unweighted philr")

# with weighted philr transform
dist_eucw <- dist(ASV_ilrw, method="euclidean")
ord4 <- phyloseq::ordinate(ps_gp_bact, 'PCoA', distance=dist_eucw)
p4 <- phyloseq::plot_ordination(ps_gp_bact, ord4, type="samples", color="SampleType") +
  geom_point(size=4) +
  theme(legend.title = element_blank()) +
  ggtitle("weighted philr")
```

Phylogenetic distance/dissimilarity

Most common metric is UniFrac, which measures among-community equivalent of Faith's PD. Calculated as the ratio of unshared to total branch length between taxa in two samples. Uses rooted trees.

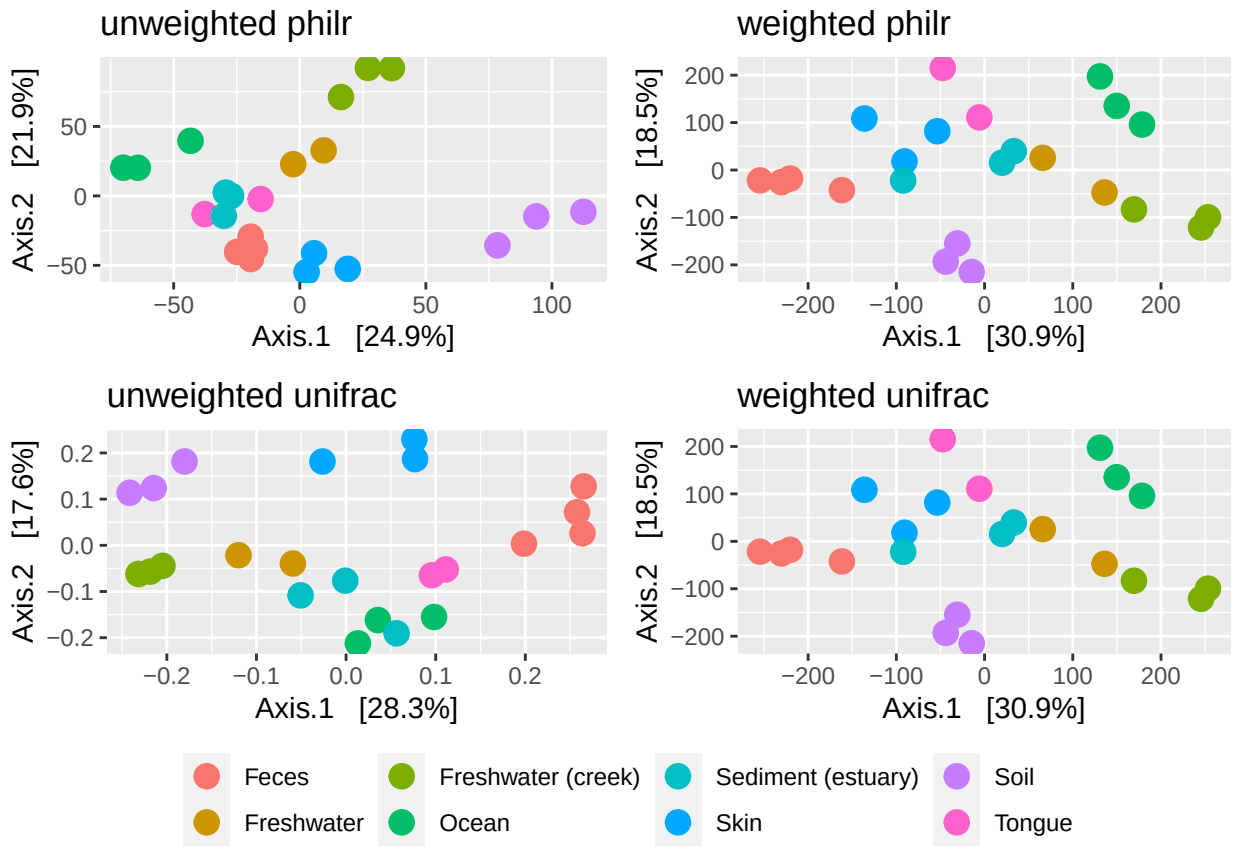
Unifrac distance in phyloseq and PCoA

```
# unweighted - considers only taxa presence/absence
# typically need to rarefy data for unweighted unifrac
# can use the following for this purpose, but for today we'll leave it as is to compare
# ps_gp_rar <- phyloseq::rarefy_even_depth(ps_gp_bact, sample.size=min(sample_sums(ps_gp_bact), rngseed
dist_ufu <- phyloseq::UniFrac(ps_gp_bact, weighted=FALSE)
ord5 <- phyloseq::ordinate(ps_gp_bact, 'PCoA', distance=dist_ufu)
p5<- phyloseq::plot_ordination(ps_gp_bact, ord5, type="samples", color="SampleType") +
  geom_point(size=4) +
  theme(legend.title = element_blank()) +
  ggtitle("unweighted unifrac")

# weighted - considers taxa abundance
dist_ufw <- phyloseq::UniFrac(ps_gp_bact, weighted=TRUE) # alt: dist_ufw2 <- distance(ps6, "wunifrac")
ord6 <- phyloseq::ordinate(ps_gp_bact, "PCoA", distance=dist_ufw)
p6 <- phyloseq::plot_ordination(ps_gp_bact, ord4, type="samples", color="SampleType") +
  geom_point(size=4) +
  theme(legend.title = element_blank()) +
  ggtitle("weighted unifrac")
```

Plot all ordinations

```
ggpubr::ggarrange(p3, p4, p5, p6, ncol=2, nrow=2, common.legend = TRUE, legend="bottom")
```


unweighted philr
Axis.2 [21.9%]
Axis.1 [24.9%]
weighted philr
Axis.2 [18.5%]
Axis.1 [30.9%]
unweighted unifrac
Axis.2 [17.6%]
Axis.1 [28.3%]
weighted unifrac
Axis.2 [18.5%]
Axis.1 [30.9%]
Feces
Freshwater
Freshwater (creek)
Ocean
Sediment (estuary)
Skin
Soil
Tongue

Coding Exercises

Please submit as a knitted html markdown to GitHub due on 2/23

1. run PCoA on clr transformed otus in phyloseq

- Use `microbiome::transform` for clr transform on ps object
  - This function adds a pseudocount if there are zeros in data
  - Resulting transform will differ from `compositions::clr`
- Run ordination via `phyloseq::ordinate` with option `PCoA`
  - Use euclidean distance on clr == Aitchison's distance
  - Note: alternatively calculate Aitchison's distance via `robCompositions::aDist function`
- Use `phyloseq::plot_scree` to evaluate variance explained by each axis
- Plot ordination results
- Access sample scores and eigenvalues for export

2. run PCoA with clr transformed otus + environmental data using an alternative method

- Select a method for PCoA outside of phyloseq
  - Examples: `vegan::wcmdscale`,`vegan::rda`, `FactoMineR::PCA`, `ade4::dudi.pca`, `stats::prcomp`, `stats::princomp`, `ecodist::pco`, `ape::pcoa`
- Examine eigenvalues
- Plot results
- Describe how the clr-based ordination results here and in #2 above differ from the vst results and what this means for analysis
- Access sample scores and eigenvalues for export

3. Examine beta-diversity in a phylogenetic context using DPCoA

- This will take ~10 min to run
  - if longer, consider further reducing GP dataset to top 100-200 taxa
- Analyze the Global Patterns bacteria data with DPCoA in `phyloseq::DPCoA`
  - https://rdrr.io/bioc/phyloseq/man/DPCoA.html

- Access the resulting list object using `$` to get the Axis 1 and 2 coordinates
- Use `data.frame` to combine the following into one file for plotting:
  - sample names from `SAM$X.SampleID`
  - Axis1 and Axis2 coordinates
  - sample types from `SAM$SampleType`
- Plot eigenvalues by axis with `phyloseq::plot_scree`
- Plot results by samples with `plot_ordination` using `color="SampleType"`
- Plot results by species with `plot_ordination` using `color="Phylum"`

Session Info

```
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] ggordiplots_0.4.0          glue_1.6.0
##  [3] ggpubr_0.4.0               compositions_2.0-4
##  [5] microbiome_1.16.0          philr_1.20.1
##  [7] ape_5.6-1                  vegan_2.5-7
##  [9] lattice_0.20-45            permute_0.9-7
## [11] DESeq2_1.34.0              SummarizedExperiment_1.24.0
## [13] Biobase_2.54.0             MatrixGenerics_1.6.0
## [15] matrixStats_0.61.0         GenomicRanges_1.46.1
## [17] GenomeInfoDb_1.30.0        IRanges_2.28.0
## [19] S4Vectors_0.32.3           BiocGenerics_0.40.0
## [21] forcats_0.5.1              stringr_1.4.0
## [23] dplyr_1.0.7                purrr_0.3.4
## [25] readr_2.1.1                tidyr_1.1.4
## [27] tibble_3.1.6               ggplot2_3.3.5
## [29] tidyverse_1.3.1            phyloseq_1.38.0
##
## loaded via a namespace (and not attached):
##   [1] readxl_1.3.1           backports_1.4.1        fastmatch_1.1-3
##   [4] plyr_1.8.6             igraph_1.2.11          lazyeval_0.2.2
##   [7] splines_4.1.2          BiocParallel_1.28.3    digest_0.6.29
##  [10] foreach_1.5.1          yulab.utils_0.0.4      htmltools_0.5.2
##  [13] fansi_0.5.0            magrittr_2.0.1         memoise_2.0.1
##  [16] cluster_2.1.2          tzdb_0.2.0             Biostrings_2.62.0
##  [19] annotate_1.72.0        modelr_0.1.8           bayesm_3.1-4
##  [22] colorspace_2.0-2       blob_1.2.2             rvest_1.0.2
##  [25] haven_2.4.3            xfun_0.29              crayon_1.4.2
##  [28] RCurl_1.98-1.5         jsonlite_1.7.3         genefilter_1.76.0
```

```
##  [31] survival_3.2-13        phangorn_2.8.1         iterators_1.0.13
##  [34] gtable_0.3.0           zlibbioc_1.40.0        XVector_0.34.0
##  [37] DelayedArray_0.20.0    car_3.0-12             Rhdf5lib_1.16.0
##  [40] DEoptimR_1.0-10        abind_1.4-5            scales_1.1.1
##  [43] DBI_1.1.2              rstatix_0.7.0          Rcpp_1.0.8
##  [46] isoband_0.2.5          xtable_1.8-4           gridGraphics_0.5-1
##  [49] tidytree_0.3.7         bit_4.0.4              httr_1.4.2
##  [52] RColorBrewer_1.1-2     ellipsis_0.3.2         farver_2.1.0
##  [55] pkgconfig_2.0.3        XML_3.99-0.8           dbplyr_2.1.1
##  [58] locfit_1.5-9.4         utf8_1.2.2             labeling_0.4.2
##  [61] ggplotify_0.1.0        tidyselect_1.1.1       rlang_0.4.12
##  [64] reshape2_1.4.4         AnnotationDbi_1.56.2   munsell_0.5.0
##  [67] cellranger_1.1.0       tools_4.1.2            cachem_1.0.6
##  [70] cli_3.1.1              generics_0.1.2         RSQLite_2.2.9
##  [73] ade4_1.7-18            broom_0.7.11           evaluate_0.14
##  [76] biomformat_1.22.0      fastmap_1.1.0          yaml_2.2.1
##  [79] ggtree_3.2.1           knitr_1.37             bit64_4.0.5
##  [82] fs_1.5.2               robustbase_0.93-9      KEGGREST_1.34.0
##  [85] nlme_3.1-155           aplot_0.1.2            xml2_1.3.3
##  [88] compiler_4.1.2         rstudioapi_0.13        png_0.1-7
##  [91] ggsignif_0.6.3         reprex_2.0.1           treeio_1.18.1
##  [94] geneplotter_1.72.0     stringi_1.7.6          highr_0.9
##  [97] Matrix_1.4-0           tensorA_0.36.2         multtest_2.50.0
## [100] vctrs_0.3.8            pillar_1.7.0           lifecycle_1.0.1
## [103] rhdf5filters_1.6.0     cowplot_1.1.1          data.table_1.14.2
## [106] bitops_1.0-7           patchwork_1.1.1        R6_2.5.1
## [109] gridExtra_2.3          codetools_0.2-18       MASS_7.3-54
## [112] assertthat_0.2.1       rhdf5_2.38.0           withr_2.4.3
## [115] GenomeInfoDbData_1.2.7 mgcv_1.8-38            parallel_4.1.2
## [118] hms_1.1.1              quadprog_1.5-8         grid_4.1.2
## [121] ggfun_0.0.5            rmarkdown_2.11         carData_3.0-5
## [124] Rtsne_0.15             lubridate_1.8.0
```