FAST method

Kim Dill-McFarland

Oct 4, 2017

Introduction

Convenient, reproducible, and rapid preservation of unique biological samples is pivotal to their use in microbiome analyses. As an increasing number of human longitudinal studies are looking to incorporate human microbiome data, the need for such sample collection and storage methods is high. Here, we describe the Fecal Aliquot Straw Technique (FAST) of fecal sample processing for long-term storage.

For 8 subjects (Sub.1-Sub.8), samples types are:

- Straw 1 (S1): straw 1 sample resulting from FAST method
- Straw 2 (S1): gavage of straw 2 sample used in humanization of mice
- Mouse fecals (F1-3): mouse fecal samples from mice humanized with different donor Subjects (N = 3 per Subject)
    - Collectively referenced as mice (M)

Data

Load packages

```
#Manipulation of tables
library(tidyr)
library(reshape2)
#Community ecology package (Bray-Curtis, Jaccard)
library(vegan)
#Community ecology package (UniFrac)
library(phyloseq)
#Linear and nonlinear mixed effects models
library(nlme)
#Pairwise comparisons within nlme models
library(multcomp)
#Plots and visualization
library(ggplot2)
#Plotting multiple ggplot2 plots in one frame
library(cowplot)
#Venn diagrams
library(venneuler)
```

Load functions

```
#Pairwise PERMANOVAs
source("Functions/Pairwise_adonis_all.r")
```

```r
#Function to identify taxa shared by S1 and S1 or S2 and M for each subject. Taxon must be at least 0.1
#Creates input for mean.counts
source("Functions/venn.abund.all.r")
#Function to calculate percentages of taxa recovered across FAST (counts from venn.abund.all) and means
#Fig. 2B & 3B
source("Functions/mean.counts.r")

#Function to identify taxa shared by S1 and S1 or S2 and M for each subject. No abundance cutoff
#Creates input for venn.perc
source("Functions/venn.all.r")
#Function to pull shared OTUs (determined by venn.all function) out of OTU table and calculate percent
#Fig. 2C & 3D
source("Functions/venn.perc.r")

#To obtain lists of beta-diversity values of mouse samples compared to samples from their matched subje
source("Functions/donor.other.r")
```

Load data

```r
#Un-rarified OTU table from mothur
OTU = read.table("Data/WLS.mouse.final.opti_mcc.unique_list.shared", sep="\t", header=TRUE)
  row.names(OTU) = OTU$Group
  OTU = OTU[,!names(OTU) %in% c("label","Group","numOtus")]

#Rarified OTU table from mothur
OTU.norm = read.table("Data/WLS.mouse.final.opti_mcc.unique_list.0.03.norm.shared", sep="\t", header=TRU
  row.names(OTU.norm) = OTU.norm$Group
  OTU.norm = OTU.norm[,!names(OTU.norm) %in% c("label","Group","numOtus")]

#Metadata and alpha-diversity measures
meta = read.table("Data/Metadata.txt", sep="\t", header=TRUE, row.names=1)

#Taxonomy of OTUs from mothur
tax = read.table("Data/WLS.mouse.final.opti_mcc.unique_list.0.03.cons.taxonomy", sep="\t", header=TRUE)
  row.names(tax) = tax$OTU
  tax = separate(tax, Taxonomy, into = c("Domain", "Phylum", "Class", "Order", "Family", "Genus", "Spec
  tax = tax[,!(names(tax) %in% c("Size", "Strain", "OTU"))]

#Taxa tables: OTUs summed at difference taxonomic levels
phyla = as.data.frame(t(read.table("Data/phyla_unrare.txt", sep="\t", header=TRUE, row.names=1)))
order = as.data.frame(t(read.table("Data/order_unrare.txt", sep="\t", header=TRUE, row.names=1)))
family = as.data.frame(t(read.table("Data/family_unrare.txt", sep="\t", header=TRUE, row.names=1)))
genus = as.data.frame(t(read.table("Data/genus_unrare.txt", sep="\t", header=TRUE, row.names=1)))
```

Order the data so that all tables have rows in the same order

```r
#Order all by samples name
OTU = OTU[order(row.names(OTU)),]
OTU.norm = OTU.norm[order(row.names(OTU.norm)),]
meta = meta[order(row.names(meta)),]
phyla = phyla[order(row.names(phyla)),]
order = order[order(row.names(order)),]
```

```
family = family[order(row.names(family)),]
genus = genus[order(row.names(genus)),]
```

Calculate OTU precent relative abundance

```
abund=OTU/rowSums(OTU)*100
```

Data manipulation

Phyloseq object

All samples

```
#Load rarified data into phyloseq objects
OTU.UF = otu_table(as.matrix(OTU.norm), taxa_are_rows=FALSE)
Tax.UF = tax_table(as.matrix(tax))
meta.UF = sample_data(meta)

#Load pre-calculated neighbor-joining tree
load("Data/NJ.tree.Rdata")

#Merge into phyloseq object
physeq.tree = merge_phyloseq(OTU.UF, Tax.UF, meta.UF, NJ.tree)
physeq.tree
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1053 taxa and 40 samples ]
## sample_data() Sample Data:       [ 40 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1053 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1053 tips and 1051 internal nodes ]
```

Mouse-derived samples only

```
OTU.norm.M = OTU.norm[meta$Type=="F",]
OTU.UF.M = otu_table(as.matrix(OTU.norm.M), taxa_are_rows=FALSE)
meta.UF.M = sample_data(meta[meta$Type=="F",])

physeq.tree.M = merge_phyloseq(OTU.UF.M, Tax.UF, meta.UF.M, NJ.tree)
physeq.tree.M
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1053 taxa and 24 samples ]
## sample_data() Sample Data:       [ 24 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1053 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1053 tips and 1051 internal nodes ]
```

Human-derived samples only

```
OTU.norm.H = OTU.norm[meta$Type !="F",]
OTU.UF.H = otu_table(as.matrix(OTU.norm.H), taxa_are_rows=FALSE)
meta.UF.H = sample_data(meta[meta$Type!="F",])
```

```
physeq.tree.H = merge_phyloseq(OTU.UF.H, Tax.UF, meta.UF.H, NJ.tree)
physeq.tree.H
```

```
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1053 taxa and 16 samples ]
## sample_data() Sample Data:       [ 16 samples by 6 sample variables ]
## tax_table()   Taxonomy Table:    [ 1053 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1053 tips and 1051 internal nodes ]
```

Calculate beta-diversity

All samples

```
set.seed(35426)
#Bray-Curtis
BC.dist=as.matrix(vegdist(OTU.norm, distance="bray"))
#Jaccard
J.dist=(BC.dist*2)/(BC.dist+1)
#Weighted UniFrac
wUF.dist = as.matrix(UniFrac(physeq.tree, weighted=TRUE, normalized=TRUE))
```

```
## Warning in UniFrac(physeq.tree, weighted = TRUE, normalized = TRUE):
## Randomly assigning root as -- Otu0220 -- in the phylogenetic tree in the
## data you provided.
```

```
#Unweighted UniFrac
uwUF.dist = as.matrix(UniFrac(physeq.tree, weighted=FALSE, normalized=TRUE))
```

```
## Warning in UniFrac(physeq.tree, weighted = FALSE, normalized = TRUE):
## Randomly assigning root as -- Otu0971 -- in the phylogenetic tree in the
## data you provided.
```

Calculate nMDS

For use in Fig. 2A, 3A, & S3

```
set.seed(35426)
#All samples
uwUF.nmds = ordinate(physeq.tree, method="NMDS", distance="unifrac", weighted=FALSE)
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu0220 --
## in the phylogenetic tree in the data you provided.
```

```
## Run 0 stress 0.1923516
## Run 1 stress 0.1916002
## ... New best solution
## ... Procrustes: rmse 0.01917848  max resid 0.07604131
## Run 2 stress 0.1923019
## Run 3 stress 0.1923489
## Run 4 stress 0.1903278
## ... New best solution
## ... Procrustes: rmse 0.04163269  max resid 0.125549
## Run 5 stress 0.1936938
## Run 6 stress 0.2044715
## Run 7 stress 0.1934726
## Run 8 stress 0.1965697
## Run 9 stress 0.1903898
## ... Procrustes: rmse 0.008550685  max resid 0.03240829
## Run 10 stress 0.1928315
## Run 11 stress 0.1934726
## Run 12 stress 0.1928306
## Run 13 stress 0.1919134
## Run 14 stress 0.2180898
## Run 15 stress 0.1934723
## Run 16 stress 0.1965697
## Run 17 stress 0.1903896
## ... Procrustes: rmse 0.008549486  max resid 0.03253131
## Run 18 stress 0.1919138
## Run 19 stress 0.1934723
## Run 20 stress 0.2106225
## *** No convergence -- monoMDS stopping criteria:
##     20: stress ratio > sratmax
```

```
#Human-derived samples
uwUF.nmds.H = ordinate(physeq.tree.H, method="NMDS", distance="unifrac", weighted=FALSE)
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu1347 --
## in the phylogenetic tree in the data you provided.
```

```
## Run 0 stress 0.07758628
## Run 1 stress 0.07758628
## ... New best solution
## ... Procrustes: rmse 1.016349e-06  max resid 1.818852e-06
## ... Similar to previous best
## Run 2 stress 0.07758628
## ... Procrustes: rmse 1.855037e-06  max resid 3.731177e-06
## ... Similar to previous best
## Run 3 stress 0.07758628
## ... Procrustes: rmse 2.847258e-06  max resid 6.736027e-06
## ... Similar to previous best
## Run 4 stress 0.07758628
## ... Procrustes: rmse 1.864243e-06  max resid 3.774792e-06
## ... Similar to previous best
## Run 5 stress 0.1273746
## Run 6 stress 0.07758628
## ... Procrustes: rmse 2.681213e-06  max resid 6.224872e-06
## ... Similar to previous best
## Run 7 stress 0.07758628
## ... Procrustes: rmse 7.136046e-07  max resid 1.543963e-06
## ... Similar to previous best
## Run 8 stress 0.07758628
## ... Procrustes: rmse 1.857904e-06  max resid 4.304173e-06
## ... Similar to previous best
## Run 9 stress 0.07758628
## ... Procrustes: rmse 2.478363e-06  max resid 5.897718e-06
## ... Similar to previous best
## Run 10 stress 0.07758628
## ... New best solution
## ... Procrustes: rmse 6.104249e-07  max resid 9.052407e-07
## ... Similar to previous best
## Run 11 stress 0.07758628
## ... Procrustes: rmse 2.437243e-06  max resid 5.624766e-06
## ... Similar to previous best
## Run 12 stress 0.07758628
## ... Procrustes: rmse 6.417563e-07  max resid 1.067141e-06
## ... Similar to previous best
## Run 13 stress 0.1284048
## Run 14 stress 0.1266009
## Run 15 stress 0.07758628
## ... Procrustes: rmse 1.194419e-06  max resid 2.346637e-06
## ... Similar to previous best
## Run 16 stress 0.07758628
## ... Procrustes: rmse 1.984611e-06  max resid 4.014279e-06
## ... Similar to previous best
## Run 17 stress 0.07758628
## ... Procrustes: rmse 1.010234e-06  max resid 2.073943e-06
## ... Similar to previous best
## Run 18 stress 0.07758628
## ... New best solution
## ... Procrustes: rmse 7.348082e-07  max resid 1.629183e-06
## ... Similar to previous best
## Run 19 stress 0.07758628
## ... Procrustes: rmse 1.065407e-06  max resid 2.218184e-06
## ... Similar to previous best
## Run 20 stress 0.07758628
## ... Procrustes: rmse 5.310128e-06  max resid 1.163049e-05
## ... Similar to previous best
## *** Solution reached
```

```
#Mouse-derived samples
uwUF.nmds.M = ordinate(physeq.tree.M, method="NMDS", distance="unifrac", weighted=FALSE)
```

```
## Warning in UniFrac(physeq, ...): Randomly assigning root as -- Otu1033 --
## in the phylogenetic tree in the data you provided.
```

```
## Run 0 stress 0.1525575
## Run 1 stress 0.1525575
## ... Procrustes: rmse 2.141091e-05  max resid 3.653147e-05
## ... Similar to previous best
## Run 2 stress 0.1525626
## ... Procrustes: rmse 0.001123412  max resid 0.00391825
## ... Similar to previous best
## Run 3 stress 0.1525575
## ... Procrustes: rmse 5.135346e-05  max resid 8.780011e-05
## ... Similar to previous best
## Run 4 stress 0.2027345
## Run 5 stress 0.1525576
## ... Procrustes: rmse 8.296127e-05  max resid 0.0001447163
## ... Similar to previous best
## Run 6 stress 0.1525575
## ... Procrustes: rmse 3.620428e-05  max resid 7.569557e-05
## ... Similar to previous best
## Run 7 stress 0.1525627
## ... Procrustes: rmse 0.001164682  max resid 0.004021918
## ... Similar to previous best
## Run 8 stress 0.2114246
## Run 9 stress 0.2123716
## Run 10 stress 0.1994142
## Run 11 stress 0.1525576
## ... Procrustes: rmse 8.318973e-05  max resid 0.0001779203
## ... Similar to previous best
## Run 12 stress 0.1525575
## ... Procrustes: rmse 2.025354e-05  max resid 3.611444e-05
## ... Similar to previous best
## Run 13 stress 0.1525575
## ... Procrustes: rmse 4.223858e-05  max resid 0.000134461
## ... Similar to previous best
## Run 14 stress 0.2145589
## Run 15 stress 0.1525625
## ... Procrustes: rmse 0.001067557  max resid 0.003721356
## ... Similar to previous best
## Run 16 stress 0.1525575
## ... Procrustes: rmse 1.460009e-05  max resid 2.507418e-05
## ... Similar to previous best
## Run 17 stress 0.1525589
## ... Procrustes: rmse 0.0001763994  max resid 0.0005606652
## ... Similar to previous best
## Run 18 stress 0.2027021
## Run 19 stress 0.1525575
## ... Procrustes: rmse 2.071616e-05  max resid 3.414725e-05
## ... Similar to previous best
## Run 20 stress 0.2012624
## *** Solution reached
```

Count taxa recover

For use in Fig. 2B & 3B

Taxa shared by S1 and S1 *or* S2 and M for each subject. Taxon must be at least 0.1% relaive abundance in at least one sample in the comparison for each subject.

Count and taxa lists at each taxonomic level

```
venn.abund.all(OTU, "OTU")
venn.abund.all(genus, "genus")
venn.abund.all(family, "family")
venn.abund.all(order, "order")
venn.abund.all(phyla, "phyla")
```

Read in results tables

```
taxa.OTU = read.table("output_venn.abund.all/OTU.venn.abund.all.count.csv", sep=",", header=FALSE, row.
  colnames(taxa.OTU) = c("count","comp","subject")
taxa.genus = read.table("output_venn.abund.all/genus.venn.abund.all.count.csv", sep=",", header=FALSE, 
  colnames(taxa.genus) = c("count","comp","subject")
taxa.family = read.table("output_venn.abund.all/family.venn.abund.all.count.csv", sep=",", header=FALSE
  colnames(taxa.family) = c("count","comp","subject")
taxa.order = read.table("output_venn.abund.all/order.venn.abund.all.count.csv", sep=",", header=FALSE, 
  colnames(taxa.order) = c("count","comp","subject")
taxa.phyla = read.table("output_venn.abund.all/phyla.venn.abund.all.count.csv", sep=",", header=FALSE, 
  colnames(taxa.phyla) = c("count","comp","subject")
```

Convert counts to percentages of taxa recovered across FAST and means and standard errors of these percentages.

Run across taxonomic levels

```
mean.counts(c("taxa.OTU", "taxa.genus", "taxa.family", "taxa.order", "taxa.phyla"))
```

Read in results

```
taxa.recover = read.table("output_mean.counts/mean.counts.csv", sep=",", header=FALSE, row.names=1)
  colnames(taxa.recover) = c("mean","se","comp","taxa")
#Add variable of taxa in order
  taxa.recover$taxa.ord = ifelse(taxa.recover$taxa == "taxa.OTU", "5.OTU",
                          ifelse(taxa.recover$taxa == "taxa.genus", "4.genus",
                          ifelse(taxa.recover$taxa == "taxa.family", "3.family",
                          ifelse(taxa.recover$taxa == "taxa.order", "2.order",
                          ifelse(taxa.recover$taxa == "taxa.phyla", "1.phyla", NA)))))
```

Relative abundance taxa recover

For Fig. 2C & 3D

How much of straw1 relative abundance occurs in straw2?
How much of straw2 relative abundance occurs in mice?

Identify taxa shared by S1 and S1 or S2 and M for each subject. No abundance cutoff.

```
venn.all(OTU, "OTU")
```

Read in results

```
OTU.list = read.table("output_venn.all/OTU.venn.all.list.csv", sep=",", row.names=1)
  colnames(OTU.list) = c("OTU","comparison","Subject")
```

Pull shared OTUs (determined by venn.all) out of OTU table and calculate percent relative abundance of those shared OTUs

```
venn.perc(OTU)
```

Read in results

```
venn.perc = read.csv("output_venn.perc/venn.perc.csv", header=FALSE, row.names=1)
  colnames(venn.perc) = c("Percent","Comparison","Group","Subject")
```

Bloom/lost in mice

Identify taxa that are more or less abundant in mouse samples compared to respective donor samples.

```
#Calculate relative abundance of genera
genus.abund = genus/rowSums(genus)*100
#Remove S1 samples as comparison is S2 to M
genus.abund = genus.abund[!meta$Type == "S1",]
#Remove genera at <1% relative abundance across all samples
genus.abund.cutoff = genus.abund[, apply(genus.abund, MARGIN=2, function(x) any(x > 1))]
```

```
#Melt data from short/wide to long/skinny for all genera (columns)
genus.meta = merge(meta, genus.abund.cutoff, by="row.names")
    genus.meta = genus.meta[,!colnames(genus.meta) %in% c("Row.names", "chao", "shannon")]

#Create variable for samples in desired order
genus.meta$Type.ord = ifelse(genus.meta$Type == "S2", "1_S2",
                  ifelse(genus.meta$Type == "F", "2_F", genus.meta$Type))
genus.meta$Type.ord = as.factor(genus.meta$Type.ord)

#Melt metadata from short/wide to long/skinny
genus.melt = melt(genus.meta, id.vars=c("Type","HM","Subject","Sample", "Type.ord"))
```

Donor vs. other

For Fig. 3E

To obtain lists of beta-diversity values of mouse samples compared to samples from their matched subject donor (DONOR) or compared to any other subject in the dataset (OTHER).

Rename subjects to numbers so that distance may be calculated

```
meta$Subject.num = ifelse(meta$Subject == "Sub.1",1,
                    ifelse(meta$Subject == "Sub.2",2,
                     ifelse(meta$Subject == "Sub.3",3,
                      ifelse(meta$Subject == "Sub.4",4,
                       ifelse(meta$Subject == "Sub.5",5,
                         ifelse(meta$Subject == "Sub.6",6,
                             ifelse(meta$Subject == "Sub.8",8,NA)))))))
```

Pull out DONOR and OTHER beta-values

```
donor.other(meta$Subject.num, meta$Type, c("BC","J","uwUF","wUF"))
```

Read in results

```
donor.other = read.csv("output_donor.other/donor.other.csv", header=FALSE, row.names=1)
  colnames(donor.other) = c("distance","comparison","metric")
```

Statistics

Beta-diversity

Mouse fecals

Do the mouse fecal samples differ within each individual?

```
adonis(BC.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Sample, data=me
```

```
##
## Call:
## adonis(formula = BC.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F", ]
##
```

```
## Blocks:  strata 
## Permutation: free
## Number of permutations: 1000
## 
## Terms added sequentially (first to last)
## 
##           Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)  
## Sample     2    0.0664 0.033223 0.14022 0.01318 0.0979 .
## Residuals 21    4.9756 0.236932         0.98682         
## Total     23    5.0420                  1.00000         
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(J.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Sample, data=meta
```

```
## 
## Call:
## adonis(formula = J.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F", ]
## 
## Blocks:  strata 
## Permutation: free
## Number of permutations: 1000
## 
## Terms added sequentially (first to last)
## 
##           Df SumsOfSqs MeanSqs F.Model      R2  Pr(>F)  
## Sample     2    0.1391 0.06953 0.21195 0.01979 0.05195 .
## Residuals 21    6.8892 0.32806         0.98021          
## Total     23    7.0283                 1.00000          
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(wUF.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Sample, data=me
```

```
## 
## Call:
## adonis(formula = wUF.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F",
## 
## Blocks:  strata 
## Permutation: free
## Number of permutations: 1000
## 
## Terms added sequentially (first to last)
## 
##           Df SumsOfSqs   MeanSqs F.Model      R2  Pr(>F)  
## Sample     2   0.01559 0.0077951 0.36748 0.03381 0.02398 *
## Residuals 21   0.44546 0.0212125         0.96619          
## Total     23   0.46105                   1.00000          
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(uwUF.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Sample, data=m
```

```
## 
## Call:
## adonis(formula = uwUF.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F"
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2 Pr(>F)
## Sample     2    0.0594 0.02969 0.15952 0.01497 0.1588
## Residuals 21    3.9085 0.18612         0.98503
## Total     23    3.9679                 1.00000
```

Do mice innoculated from different donors differ from one another?

```
adonis(BC.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Subject, data=me
```

```
##
## Call:
## adonis(formula = BC.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F", ]
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Subject    7    4.6432 0.66332  26.614 0.92091 0.000999 ***
## Residuals 16    0.3988 0.02492         0.07909
## Total     23    5.0420                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(J.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Subject, data=met
```

```
##
## Call:
## adonis(formula = J.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F", ]]
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model     R2   Pr(>F)
## Subject    7    6.0893 0.86990  14.823 0.8664 0.000999 ***
## Residuals 16    0.9390 0.05869         0.1336
## Total     23    7.0283                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
adonis(wUF.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Subject, data=m
```

```
##
## Call:
## adonis(formula = wUF.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F",
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs  MeanSqs F.Model      R2   Pr(>F)
## Subject    7   0.39027 0.055753  12.603 0.84648 0.000999 ***
## Residuals 16   0.07078 0.004424         0.15352
## Total     23   0.46105                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(uwUF.dist[row.names(meta[meta$Type == "F",]),row.names(meta[meta$Type == "F",])] ~ Subject, data
```

```
##
## Call:
## adonis(formula = uwUF.dist[row.names(meta[meta$Type == "F", ]),      row.names(meta[meta$Type == "F"
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Subject    7    3.5556 0.50794  19.711 0.89609 0.000999 ***
## Residuals 16    0.4123 0.02577         0.10391
## Total     23    3.9679                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mouse vs human

Do human- (S1, S2) and mouse-derived (F) samples differ overall, regardless of subject?

```r
adonis(BC.dist ~ HM, data=meta, permutations=1000)
```

```
##
## Call:
## adonis(formula = BC.dist ~ HM, data = meta, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## HM         1    2.5604 2.56045  10.828 0.22176 0.000999 ***
## Residuals 38    8.9854 0.23646         0.77824
## Total     39   11.5458                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(J.dist ~ HM, data=meta, permutations=1000)
```

```
##
## Call:
```

```
## adonis(formula = J.dist ~ HM, data = meta, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## HM         1    2.2059 2.20586  6.9371 0.15437 0.000999 ***
## Residuals 38   12.0833 0.31798         0.84563
## Total     39   14.2891                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(wUF.dist ~ HM, data=meta, permutations=1000)
```

```
##
## Call:
## adonis(formula = wUF.dist ~ HM, data = meta, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## HM         1   0.63435 0.63435  20.065 0.34556 0.000999 ***
## Residuals 38   1.20136 0.03161         0.65444
## Total     39   1.83571                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(uwUF.dist ~ HM, data=meta, permutations=1000)
```

```
##
## Call:
## adonis(formula = uwUF.dist ~ HM, data = meta, permutations = 1000)
##
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## HM         1    1.3986 1.39857  7.8241 0.17074 0.000999 ***
## Residuals 38    6.7926 0.17875         0.82926
## Total     39    8.1911                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Full model

Do samples (S1, S2, F) differ within subjects?

```r
adonis(BC.dist ~ Type, data=meta, permutations=1000, strata=meta$Subject)
```

```
##
## Call:
## adonis(formula = BC.dist ~ Type, data = meta, permutations = 1000,      strata = meta$Subject)
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model     R2   Pr(>F)
## Type       2    2.5817 1.29083   5.328 0.2236 0.000999 ***
## Residuals 37    8.9642 0.24227         0.7764
## Total     39   11.5458                 1.0000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(J.dist ~ Type, data=meta, permutations=1000, strata=meta$Subject)
```

```
##
## Call:
## adonis(formula = J.dist ~ Type, data = meta, permutations = 1000,      strata = meta$Subject)
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Type       2    2.2473 1.12363  3.4525 0.15727 0.000999 ***
## Residuals 37   12.0419 0.32546         0.84273
## Total     39   14.2891                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(wUF.dist ~ Type, data=meta, permutations=1000, strata=meta$Subject)
```

```
##
## Call:
## adonis(formula = wUF.dist ~ Type, data = meta, permutations = 1000,      strata = meta$Subject)
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Type       2   0.63855 0.31928  9.8678 0.34785 0.000999 ***
## Residuals 37   1.19716 0.03236         0.65215
## Total     39   1.83571                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
adonis(wUF.dist ~ Type, data=meta, permutations=1000, strata=meta$Subject)
```

```
##
## Call:
## adonis(formula = wUF.dist ~ Type, data = meta, permutations = 1000,      strata = meta$Subject)
##
## Blocks:  strata
## Permutation: free
## Number of permutations: 1000
##
## Terms added sequentially (first to last)
##
##           Df SumsOfSqs MeanSqs F.Model      R2   Pr(>F)
## Type       2   0.63855 0.31928  9.8678 0.34785 0.000999 ***
## Residuals 37   1.19716 0.03236         0.65215
## Total     39   1.83571                 1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pairwise between all sample types

```r
pairwise.adonis.all(OTU.norm, meta, meta$Type, "bray", NJ.tree, stratify=TRUE, meta$Subject, "bonferron
```

```
## 'nperm' >= set of all permutations: complete enumeration.
```

```
## Set of permutations < 'minperm'. Generating entire set.
```

```
##      pairs    F.Model          R2   p.value p.adjusted
## 1  F vs S1 6.95051533 0.188103339 0.0010000  0.0030000
## 2  F vs S2 6.82856775 0.185414969 0.0010000  0.0030000
## 3 S1 vs S2 0.07572307 0.005379693 0.0078125  0.0234375
```

```r
pairwise.adonis.all(OTU.norm, meta, meta$Type, "jaccard", NJ.tree, stratify=TRUE, meta$Subject, "bonfer
```

```
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
```

```
##      pairs    F.Model          R2   p.value p.adjusted
## 1  F vs S1 6.95051533 0.188103339 0.0010000  0.0030000
## 2  F vs S2 6.82856775 0.185414969 0.0010000  0.0030000
## 3 S1 vs S2 0.07572307 0.005379693 0.0078125  0.0234375
```

```r
pairwise.adonis.all(OTU.norm, meta, meta$Type, "wunifrac", NJ.tree, stratify=TRUE, meta$Subject, "bonfe
```

```
## Warning in UniFrac(phy.object, weighted = TRUE, normalized = TRUE):
## Randomly assigning root as -- Otu3033 -- in the phylogenetic tree in the
## data you provided.
```

```
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
```

```
##      pairs     F.Model         R2   p.value p.adjusted
## 1  F vs S1 14.56814632 0.32687351 0.0010000  0.0030000
## 2  F vs S2 14.15651770 0.32059860 0.0010000  0.0030000
## 3 S1 vs S2  0.07894359 0.00560721 0.0234375  0.0703125
```

```r
pairwise.adonis.all(OTU.norm, meta, meta$Type, "uwunifrac", NJ.tree, stratify=TRUE, meta$Subject, "bonf
```

```
## Warning in UniFrac(phy.object, weighted = FALSE, normalized = TRUE):
```

```
## Randomly assigning root as -- Otu0439 -- in the phylogenetic tree in the
## data you provided.
```

```
## 'nperm' >= set of all permutations: complete enumeration.
## Set of permutations < 'minperm'. Generating entire set.
```

```
##      pairs   F.Model         R2   p.value p.adjusted
## 1  F vs S1 4.7238349 0.13604013 0.0010000  0.0030000
## 2  F vs S2 5.0679927 0.14451904 0.0010000  0.0030000
## 3 S1 vs S2 0.1713675 0.01209252 0.2578125  0.7734375
```

Donor vs. other

Is a mouse more like it's human donor than other humans in the dataset?

```
kruskal.test(donor.other[donor.other$metric == "BC",]$distance ~ donor.other[donor.other$metric == "BC"
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  donor.other[donor.other$metric == "BC", ]$distance by donor.other[donor.other$metric == "BC",
## Kruskal-Wallis chi-squared = 38.71, df = 1, p-value = 4.917e-10
```

```
kruskal.test(donor.other[donor.other$metric == "J",]$distance ~ donor.other[donor.other$metric == "J",]
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  donor.other[donor.other$metric == "J", ]$distance by donor.other[donor.other$metric == "J", ]
## Kruskal-Wallis chi-squared = 38.71, df = 1, p-value = 4.917e-10
```

```
kruskal.test(donor.other[donor.other$metric == "wUF",]$distance ~ donor.other[donor.other$metric == "wU
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  donor.other[donor.other$metric == "wUF", ]$distance by donor.other[donor.other$metric == "wUF
## Kruskal-Wallis chi-squared = 7.5078, df = 1, p-value = 0.006143
```

```
kruskal.test(donor.other[donor.other$metric == "uwUF",]$distance ~ donor.other[donor.other$metric == "u
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  donor.other[donor.other$metric == "uwUF", ]$distance by donor.other[donor.other$metric == "uw
## Kruskal-Wallis chi-squared = 96.546, df = 1, p-value < 2.2e-16
```

Alpha-diversity

Mouse fecals

Do the mouse fecal samples differ within each individual?

```
summary(aov(meta[meta$Type == "F",]$shannon ~ Sample + Error(Subject/Sample), data=meta[meta$Type == "F
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  7  2.494  0.3564
##
## Error: Subject:Sample
##           Df  Sum Sq Mean Sq F value Pr(>F)
## Sample     2 0.03443 0.01721   4.782 0.0261 *
## Residuals 14 0.05039 0.00360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(meta[meta$Type == "F",]$chao ~ Sample + Error(Subject/Sample), data=meta[meta$Type == "F",]
```

```
##
## Error: Subject
##           Df Sum Sq Mean Sq F value Pr(>F)
## Residuals  7  20597    2942
##
## Error: Subject:Sample
##           Df Sum Sq Mean Sq F value Pr(>F)
## Sample     2   1560   779.8    2.13  0.156
## Residuals 14   5124   366.0
```

Do mice innoculated from different donors differ?

```
summary(aov(meta[meta$Type == "F",]$shannon ~ Subject, data=meta[meta$Type == "F",]))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Subject      7 2.4945  0.3564   67.22 1.13e-10 ***
## Residuals   16 0.0848  0.0053
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(meta[meta$Type == "F",]$chao ~ Subject, data=meta[meta$Type == "F",]))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## Subject      7  20597  2942.4   7.043 0.000625 ***
## Residuals   16   6684   417.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Mouse vs human

Do human- (S1, S2) and mouse-derived (F) samples differ overall, regardless of subject?

```
summary(aov(shannon ~ HM, data=meta))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## HM           1  5.450    5.45   49.46 2.22e-08 ***
## Residuals   38  4.188    0.11
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(aov(chao ~ HM, data=meta))
```

```
##             Df Sum Sq Mean Sq F value   Pr(>F)
## HM           1 132306  132306   57.25 4.27e-09 ***
## Residuals   38  87824    2311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Full model

Do samples (S1, S2, F) differ within subjects?

```r
shan.mod = lme(shannon ~ Type, random = ~1|Subject, data=meta)
anova(shan.mod)
```

```
##             numDF denDF  F-value p-value
## (Intercept)     1    30 803.2115  <.0001
## Type            2    30 108.0438  <.0001
```

```r
#Pairwise
summary(glht(shan.mod,linfct=mcp(Type="Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme.formula(fixed = shannon ~ Type, data = meta, random = ~1 |
##     Subject)
##
## Linear Hypotheses:
##              Estimate Std. Error z value Pr(>|z|)
## S1 - F == 0   0.71209    0.06500   10.96   <1e-04 ***
## S2 - F == 0   0.79488    0.06500   12.23   <1e-04 ***
## S2 - S1 == 0  0.08278    0.07961    1.04    0.548
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

```r
chao.mod = lme(chao ~ Type, random = ~1|Subject, data=meta)
anova(chao.mod)
```

```
##             numDF denDF  F-value p-value
## (Intercept)     1    30 186.4522  <.0001
## Type            2    30  69.0471  <.0001
```

```r
#Pairwise
summary(glht(chao.mod,linfct=mcp(Type="Tukey")))
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lme.formula(fixed = chao ~ Type, data = meta, random = ~1 | Subject)
##
## Linear Hypotheses:
##              Estimate Std. Error z value Pr(>|z|)
## S1 - F == 0    104.85      12.76   8.220   <1e-04 ***
## S2 - F == 0    129.94      12.76  10.186   <1e-04 ***
## S2 - S1 == 0    25.09      15.62   1.606     0.24
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

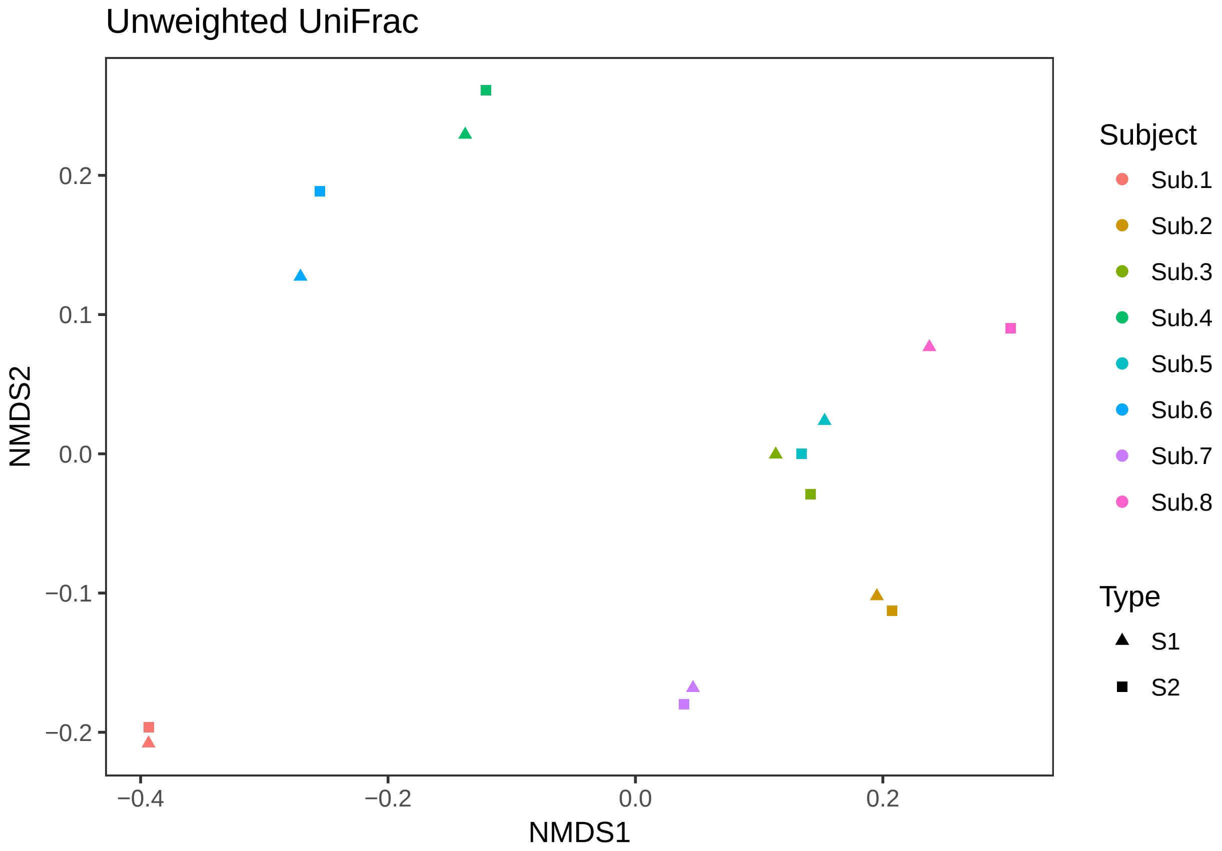
Figures

Figure 2A

Non-metric multidimensional scaling plots of the unweighted UniFrac metric between human-derived samples, colored by subject. Colored ellipses are smallest area for human samples from each subject drawn in Illustrator.

```
nMDS_uwUF_H=plot_ordination(physeq.tree.H, uwUF.nmds.H, type="sites", color="Subject", shape="Type") +
  theme_bw() +
  scale_shape_manual(values=c(17,15,3)) +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme
  ggtitle("Unweighted UniFrac")

#View
nMDS_uwUF_H
```



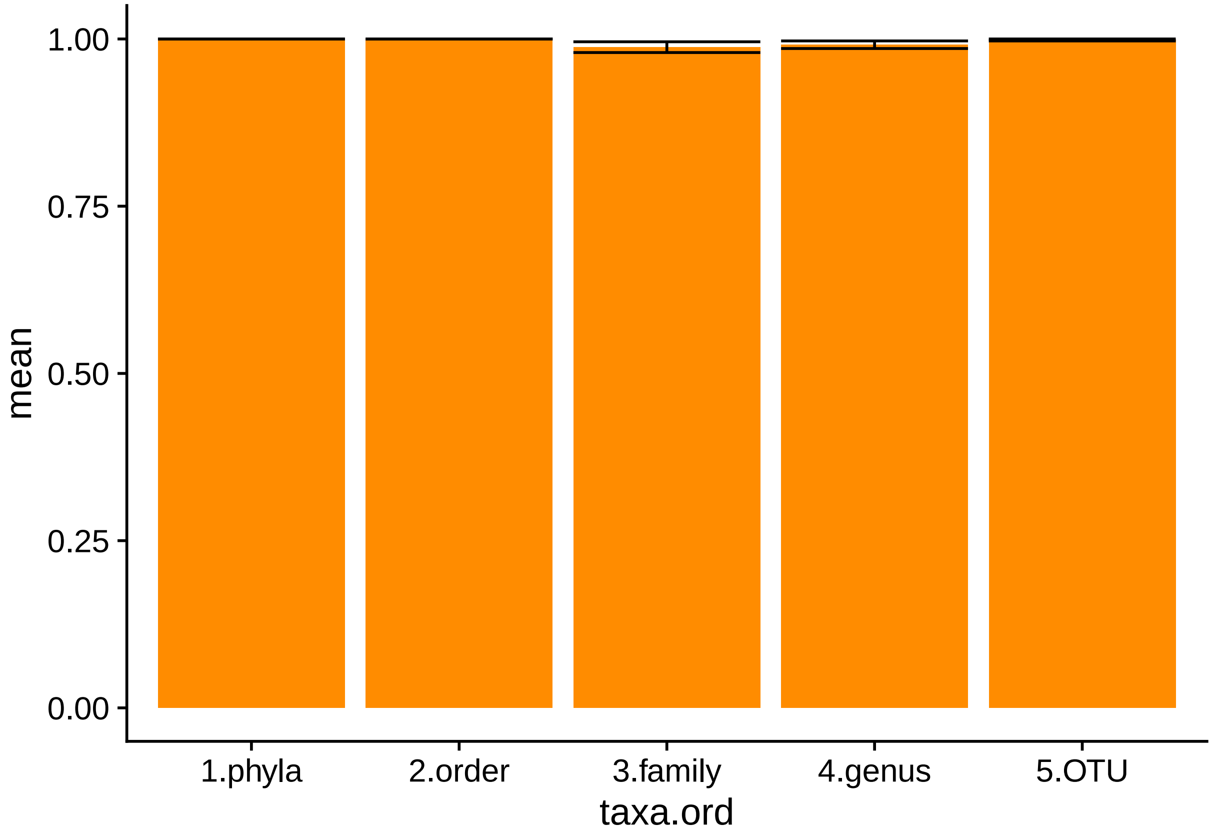
```
#Save
save_plot("Figures/Fig2A.eps", nMDS_uwUF_H)
```

Figure 2B

Taxa recovery in S2 compared to S1. Calculated with non-rarified data. Identified taxa were at least 0.1% relative abundance in at least one of the two samples in each comparison.

```
S12.recovery=ggplot(taxa.recover[taxa.recover$comp == "S12",], aes(taxa.ord, mean)) +
  geom_bar(fill = "darkorange", position = "dodge", stat="identity") +
  geom_errorbar(aes(ymax=taxa.recover[taxa.recover$comp == "S12",]$mean +
  taxa.recover[taxa.recover$comp == "S12",]$se, ymin=taxa.recover[taxa.recover$comp == "S12",]$mean - ta
  theme(legend.position="none")

#View
S12.recovery
```



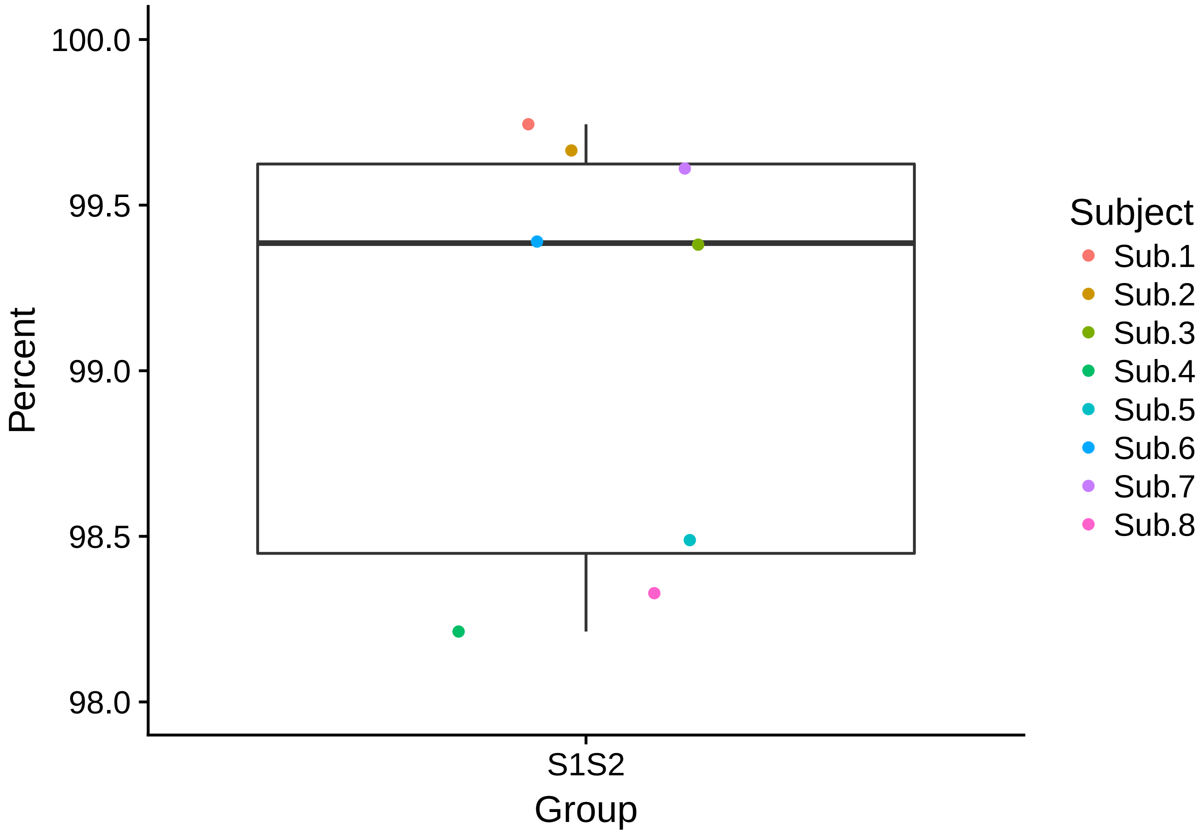
```
#Save
save_plot("Figures/Fig2B.eps", S12.recovery, base_height=3, base_width=4)
```

Figure 2C

Percent relative abundance of S1 community captured in oral gavage preparation (S2). Where shown, bars represent mean ± standard error.

```
perc.retain.S12 = ggplot(venn.perc[venn.perc$Group %in% c("S1S2"),], aes(x=Group, y=Percent)) +
  geom_boxplot(aes(group=Group)) +
  geom_jitter(width=0.15, height=0, aes(color=Subject)) +
  ylim(98,100)
```

```
#View
perc.retain.S12
```



```
#Save
postscript("Figures/Fig2C.ps", width = 3, height = 3, horizontal = FALSE, colormodel = "rgb", fonts=c("
perc.retain.S12
dev.off()
```

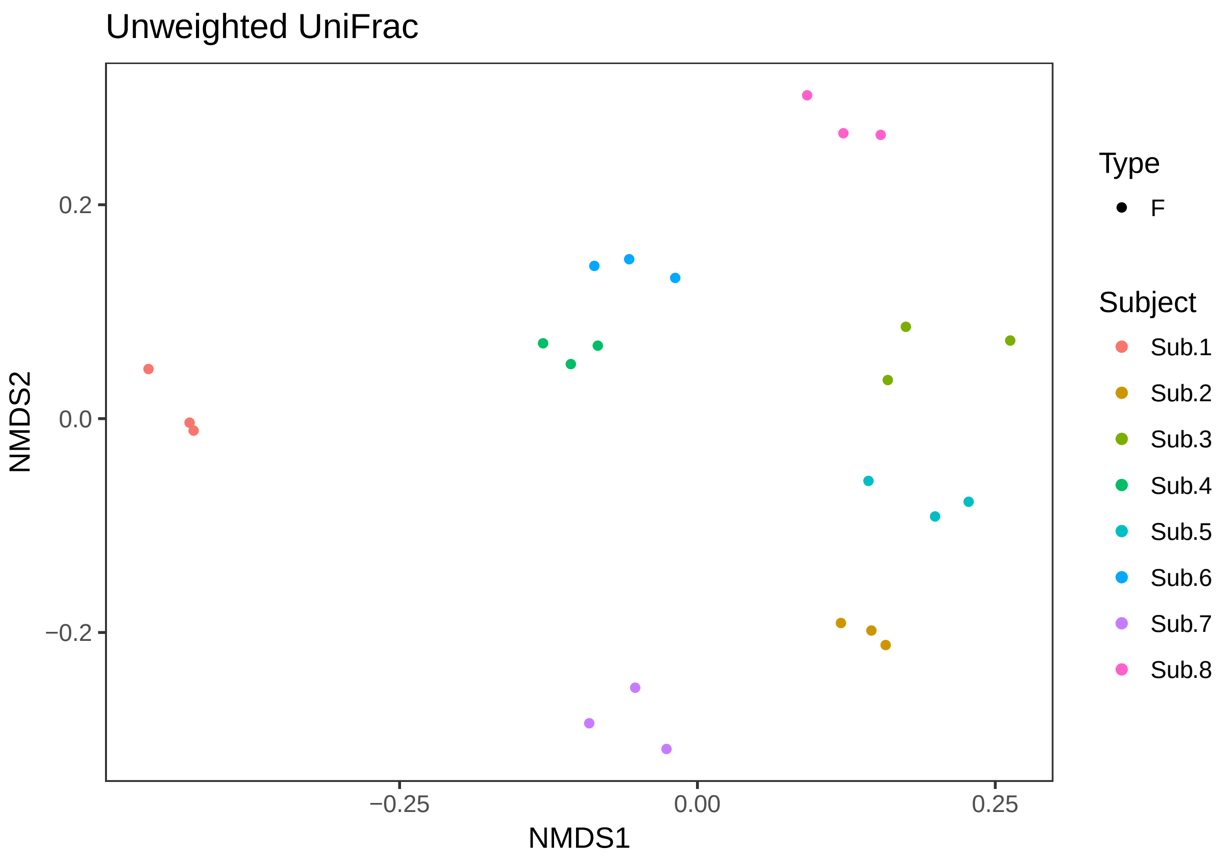
```
## pdf
##   2
```

Figure 3A

Non-metric multidimensional scaling plots of the unweighted UniFrac metric between mouse fecal samples, colored by donor. Colored ellipses are smallest area for mouse-derived samples from each subject drawn in Illustrator.

```
nMDS_uwUF_M=plot_ordination(physeq.tree.M, uwUF.nmds.M, type="sites", color="Subject", shape="Type") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = eleme
  ggtitle("Unweighted UniFrac")

#View
nMDS_uwUF_M
```



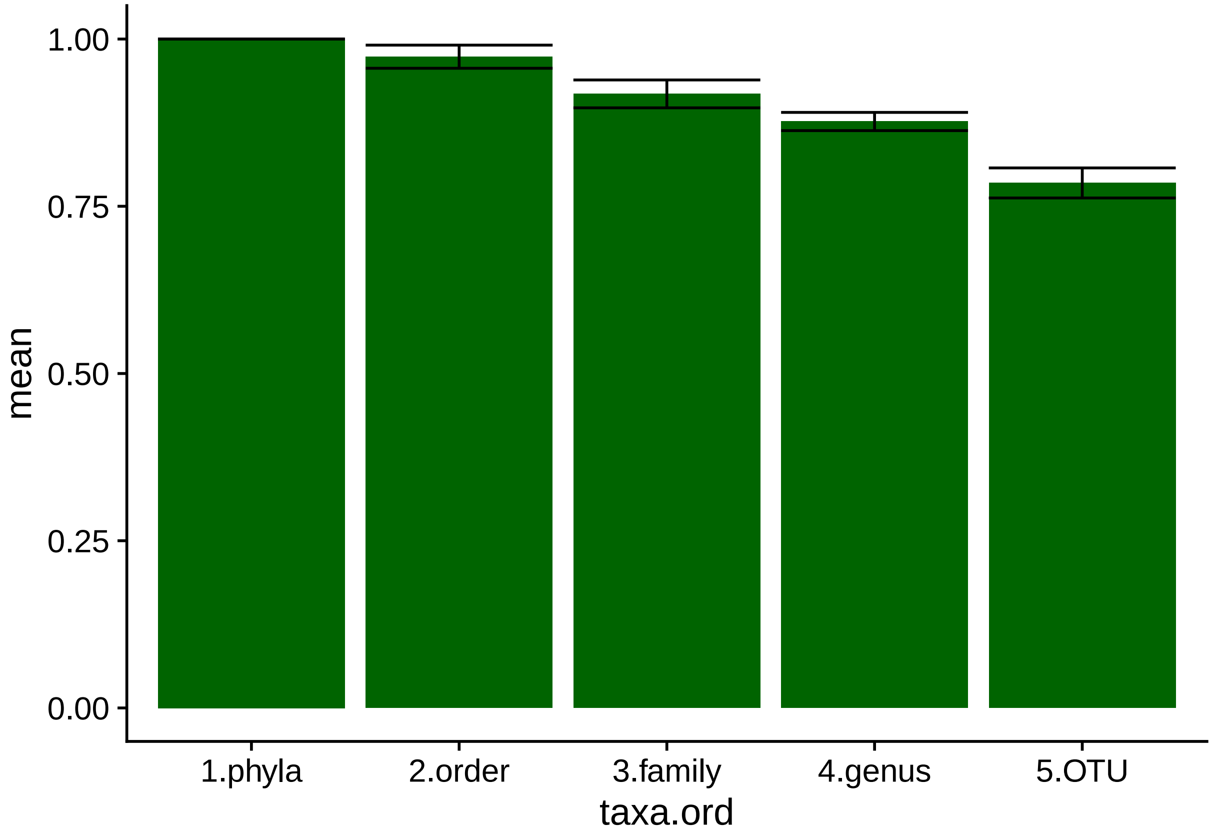
```
#Save
save_plot("Figures/Fig3A.eps", nMDS_uwUF_M)
```

Figure 3B

Taxa recovery in mouse fecal samples compared to oral inoculum (S2). Calculated with non-rarified data. Identified taxa were at least 0.1% relative abundance in at least one of the four samples in each comparison (S2 and 3 mouse fecal samples).

```
S2M.recovery=ggplot(taxa.recover[taxa.recover$comp == "S2M",], aes(taxa.ord, mean)) +
  geom_bar(fill = "darkgreen", position = "dodge", stat="identity")+
  geom_errorbar(aes(ymax=taxa.recover[taxa.recover$comp == "S2M",]$mean +
  taxa.recover[taxa.recover$comp == "S2M",]$se, ymin=taxa.recover[taxa.recover$comp == "S2M",]$mean - ta
    theme(legend.position="none")

#View
S2M.recovery
```



```
#Save
save_plot("Figures/Fig3B.eps", S2M.recovery, base_height=3, base_width=4)
```

Figure 3C

OTU Venn diagrams. Yellow circles represent OTUs in the oral inoculum (A=S2) and blue circles represent OTUs recovered in the mouse (B=M).

Create venns scaled to total OTU counts from `venn.abund.all` function. Cutoff of 0.1% in at least one sample in a given venn.

```
Sub.1_S2M = venneuler(c('A'=13,'A&B'=61,'B'=3))
Sub.2_S2M = venneuler(c('A'=23,'A&B'=90,'B'=0))
Sub.3_S2M = venneuler(c('A'=27,'A&B'=73,'B'=8))
Sub.4_S2M = venneuler(c('A'=22,'A&B'=83,'B'=5))
Sub.5_S2M = venneuler(c('A'=23,'A&B'=79,'B'=14))
Sub.6_S2M = venneuler(c('A'=20,'A&B'=74,'B'=1))
Sub.7_S2M = venneuler(c('A'=17,'A&B'=74,'B'=1))
Sub.8_S2M = venneuler(c('A'=44,'A&B'=89,'B'=12))
```

Save Venns in 1 figure

```
pdf("Figures/Fig.3C.pdf", width = 20, height = 8, colormodel = "rgb", fonts=c("serif"))

layout(matrix(c(1,2,3,4,5,6,7,8), 2, 4))
plot(Sub.1_S2M, col=c("yellow","blue"))
plot(Sub.2_S2M, col=c("yellow","blue"))
```

```r
plot(Sub.3_S2M, col=c("yellow","blue"))
plot(Sub.4_S2M, col=c("yellow","blue"))
plot(Sub.5_S2M, col=c("yellow","blue"))
plot(Sub.6_S2M, col=c("yellow","blue"))
plot(Sub.7_S2M, col=c("yellow","blue"))
plot(Sub.8_S2M, col=c("yellow","blue"))

dev.off()
```

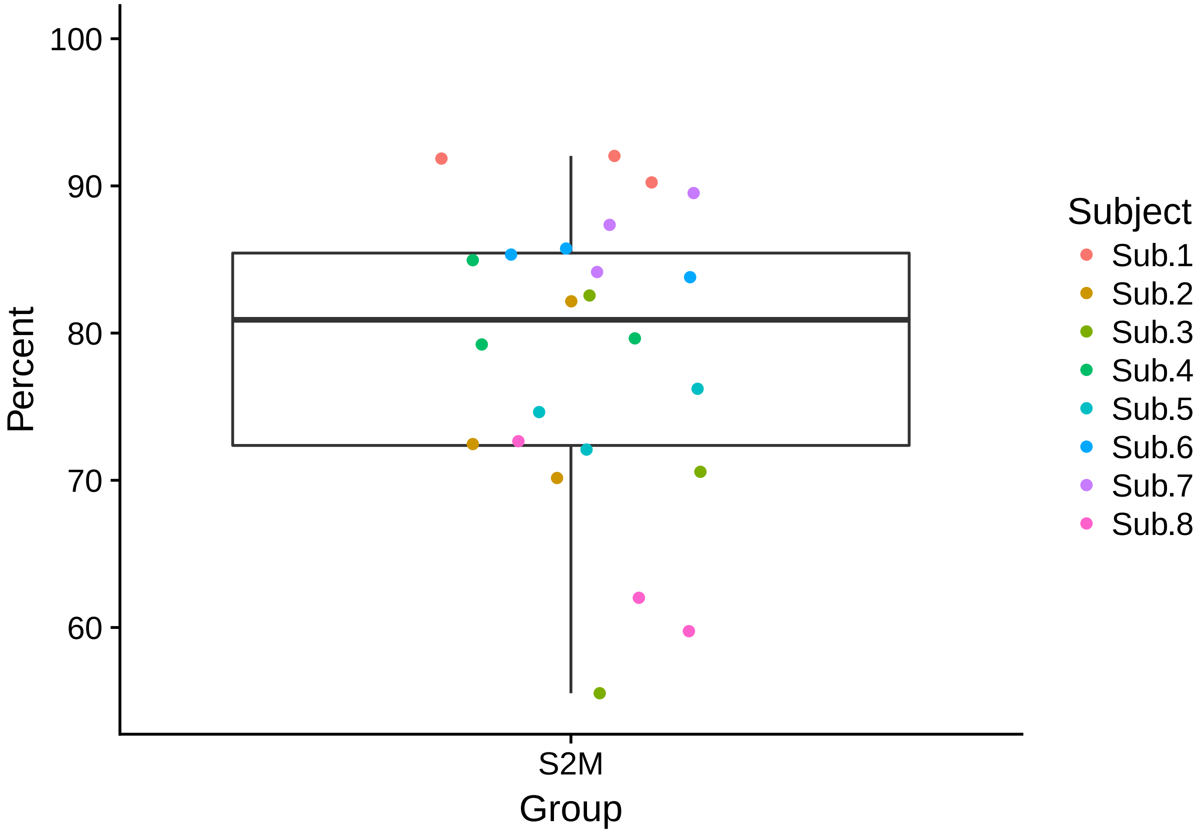
```
## pdf
##   2
```

Figure 3D

Percent relative abundance of the oral gavage community (S2) captured in mouse fecal samples.

```r
perc.retain.M = ggplot(venn.perc[venn.perc$Group %in% c("S2M"),], aes(x=Group, y=Percent)) +
  geom_boxplot(aes(group=Group)) +
  geom_jitter(width=0.15, height=0, aes(color=Subject)) +
  ylim(55,100)

#View
perc.retain.M
```



```r
#Save
postscript("Figures/Fig3D.ps", width = 3, height = 3, horizontal = FALSE, colormodel = "rgb", fonts=c("
perc.retain.M
dev.off()
```

```
## pdf
##   2
```

Figure 3E

Bray-Curtis (BC), Jaccard (J), weighted UniFrac (wUF), and unweighted UniFrac (uwUF) beta-diversity measures of mouse samples compared to samples from their matched subject donor (DONOR) or compared to any other subject in the dataset (OTHER).

Pull out distances from donor.other for each comparison and metric

```
match.BC = as.vector(donor.other[donor.other$comparison == "match" & donor.other$metric == "BC",]$distan
unmatch.BC = as.vector(donor.other[donor.other$comparison == "unmatch" & donor.other$metric == "BC",]$di

match.J = as.vector(donor.other[donor.other$comparison == "match" & donor.other$metric == "J",]$distanc
unmatch.J = as.vector(donor.other[donor.other$comparison == "unmatch" & donor.other$metric == "J",]$dis

match.wUF = as.vector(donor.other[donor.other$comparison == "match" & donor.other$metric == "wUF",]$dis
unmatch.wUF = as.vector(donor.other[donor.other$comparison == "unmatch" & donor.other$metric == "wUF",]

match.uwUF = as.vector(donor.other[donor.other$comparison == "match" & donor.other$metric == "uwUF",]$di
unmatch.uwUF = as.vector(donor.other[donor.other$comparison == "unmatch" & donor.other$metric == "uwUF"
```

Calculate mean and standard errors

```
dist.summary = data.frame(
  mean = c(mean(match.BC), mean(unmatch.BC),
           mean(match.J), mean(unmatch.J),
           mean(match.wUF), mean(unmatch.wUF),
           mean(match.uwUF), mean(unmatch.uwUF)),
  se = c(sd(match.BC)/sqrt(length(match.BC)),
         sd(unmatch.BC)/sqrt(length(unmatch.BC)),
         sd(match.J)/sqrt(length(match.J)),
         sd(unmatch.J)/sqrt(length(unmatch.J)),
         sd(match.wUF)/sqrt(length(match.wUF)),
         sd(unmatch.wUF)/sqrt(length(unmatch.wUF)),
         sd(match.uwUF)/sqrt(length(match.uwUF)),
         sd(unmatch.uwUF)/sqrt(length(unmatch.uwUF))),
  comp = rep(c("match", "unmatch"), times=4),
  dist = c("BC","BC","J","J","wUF","wUF","uwUF","uwUF"),
  group = c("1mBC","2uBC","3mJ","4uJ","5mwUF","6uwUF","7muwUF","8uuwUF"))
```

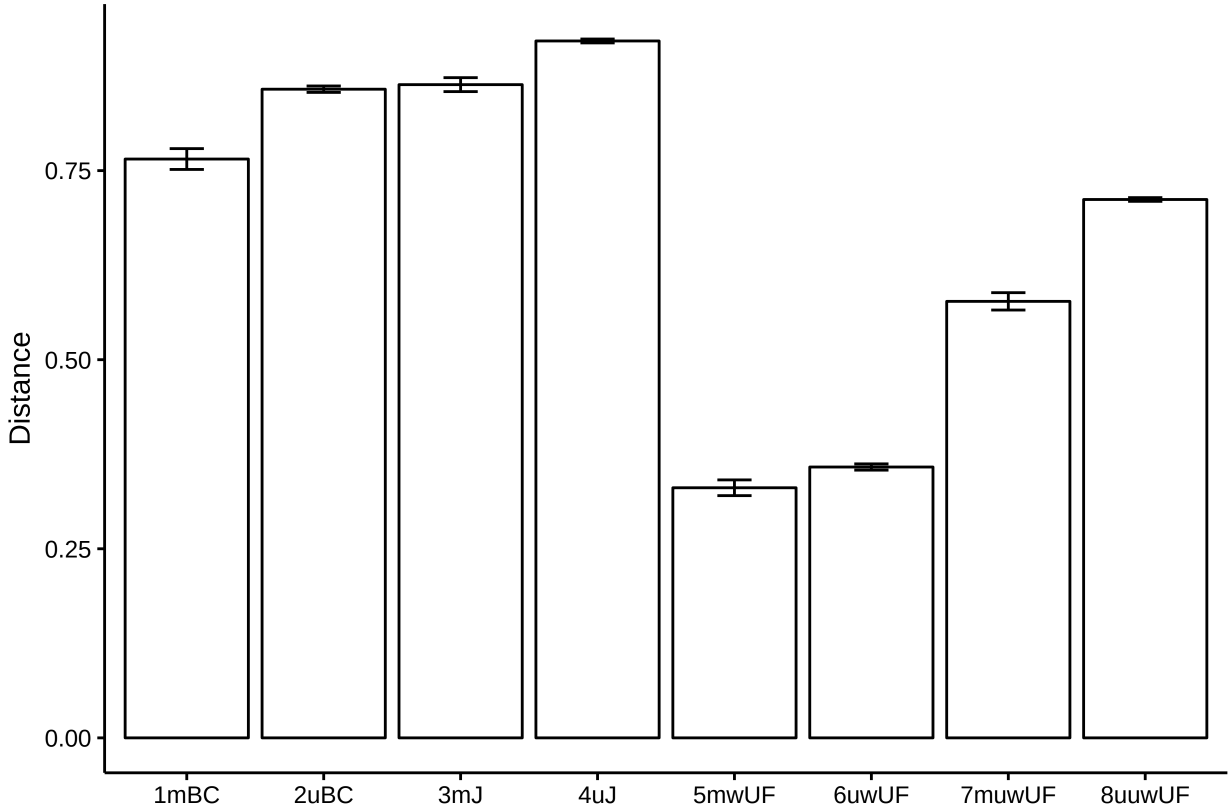
Plot means with se error bars

```
#Set limits of error bars to be +/- standard error
limits = aes(ymax = dist.summary$mean + dist.summary$se, ymin = dist.summary$mean - dist.summary$se)

#Create base plot
p1 = ggplot(data = dist.summary, aes(x = group, y = mean, color=comp))

#Add data bars
bar_donor_other=
  p1 + geom_bar(fill='white', stat = "identity", position = position_dodge(0.9)) +
  geom_errorbar(limits, position = position_dodge(0.9),  width = 0.25) +
  labs(x = "", y = "Distance") +
  theme_bw() +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_
  scale_color_manual(values=c('black','black'))  +
    theme(legend.position="none")

#View
bar_donor_other
```



```
#Save
save_plot("Figures/Fig3E.eps", bar_donor_other, base_height=2.5, base_width=3.5)
```

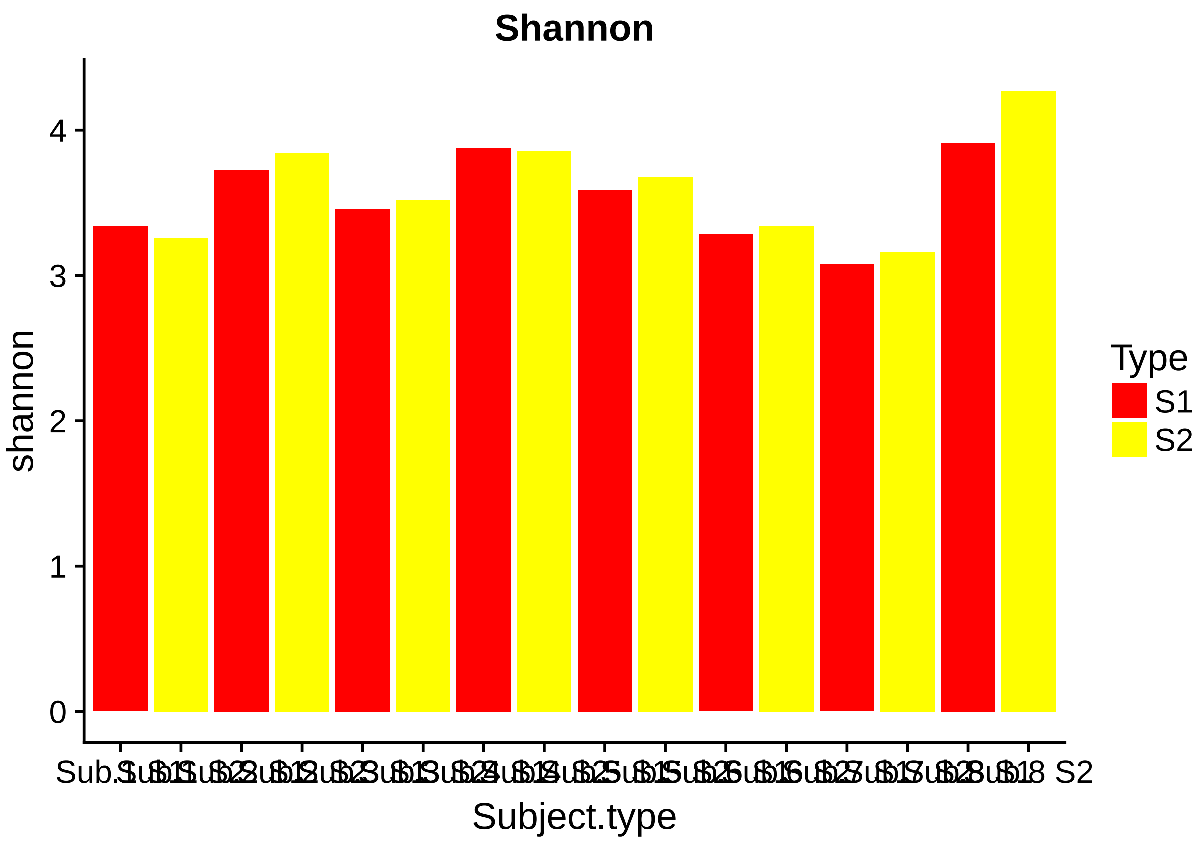
Figure S1

Shannon's diversity of microbial communities for the eight sequenced human samples. S1 (red) and S2 (yellow) represent randomly selected straws for each subject.

```
#Create variable of subject wih sample type
meta$Subject.type = paste(meta$Subject, meta$Type)

shannon.plot = ggplot(meta[meta$Type %in% c("S1","S2"),], aes(Subject.type, shannon)) +
  geom_bar(aes(fill = Type), position = "dodge", stat="identity") +
  ggtitle("Shannon") +
  scale_fill_manual(values=c("red","yellow"))

#View
shannon.plot
```



```
#Save
save_plot("Figures/FigS1.eps", shannon.plot, base_height=3, base_width=4)
```

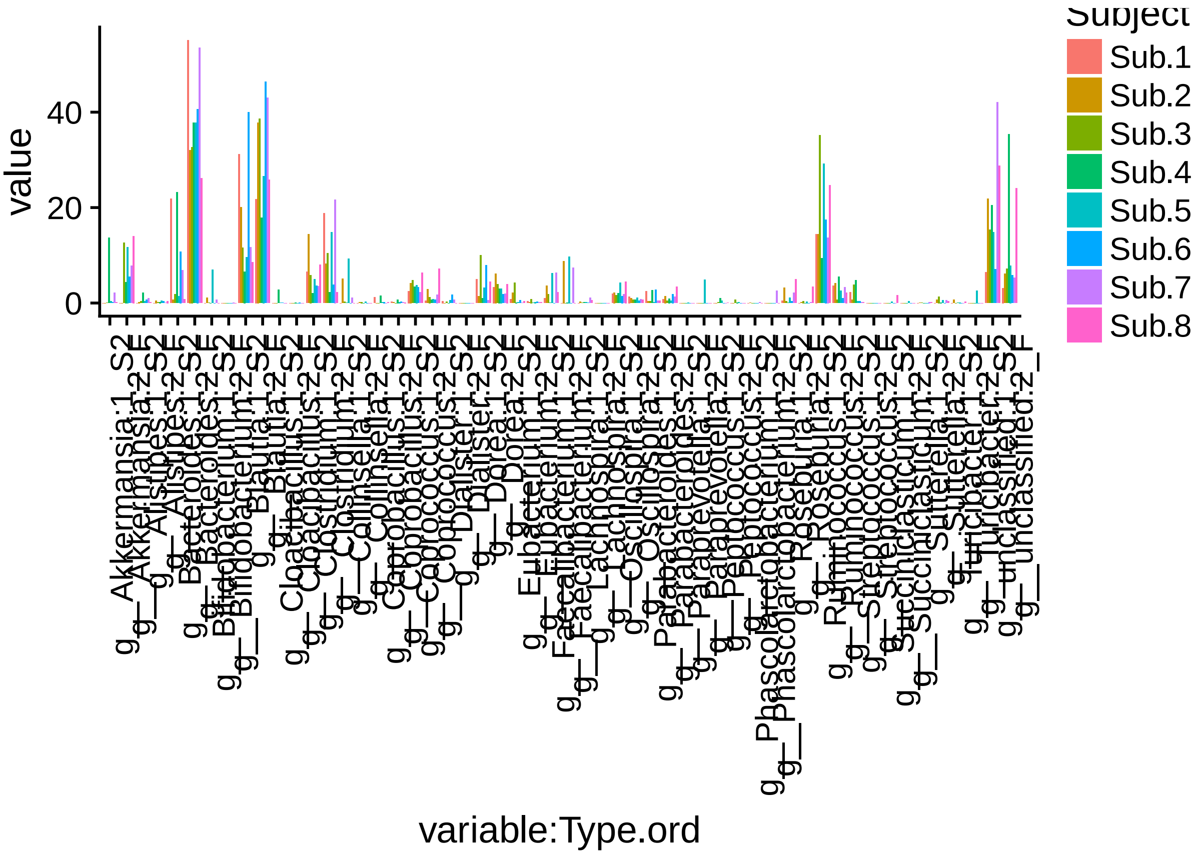
Figure S2

Genera that tended to be lost or bloomed in humanized mice. Relative abundances of genera at least 1% relative abundance in at least one straw2 (S2) or mouse (M) sample. Colored by subject and split into (A) more highly abundant and (B) lower abundance genera. Bars for mice are averages across the three mouse fecal samples for each subject.

First, explore all taxa

```
bloom.taxa.all=ggplot(genus.melt, aes(variable:Type.ord, value)) +
  geom_bar(aes(fill = Subject), position = "dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

bloom.taxa.all
```

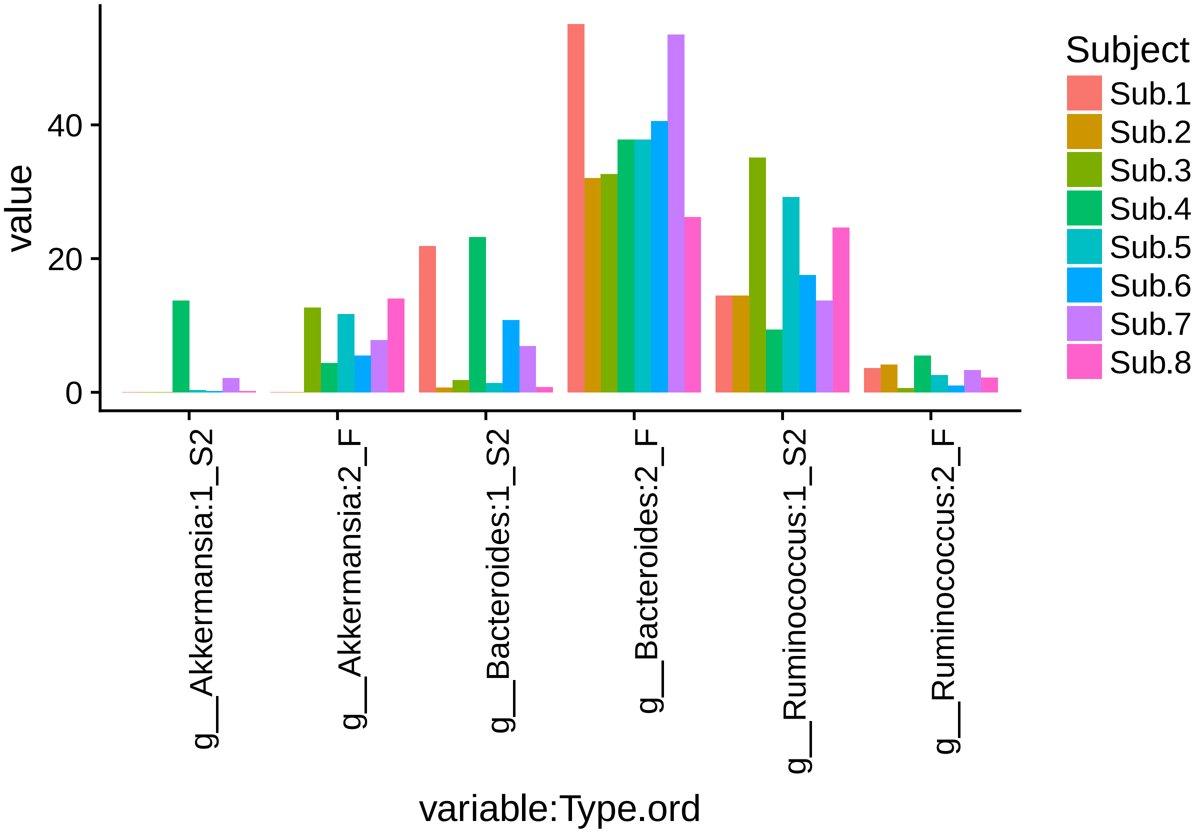


Select those with visual pattern of higher or lower in mice than respective donor.

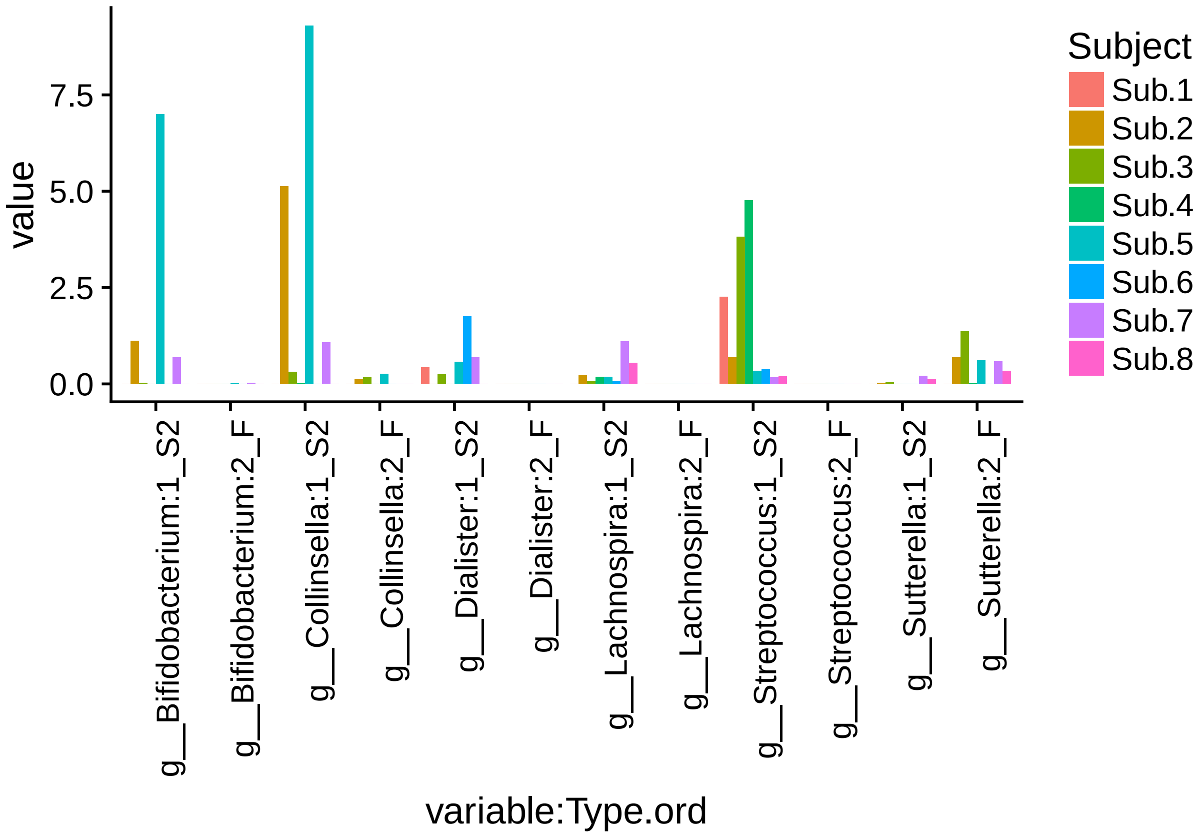
```
#Fig S2A - Higher abundance taxa
bloom.taxa.high=ggplot(genus.melt[genus.melt$variable %in% c("g__Akkermansia","g__Bacteroides","g__Rumi
  geom_bar(aes(fill = Subject), position = "dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

#Fig S2B - Lower abundance taxa
bloom.taxa.low=ggplot(genus.melt[genus.melt$variable %in% c("g__Bifidobacterium","g__Collinsella","g__Di
  geom_bar(aes(fill = Subject), position = "dodge", stat="identity") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

bloom.taxa.high
```



```
bloom.taxa.low
```



```
postscript("Figures/FigS2A.ps", width = 3, height = 4, horizontal = FALSE, colormodel = "rgb", fonts=c(
bloom.taxa.high
dev.off()
```

```
## pdf
##   2
```

```
postscript("Figures/FigS2B.ps", width = 6, height = 4, horizontal = FALSE, colormodel = "rgb", fonts=c(
bloom.taxa.low
dev.off()
```

```
## pdf
##   2
```

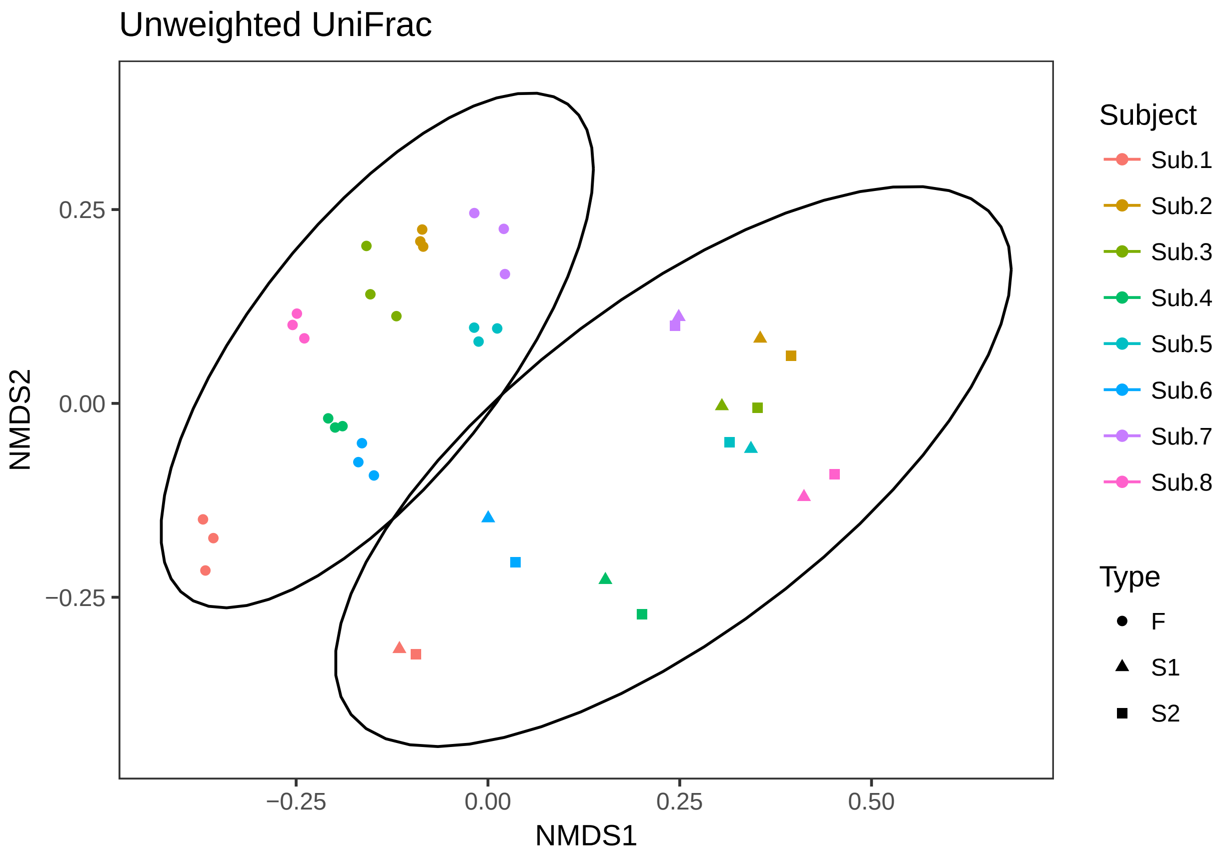
Figure S4

Unweighted UniFrac beta-diversity of FAST and mouse fecal microbiota. Non-metric multidimensional scaling plots of the unweighted UniFrac metric between all samples. Samples are colored by subject. Shapes indicate sample type. Black ellipses are standard error for human and mouse groups.

```
nMDS_uwUF_all=plot_ordination(physeq.tree, uwUF.nmds, type="sites", color="Subject", shape="Type") +
  theme_bw() +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.background = elem
  ggtitle("Unweighted UniFrac") +
  stat_ellipse(aes(group=HM), level=0.95)

#View
```

```
nMDS_uwUF_all
```



```
#Save
save_plot("Figures/FigS4.eps", nMDS_uwUF_all)
```

FIN