Core microbiome analysis

Alice Risely

15/01/2021

Background

Analysis for the manuscript:

Accounting for phylogeny and abundance allows for consistent comparison of core microbiome diversity indexes across species Risely, A., Gillingham, M.A.F. , Béchet, A., Brändel, S., Heni, A.C. 1, Heurich, M., Menke, S., Manser, M., Tschapka, M., Wasimuddin, & Sommer, S.

ABSTRACT

The filtering of gut microbial datasets to retain high prevalence taxa is often performed to identify a common core microbiome that may be important for host biological functions. However, prevalence thresholds used to identify a common core are highly variable and it remains unclear whether insights stemming from core microbiomes are comparable across studies. We hypothesized that if macroecological patterns in gut microbiome prevalence and abundance are similar across host species, then we would expect that increasing prevalence thresholds would yield similar changes to alpha diversity and beta dissimilarity across host species datasets. To test this, we analysed eight gut microbiome datasets collected from different host species to examine the extent to which gut microbial communities exhibit similar macroecological patterns in amplicon sequence variant (ASV) prevalence and abundance, and to test whether increasing prevalence thresholds to identify a common core generates universal or host-species specific effects on alpha and beta diversity scores at the sample level. We found that increasing prevalence thresholds generated remarkably similar trends in standardized alpha diversity and beta dissimilarity across the different host species datasets, and that these analogous responses were underpinned by similar macroecological patterns in prevalence and abundance across sampled populations. For both alpha and beta diversity, metrics that accounted for both abundance and phylogeny (Balance-weighted phylogenetic diversity and Weighted Unifrac, respectively) were insensitive to prevalence thresholds and therefore represented the common core microbiome without the need for filtering. In addition, we found that a sample size of approximately 150 sampled individuals was generally (but not always) sufficient to detect 90% of ASVs predicted to be present in the sampled population, whilst a sequencing depth of 10,000 was broadly sufficient for detecting the vast majority of ASVs per sample. Overall, we conclude that accounting for phylogeny abundance in diversity estimates allows for the consistent comparison of core microbiomes across studies, and that, generally, at least 150 individuals need to be sampled from a population to detect 90% of ASVs across the host population and therefore estimate prevalence distributions.

Analysis in R

Load packages

```
library(phyloseq)
library(ggplot2)
library(plyr)
library(metagMisc)
library(tidyr)
library(RColorBrewer)
library(ggpubr)
library(dplyr)
library(vegan)
```

Import data

This is a list of phyloseq object per species. These phyloseq objects are not normalised

```
phylo_list<-readRDS("phylo_list.RDS")

#View data

phylo_list
```

```
## $Human
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2526 taxa and 228 samples ]
## sample_data() Sample Data:       [ 228 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 2526 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2526 tips and 2525 internal nodes ]
##
## $Meerkat
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 4475 taxa and 137 samples ]
## sample_data() Sample Data:       [ 137 samples by 49 sample variables ]
## tax_table()   Taxonomy Table:    [ 4475 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 4475 tips and 4474 internal nodes ]
##
## $Deer
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 6442 taxa and 136 samples ]
## sample_data() Sample Data:       [ 136 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 6442 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 6442 tips and 6441 internal nodes ]
##
## $Bat
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2637 taxa and 161 samples ]
## sample_data() Sample Data:       [ 161 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 2637 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2637 tips and 2636 internal nodes ]
##
## $Spinyrat
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 3102 taxa and 196 samples ]
```

```
## sample_data() Sample Data:       [ 196 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 3102 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 3102 tips and 3101 internal nodes ]
##
## $Mouselemur
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 1213 taxa and 182 samples ]
## sample_data() Sample Data:       [ 182 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 1213 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 1213 tips and 1212 internal nodes ]
##
## $Flamingo
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2008 taxa and 356 samples ]
## sample_data() Sample Data:       [ 356 samples by 16 sample variables ]
## tax_table()   Taxonomy Table:    [ 2008 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2008 tips and 2007 internal nodes ]
##
## $Stint
## phyloseq-class experiment-level object
## otu_table()   OTU Table:         [ 2357 taxa and 97 samples ]
## sample_data() Sample Data:       [ 97 samples by 9 sample variables ]
## tax_table()   Taxonomy Table:    [ 2357 taxa by 7 taxonomic ranks ]
## phy_tree()    Phylogenetic Tree: [ 2357 tips and 2356 internal nodes ]
```

Figure 1a and 1b

Here we use vegan package to generate ASV accumulation curves per dataset.

```
SAClist<-list() #make an enpty list

uniq <- names(phylo_list) # make a list of species to subset sequentially

for (i in 1:length(uniq)){ #for species i

  data_1<-phylo_list[[i]] #subset the phyloseq object for species i
  data_1<-prune_taxa(taxa_sums(data_1)>0, data_1)  #remove any traces of taxa that are no longer presen
  data_1_matrix<-data.frame(t(data.frame(otu_table(data_1)))) #transpose the OTU table
  data_1_specaccum<-vegan::specaccum(data_1_matrix, method="random", permutations = 999) #apply specacc

  ## the output is in list form, so we need to make this into a dataframe

  sac_df<- data_1_specaccum$sites ##sites = samples
  sac_df<-data.frame(sac_df)
  names(sac_df)[1]<-"Site"
  sac_df$Richness <-  data_1_specaccum$richness #import ASV richness to dataframe
  sac_df$SD <-  data_1_specaccum$sd #import the standard deviation

  ## this next step estimates the TOTAL number of ASVs in the ASV pool.

  sac_total_estimated<-vegan::specpool(data_1_matrix) #estimates total ASV pool from our otu matrix gee
  sac_df$Total <- sac_total_estimated$boot ##add this to our dataframe
```

```
  sac_df$Species <- as.character(uniq[[i]]) #add species name, for when we combine dataframes for all sp
  SAClist[[i]]<-sac_df #add this dataframe as an element in the empty list and repeat for the next spec

}

names(SAClist)<-uniq #name elements of the list by species

sac_df_all<-do.call(rbind, SAClist) #rbind all our 8 dataframes together

head(sac_df_all) #final dataframe we use to generate the second figure
```

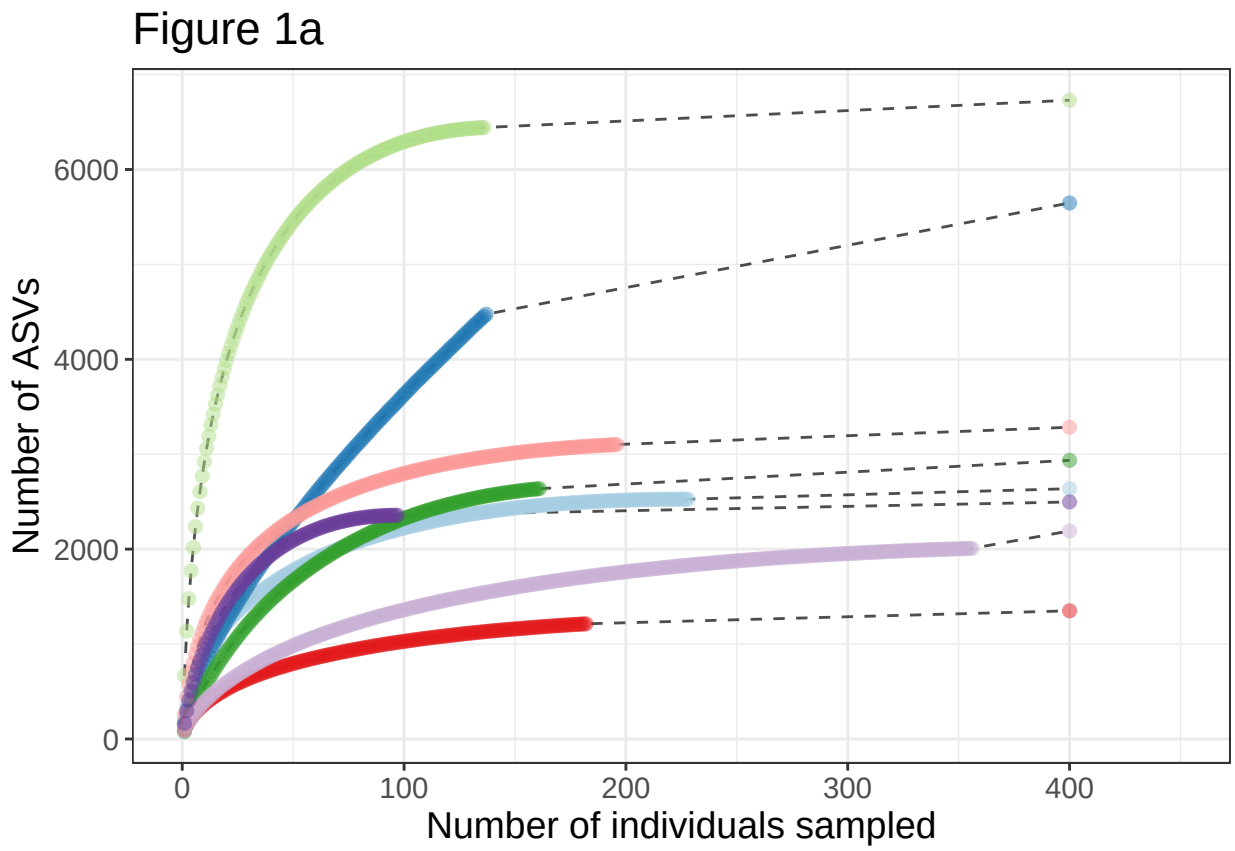
```
##         Site Richness       SD    Total Species
## Human.1    1 215.6697 63.55003 2636.761   Human
## Human.2    2 349.5816 65.97117 2636.761   Human
## Human.3    3 451.1742 66.75617 2636.761   Human
## Human.4    4 535.9479 67.47951 2636.761   Human
## Human.5    5 608.3824 67.35917 2636.761   Human
## Human.6    6 673.3914 66.05011 2636.761   Human
```

```
species_totals<-sac_df_all %>% distinct(Total, .keep_all = T) #subset just the eight distinct estimated

species_totals[,1]<-400 #here we put 400 just because we want a number that is larger than the largest
species_totals[,2]<-species_totals$Total
species_totals[,3]<-NA
species_totals[,4]<-NA

head(species_totals)
```

```
##              Site Richness SD Total    Species
## Human.1       400 2636.761 NA    NA      Human
## Meerkat.1     400 5647.042 NA    NA    Meerkat
## Deer.1        400 6730.058 NA    NA       Deer
## Bat.1         400 2935.775 NA    NA        Bat
## Spinyrat.1    400 3284.068 NA    NA   Spinyrat
## Mouselemur.1  400 1350.205 NA    NA Mouselemur
```

```
sac_df_fig<-rbind(sac_df_all, species_totals) #combine

sac_df_fig$Species<-factor(sac_df_fig$Species, levels = uniq)

## colour palette

palette<-brewer.pal(10,"Paired")
palette<-palette[c(-7,-8)]
#show_col(palette)

#Fig1a

ggplot(sac_df_fig, aes(x = Site, y = Richness, group = Species))+
  geom_line(alpha=0.7, linetype = "dashed")+
  geom_point( aes(col=Species), size = 2, alpha = 0.5)+
  theme_bw()+
```

```
  xlab("Number of individuals sampled")+
  ylab("Number of ASVs")+
  xlim(0,450)+
  theme(text=element_text(size=14))+
  theme(legend.position = "none")+
  scale_color_manual(values = palette)+
  ggtitle("Figure 1a")
```




```
### summarise percent ASVs detected in comparison to total number predicted to be within the regional sp

summary<-sac_df_all %>% group_by(Species) %>% summarize(max_asvs = max(Richness), predicted = max(Total

## `summarise()` ungrouping output (override with `.groups` argument)

summary$percent<-summary$max_asvs/summary$predicted
summary # summarizes percetage of ASVs detected per dataset

## # A tibble: 8 x 4
##   Species    max_asvs predicted percent
##   <chr>         <dbl>     <dbl>   <dbl>
## 1 Bat            2637     2936.   0.898
## 2 Deer           6442     6730.   0.957
## 3 Flamingo       2008     2191.   0.916
## 4 Human          2526     2637.   0.958
```

```
## 5 Meerkat        4475     5647.   0.792
## 6 Mouselemur     1213     1350.   0.898
## 7 Spinyrat       3102     3284.   0.945
## 8 Stint          2357     2497.   0.944
```

```
sac_df_all$Percent<-sac_df_all$Richness/sac_df_all$Total
head(sac_df_all)
```

```
##         Site Richness       SD    Total Species    Percent
## Human.1    1 215.6697 63.55003 2636.761   Human 0.08179342
## Human.2    2 349.5816 65.97117 2636.761   Human 0.13257994
## Human.3    3 451.1742 66.75617 2636.761   Human 0.17110926
## Human.4    4 535.9479 67.47951 2636.761   Human 0.20325999
## Human.5    5 608.3824 67.35917 2636.761   Human 0.23073098
## Human.6    6 673.3914 66.05011 2636.761   Human 0.25538585
```

```
sac_df_all$Species<-factor(sac_df_all$Species, levels = uniq)

#Fig 1b

ggplot(sac_df_all, aes(x = Site, y = Percent, group = Species))+
  geom_line(alpha=0.7, linetype = "dashed")+
  geom_point( aes(col=Species), size = 2, alpha = 0.9)+
  theme_bw()+
  xlab("Number of individuals sampled")+
  ylab("Percent ASVs detected")+
  xlim(0,360)+
  theme(text=element_text(size=14))+
  theme(legend.position = "none")+
  scale_color_manual(values = palette)+
  geom_hline(yintercept = 0.9, linetype = "dashed")+
  guides(colour = guide_legend(override.aes = list(size=7)))+
  ggtitle("Figure 1b")
```

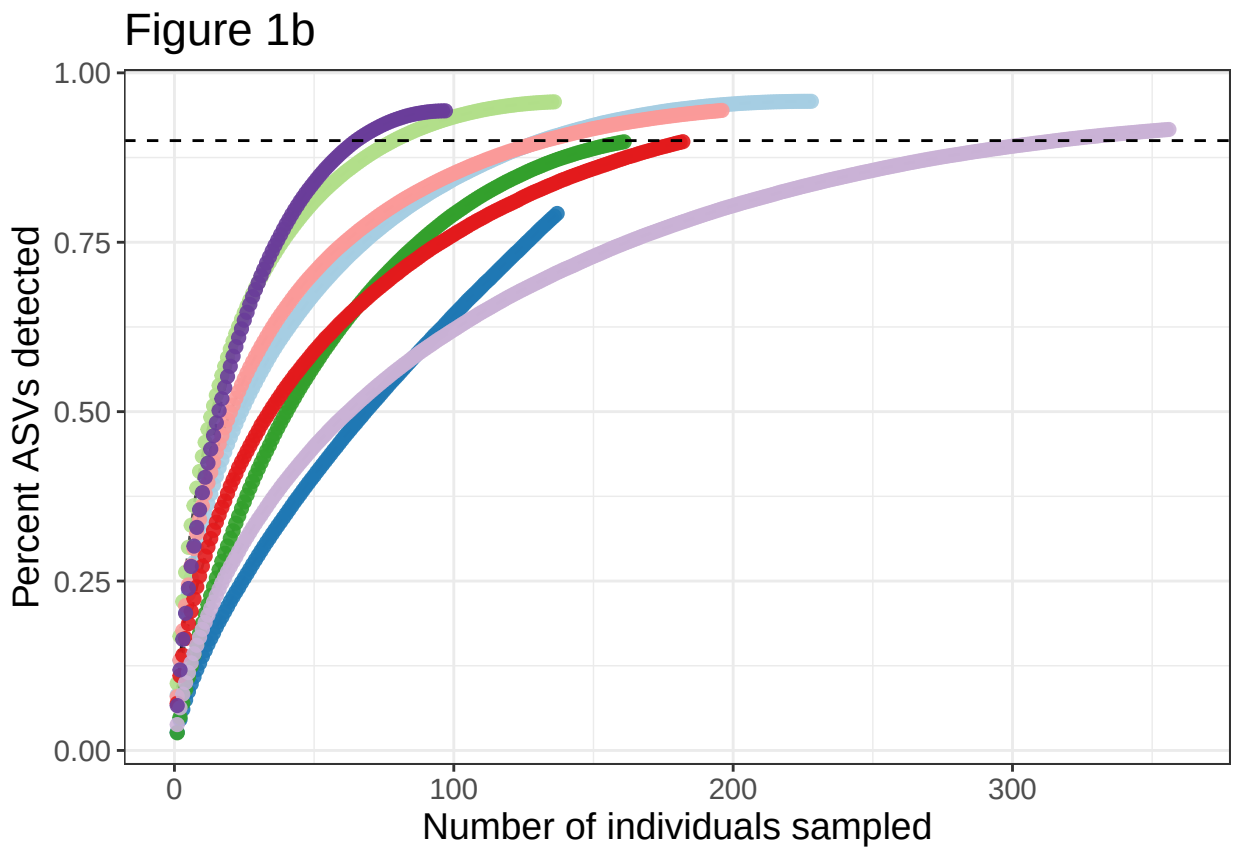


Figure 1c

Here we use the package ranacapa to generate rarefaction curves per sample per species dataset

```
rarefaction_fig<-list() #make an enpty list

uniq <- names(phylo_list) # make a list of species to subset sequentially

for (i in 1:length(uniq)){ #for species i
  data_1<-phylo_list[[i]] #subset the phyloseq object for species i
  data_1<-prune_taxa(taxa_sums(data_1)>0, data_1)  #remove any traces of taxa that are no longer presen


p <- ranacapa::ggrare(data_1, step = 500,   se = FALSE, plot = F)+
  xlim(c(0,30000))

#p + xlim(c(0,30000))

p1<- p +geom_line(col = palette[[i]], size = 0.1)+
  geom_vline(xintercept=10000)+
  geom_hline(yintercept=200, linetype = "dashed", size = 0.5)+
  theme(legend.position = "none")+
  theme_bw(base_size = 11)+
  theme(axis.text.x = element_blank(), axis.title.y = element_blank())+
```

```
  xlab(print(uniq[[i]]))+
  theme(plot.margin=unit(c(0.2,0.2,0.5,0.2),"cm"))

rarefaction_fig[[i]]<-p1


}
```

```
#Figure 1c


Fig1c<-ggarrange(rarefaction_fig[[1]], rarefaction_fig[[2]],rarefaction_fig[[3]],rarefaction_fig[[4]],
  rarefaction_fig[[5]],rarefaction_fig[[6]],rarefaction_fig[[7]],rarefaction_fig[[8]], ncol = 4, nrow =
  theme(plot.margin = margin(1,0.1,0.1,0.2, "cm"))

annotate_figure(Fig1c, fig.lab = "Figure 1c", fig.lab.size = 20,
  left = text_grob("Observed richness",  rot = 90, size = 16),
  bottom = text_grob("Number of reads sampled",  size = 16))
```

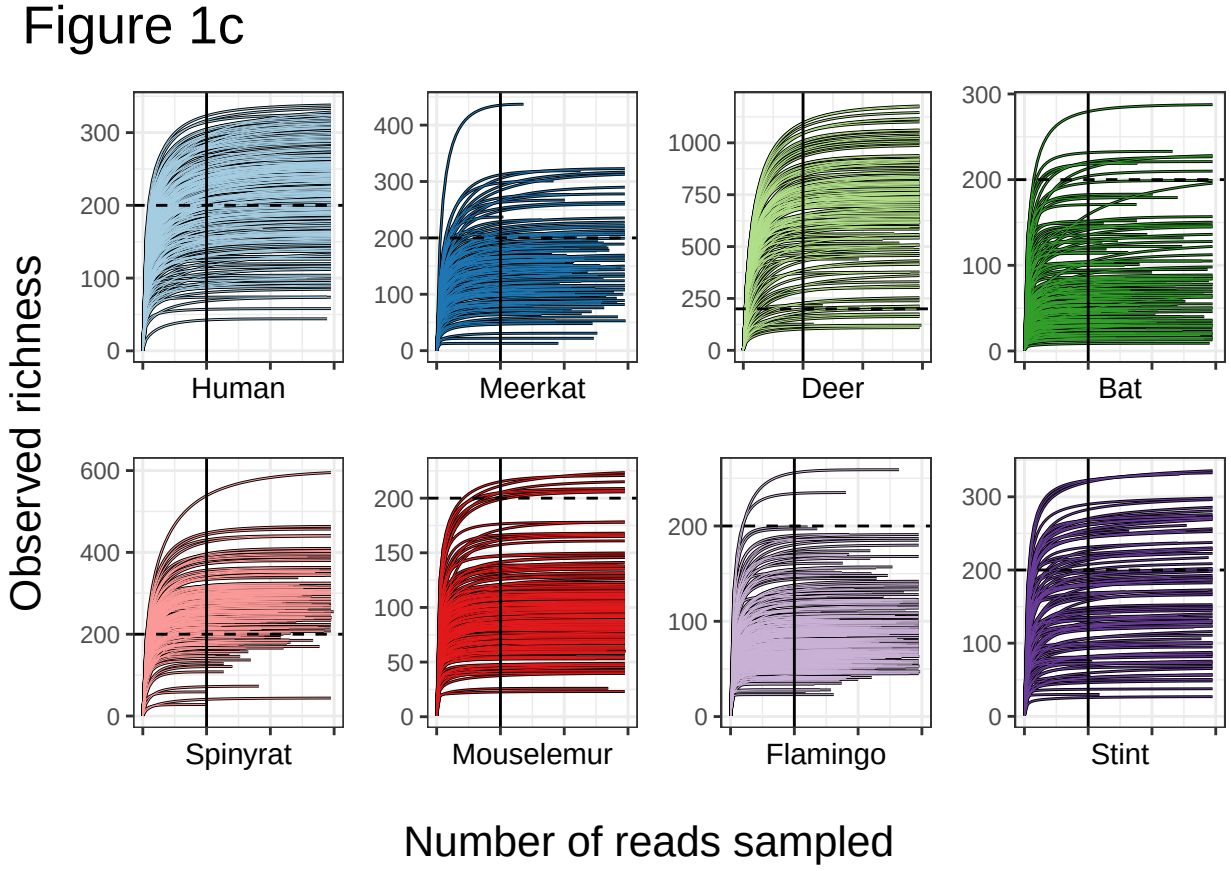
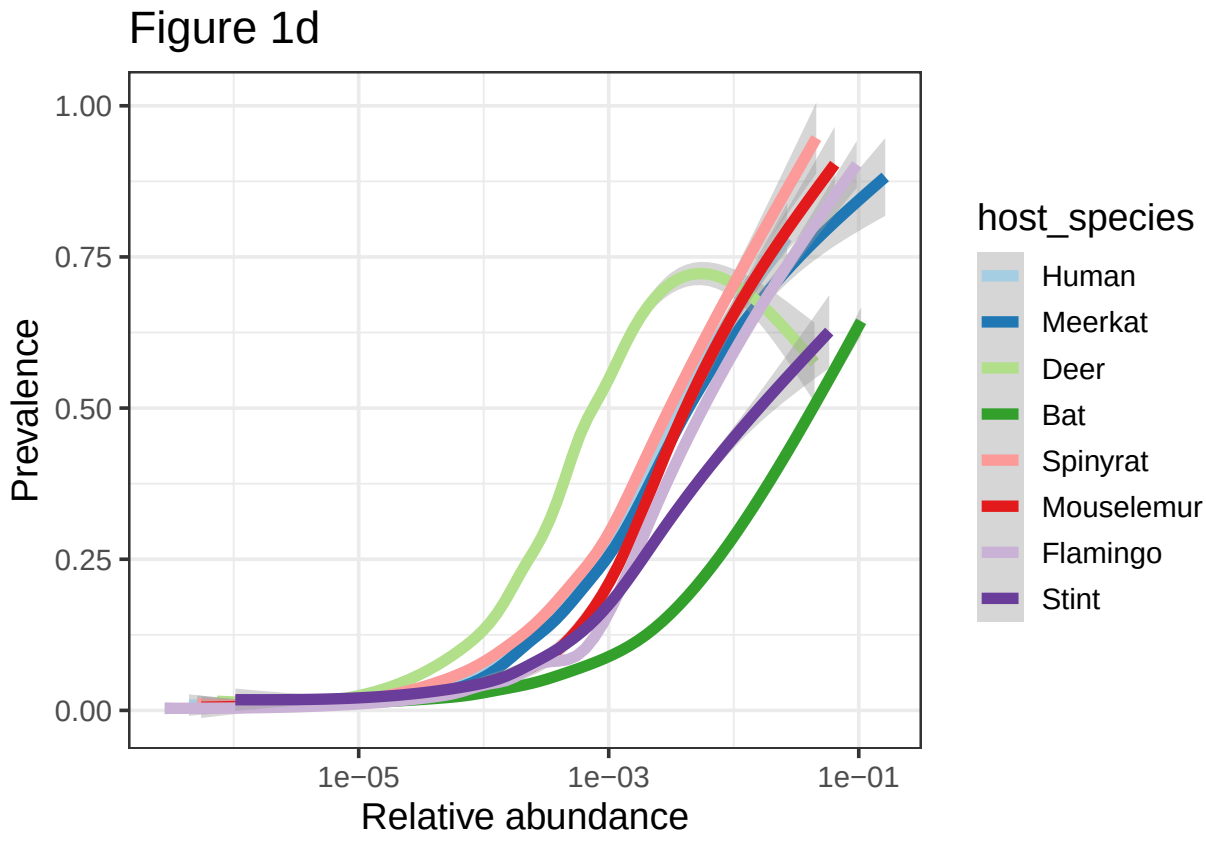


Figure 1d, e and f

Next we estimate prevalence and abundance values for every ASV per dataset and plot some macroecological patterns

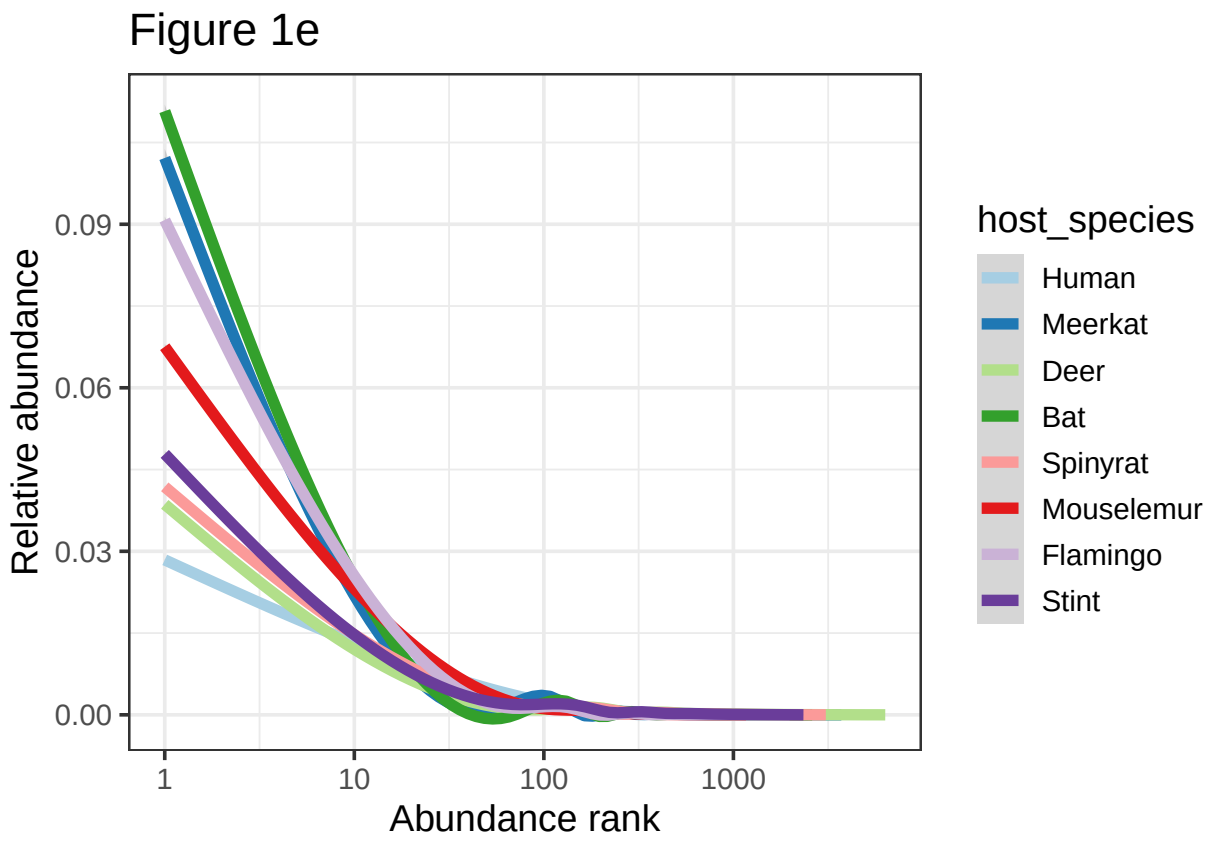
```r
## start loop to generate dataframe with prevalence and abundance of every ASV per species dataset

Prevlist<-list()
uniq <- names(phylo_list)


for (i in 1:length(uniq)){

  data_1<-phylo_list[[i]]  #for loop 1 (uniq)
  data_1<-prune_taxa(taxa_sums(data_1)>0, data_1)
  data_1<-rarefy_even_depth(data_1, sample.size = 10000, rngsee = 100, replace = TRUE, trimOTUs=TRUE,ve
  occupancy_abundance<-prevalence(data_1)
  occupancy_abundance$host_species <- as.character(uniq[[i]])
  occupancy_abundance$RelAbundance<- (occupancy_abundance$TotalAbundance/sum(occupancy_abundance$TotalA
  occupancy_abundance$RelPrev<-(occupancy_abundance$Prevalence / length(sample_data(data_1)$feature.id)
  occupancy_abundance$RelAbundanceInd<-(occupancy_abundance$TotalAbundance/ occupancy_abundance$Prevale
  occupancy_abundance$ASV<-taxa_names(data_1) #this is important! Don't use 'row.names' as it sneakily
  occupancy_abundance$Sample_size<-length(unique(sample_data(data_1)$Sample))
  Prevlist[[i]]<-occupancy_abundance

}

#combine loop outputs

occupancy_abundance_df<-do.call(rbind, Prevlist)

#make sure species in right order
occupancy_abundance_df$host_species<-factor(occupancy_abundance_df$host_species, levels = uniq)
#head(occupancy_abundance_df)

# Figure 1d

ggplot(occupancy_abundance_df, aes(y = RelPrev, x = RelAbundance, group = host_species, col = host_speci
  geom_smooth(size = 2)+
  theme_bw(base_size = 14)+
  scale_x_log10()+
  scale_color_manual(values = palette)+
  xlab("Relative abundance")+
  ylab("Prevalence")+
  ggtitle("Figure 1d")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



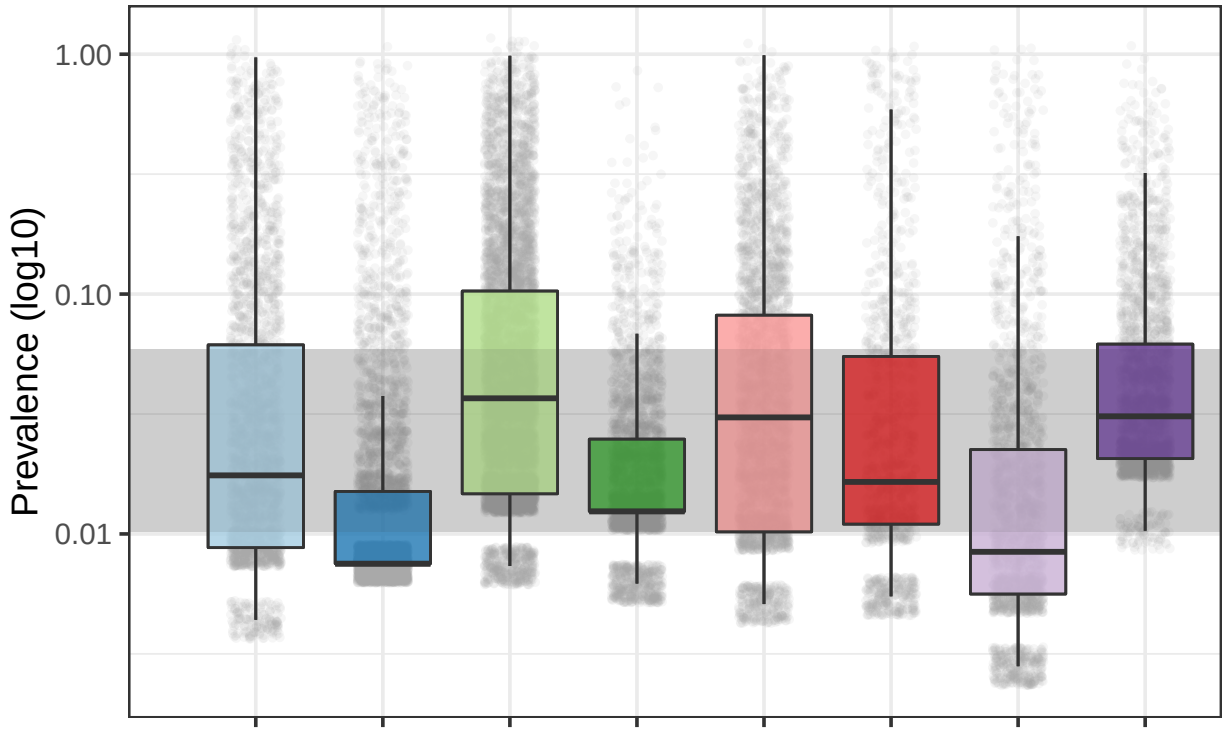

```
## add abundance rank as column

occupancy_abundance_df<-occupancy_abundance_df %>%
  arrange(host_species, -RelAbundance)%>%
  group_by(host_species) %>%
  mutate(Rank = rank(-RelAbundance))

# Fig 1e

ggplot(occupancy_abundance_df, aes(y = RelAbundance, x = Rank, group = host_species, col = host_species)
  geom_smooth(size = 2)+
  theme_bw(base_size = 14)+
  scale_x_log10()+
  scale_color_manual(values = palette)+
  ylab("Relative abundance")+
  xlab("Abundance rank")+
  ggtitle("Figure 1e")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
# Fig 1f

ggplot(occupancy_abundance_df, aes(y = RelPrev, x = host_species))+
  scale_fill_manual(values = palette)+
  geom_jitter(alpha = 0.1, size = 1, width = 0.2, height = 0.08, col = "darkgrey")+
  geom_boxplot(aes(fill = host_species), alpha=0.6, outlier.shape = NA)+
  scale_y_log10()+
  theme_bw(base_size = 14)+
  ylab("Prevalence (log10)")+xlab("")+
  theme(legend.position = "none")+
  theme(axis.text.x = element_blank())+
  annotate("rect", ymin=summary(occupancy_abundance_df$RelPrev)[2],
    ymax=summary(occupancy_abundance_df$RelPrev)[5], xmin=0, xmax=Inf, alpha = .3)+
  geom_boxplot(aes(fill = host_species), alpha=0.5, outlier.shape = NA)+
  ggtitle("Figure 1f")
```

Figure 1f



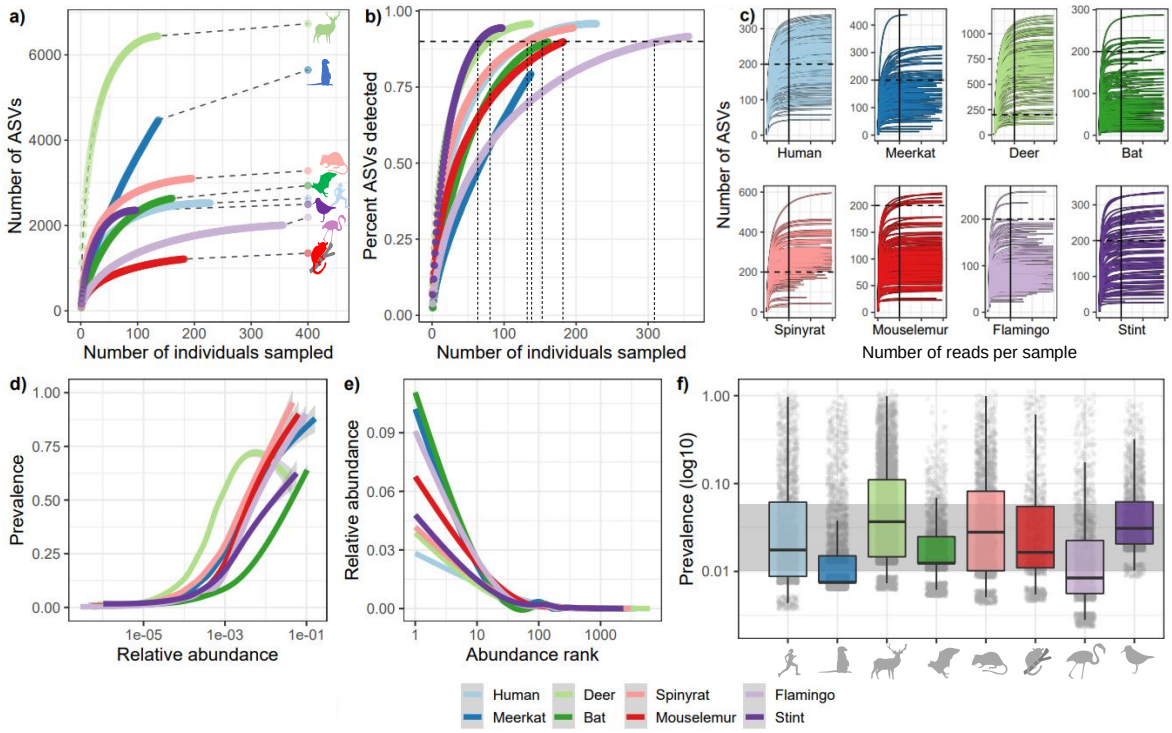

```
### Estimate mean and median prevalence values for all datasets

print(mean(occupancy_abundance_df$RelPrev))*100
```

```
## [1] 0.06834305

## [1] 6.834305
```

```
print(median(occupancy_abundance_df$RelPrev))*100
```

```
## [1] 0.02061856

## [1] 2.061856
```

Figure 1



Alpha diversity

Next we generate alpha diversity estimates per sample and per prevalence threshold

```
## Nested loop to estimate alpha diversty metrics per [j] species and [i] prevalence threshold

list1<-list()

uniq<-names(phylo_list)

thresholds<-c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9)

# for loop 1 to repeat for every species

for (j in 1:length(uniq)){

data<-phylo_list[[j]]  #for loop 1 (uniq)
data<-prune_taxa(taxa_sums(data) > 0, data)
data<-rarefy_even_depth(data, sample.size = 10000, rngsee = 100, replace = TRUE, trimOTUs=TRUE,verbose=

#### for loop 2 to repeat for every prevalence threshold

list2<-list()

## for loop 2 (thresholds)

for (i in 1:length(thresholds)){
```

```
   tryCatch({ #catch errors

data_subset<-phyloseq_filter_prevalence(data, prev.trh = thresholds[i]) #edit

alpha<-estimate_richness(data_subset, measures=c("Observed", "Shannon"))
alpha$Faiths<-metagMisc::phyloseq_phylo_div(data_subset, measures = c("PD"))$PD
alpha$BWPD<-estimate_bwpd(data_subset)$PSEs

alpha$Sample<-sample_names(data_subset)
alpha$Species<-as.character(uniq[[j]])
alpha$Prevalence<-thresholds[i] #edit

list2[[i]]<-alpha

   }, error=function(e){})

}

alpha_df<-do.call(rbind, list2)

list1[[j]]<-alpha_df

}
```

```
## combine

alpha_df_all<-do.call(rbind, list1)

### change NAs into zeros

alpha_df_all$Faiths[is.na(alpha_df_all$Faiths)] <- 0
alpha_df_all$BWPD[is.na(alpha_df_all$BWPD)] <- 0

################## generate standarised values per species

alpha_df_all<-transform(alpha_df_all, Observed_scaled=ave(Observed, Species, FUN=scale))
alpha_df_all<-transform(alpha_df_all, Faiths_scaled=ave(Faiths, Species, FUN=scale))
alpha_df_all<-transform(alpha_df_all, Shannon_scaled=ave(Shannon, Species, FUN=scale))
alpha_df_all<-transform(alpha_df_all, BWPD_scaled=ave(BWPD, Species, FUN=scale))


alpha_df_all<-alpha_df_all[,c(5,6,7,1,2,3,4,8,9,10,11)]
```

Beta dissimilarity

Next we do the same for beta dissimilarity

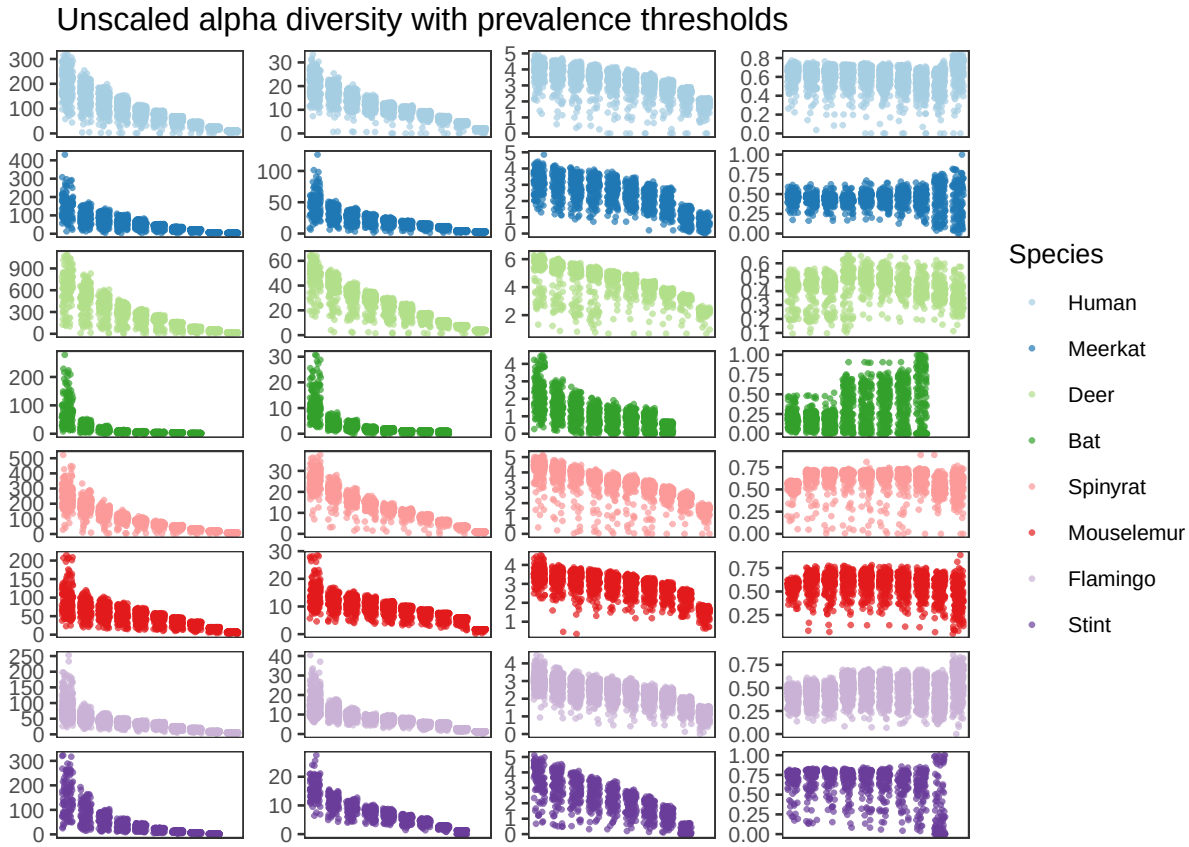
```
# Nested loop

list1<-list()

uniq<-names(phylo_list)
```

```
thresholds<-c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9)

# for loop 1 to repeat for every species

for (j in 1:length(uniq)){

  data<-phylo_list[[j]]  #for loop 1 (uniq)
  data<-prune_taxa(taxa_sums(data) > 0, data)
  data<-rarefy_even_depth(data, sample.size = 10000, rngsee = 100, replace = TRUE, trimOTUs=TRUE,verbos

  #### for loop 2 to repeat for every prevalence threshold

  list2<-list()

  ## for loop 2 (thresholds)

  for (i in 1:length(thresholds)){

    tryCatch({ #catch errors

      data_subset<-phyloseq_filter_prevalence(data, prev.trh = thresholds[i]) #edit
    #  data_subset<-phyloseq_filter_prevalence(data, prev.trh = 0.5) #edit
     # data_subset<-prune_taxa(taxa_sums(data_subset) > 0, data_subset)

      unifrac<-as.matrix(phyloseq::distance(data_subset, method = "unifrac"))
      wunifrac<-as.matrix(phyloseq::distance(data_subset, method = "wunifrac"))
      morisita<-as.matrix(phyloseq::distance(data_subset, method = "morisita"))
      jaccard<-as.matrix(phyloseq::distance(data_subset, method = "jaccard"))

      beta<-data.frame(colMeans(unifrac, na.rm=T))
      names(beta)<-"unifrac"
      beta$wunifrac<-colMeans(wunifrac, na.rm=T)
      beta$morisita<-colMeans(morisita, na.rm=T)
      beta$jaccard<-colMeans(jaccard, na.rm=T)

      beta$Sample<-sample_names(data_subset)
      beta$Species<-as.character(uniq[[j]])
      beta$Prevalence<-thresholds[i] #edit

      list2[[i]]<-beta

    }, error=function(e){})

  }

  beta_df<-do.call(rbind, list2)

  list1[[j]]<-beta_df

}

# combine loop outputs
```

```
beta_df_all<-do.call(rbind, list1)
```

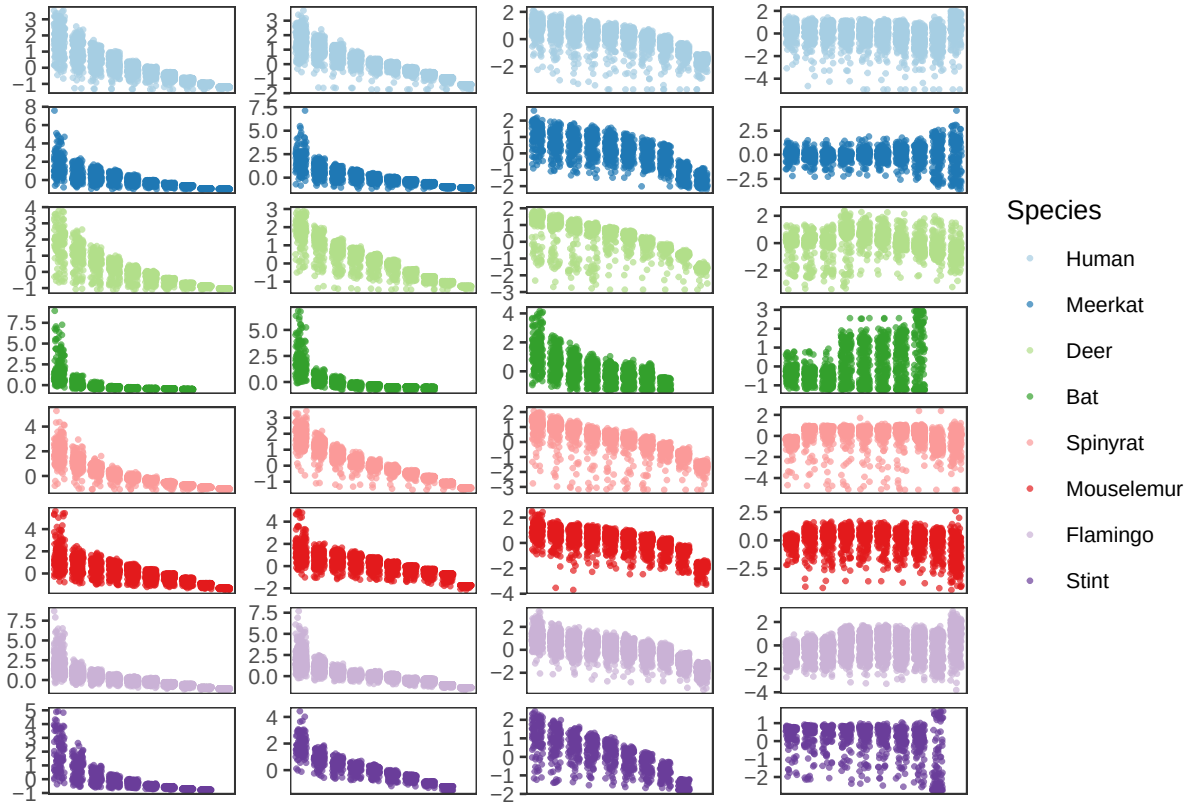
Figures 2 and 3

```
############# alpha diversity

alpha_df_all$Species<-factor(alpha_df_all$Species, levels = uniq)
alpha_df_all$Prevalence<-factor(alpha_df_all$Prevalence)

## only used scaled values
alpha_short_scaled<-alpha_df_all[,c(8,9,10,11,1,2,3)]
alpha_long_scaled<-gather(alpha_short_scaled, Index, Distance, Observed_scaled, Faiths_scaled, Shannon_s
alpha_long_scaled$Index<-factor(alpha_long_scaled$Index, level = c("Observed_scaled", "Faiths_scaled",


# generate dataset for raw values too, for comparison
alpha_short_unscaled<-alpha_df_all[,c(4,5,6,7,1,2,3)]
alpha_long_unscaled<-gather(alpha_short_unscaled, Index, Distance, Observed, Faiths, Shannon, BWPD, fac
alpha_long_unscaled$Index<-factor(alpha_long_unscaled$Index, level = c("Observed", "Faiths", "Shannon",

## first look at raw (unscaled) alpha diversity

ggplot(alpha_long_unscaled, aes(x =Prevalence, y = Distance))+
  geom_jitter(aes(fill = Species, col = Species), pch=21, size=0.5,  alpha = 0.7, width =0.3)+
  # geom_boxplot(alpha = 0.7, outlier.shape = NA) +
  facet_wrap(~Species+Index, ncol = 4, scales = "free_y")+
  scale_fill_manual(values = palette)+
  scale_color_manual(values = palette)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  theme_bw(base_size = 10)+
  ylab("")+
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
# theme(legend.position="none")+
  theme(strip.background = element_blank(), strip.text.x = element_blank())+
  ggtitle("Unscaled alpha diversity with prevalence thresholds")
```



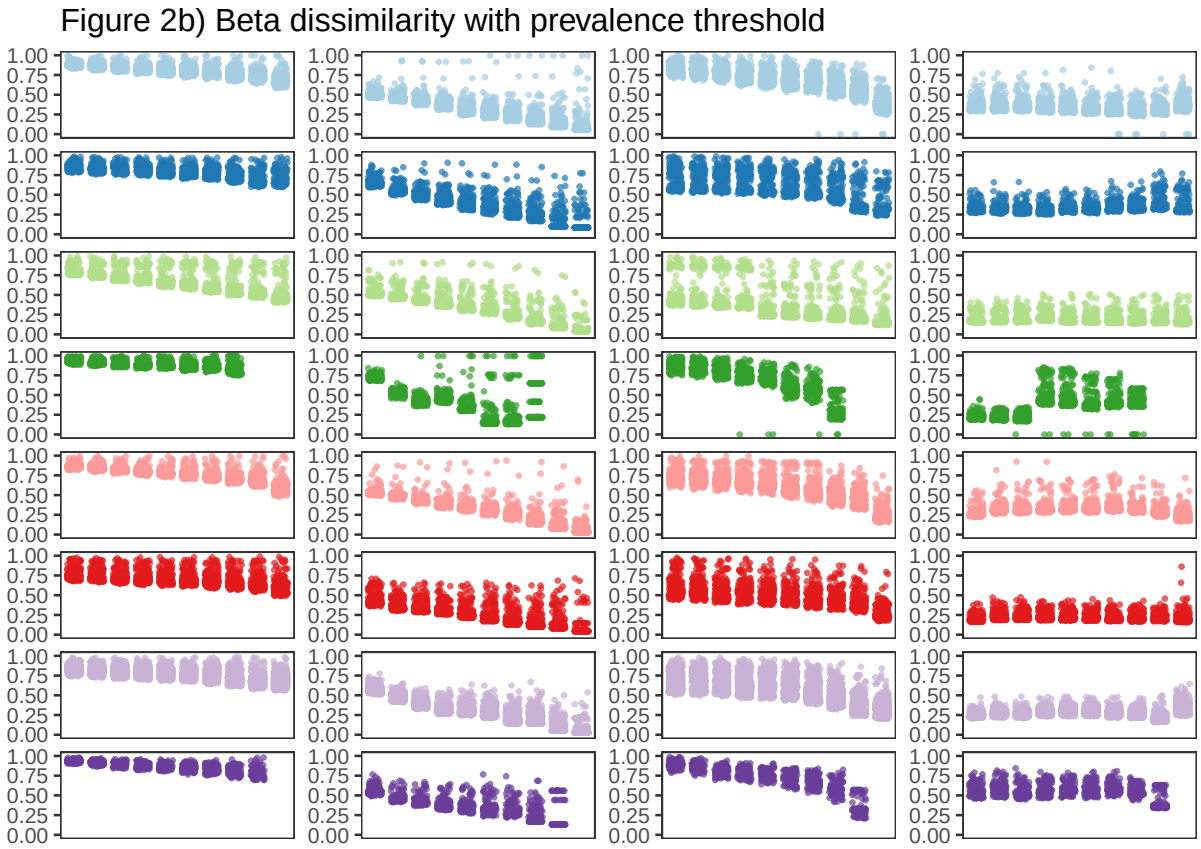

```
##############################

# Now Fig 2a - scaled alpha diversity with increasing prevalence thresholds


ggplot(alpha_long_scaled, aes(x =Prevalence, y = Distance))+
  geom_jitter(aes(fill = Species, col = Species), pch=21, size=0.5,  alpha = 0.7, width =0.3)+
  facet_wrap(~Species+Index, ncol = 4, scales = "free_y")+
  scale_fill_manual(values = palette)+
  scale_color_manual(values = palette)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  theme_bw(base_size = 10)+
  ylab("")+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
 # theme(legend.position="none")+
  theme(strip.background = element_blank(), strip.text.x = element_blank())+
  ggtitle("Fig2a) Scaled alpha diversity with prevalence thresholds")
```

Fig2a) Scaled alpha diversity with prevalence thresholds




```
#################################### beta diversity ######################
#################################### beta diversity ######################
#################################### beta diversity ######################
#################################### beta diversity ######################

beta_df_all$Species<-factor(beta_df_all$Species, levels = uniq)
beta_df_all$Prevalence<-factor(beta_df_all$Prevalence)
beta_short_normal<-beta_df_all[,c(1:7)]
beta_long_normal<-gather(beta_short_normal, Index, Distance, jaccard, unifrac,  morisita,  wunifrac, fa
beta_long_normal$Index<-factor(beta_long_normal$Index, level = c("jaccard", "unifrac",  "morisita",  "w


##############################

# Figure 2b


ggplot(beta_long_normal, aes(x =Prevalence, y = Distance))+
  geom_jitter(aes(fill = Species, col = Species), pch=21, size=0.5, alpha = 0.7, width =0.3)+
  facet_wrap(~Species+Index, ncol = 4, scales = "free_y")+
  scale_fill_manual(values = palette)+
  scale_color_manual(values = palette)+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))+
  theme_bw(base_size = 10)+
  ylab("")+
  ylim(0,1)+
```

```
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+
  theme(axis.title.x=element_blank(),axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank())+theme(legend.position="
 # theme(legend.position="none")+
  theme(strip.background = element_blank(), strip.text.x = element_blank())+
  ggtitle("Figure 2b) Beta dissimilarity with prevalence threshold")
```




```
#### Bartlett's variance test


bartlett.test(alpha_short_scaled$Observed_scaled, alpha_short_scaled$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  alpha_short_scaled$Observed_scaled and alpha_short_scaled$Prevalence
## Bartlett's K-squared = 17262, df = 9, p-value < 2.2e-16
```

```
bartlett.test(alpha_short_scaled$Faiths_scaled, alpha_short_scaled$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  alpha_short_scaled$Faiths_scaled and alpha_short_scaled$Prevalence
## Bartlett's K-squared = 10496, df = 9, p-value < 2.2e-16
```

```r
bartlett.test(alpha_short_scaled$Shannon_scaled, alpha_short_scaled$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  alpha_short_scaled$Shannon_scaled and alpha_short_scaled$Prevalence
## Bartlett's K-squared = 1231.2, df = 9, p-value < 2.2e-16
```

```r
bartlett.test(alpha_short_scaled$BWPD_scaled, alpha_short_scaled$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  alpha_short_scaled$BWPD_scaled and alpha_short_scaled$Prevalence
## Bartlett's K-squared = 1037.2, df = 9, p-value < 2.2e-16
```

```r
bartlett.test(beta_df_all$jaccard, beta_df_all$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  beta_df_all$jaccard and beta_df_all$Prevalence
## Bartlett's K-squared = 798.17, df = 9, p-value < 2.2e-16
```

```r
bartlett.test(beta_df_all$unifrac, beta_df_all$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  beta_df_all$unifrac and beta_df_all$Prevalence
## Bartlett's K-squared = 941.12, df = 9, p-value < 2.2e-16
```

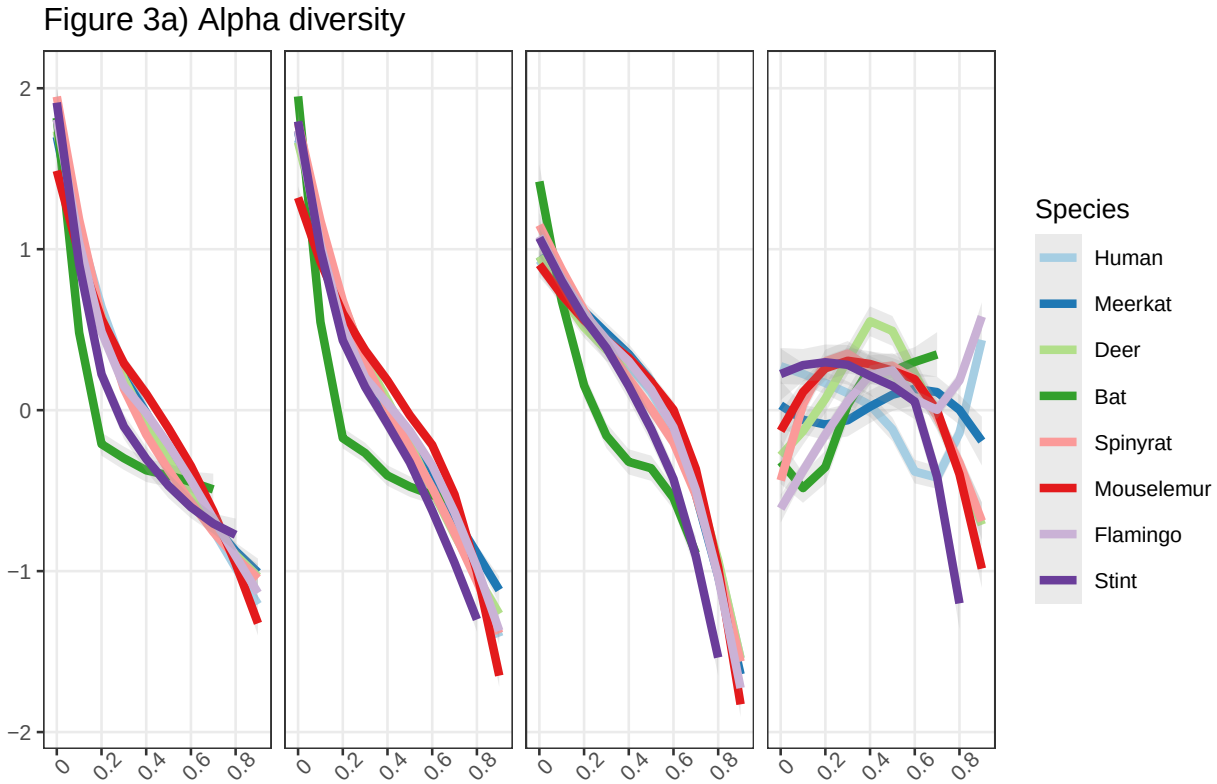
```r
bartlett.test(beta_df_all$morisita, beta_df_all$Prevalence)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  beta_df_all$morisita and beta_df_all$Prevalence
## Bartlett's K-squared = 74.538, df = 9, p-value = 1.95e-12
```

```r
bartlett.test(beta_df_all$wunifrac, beta_df_all$Prevalence)
```

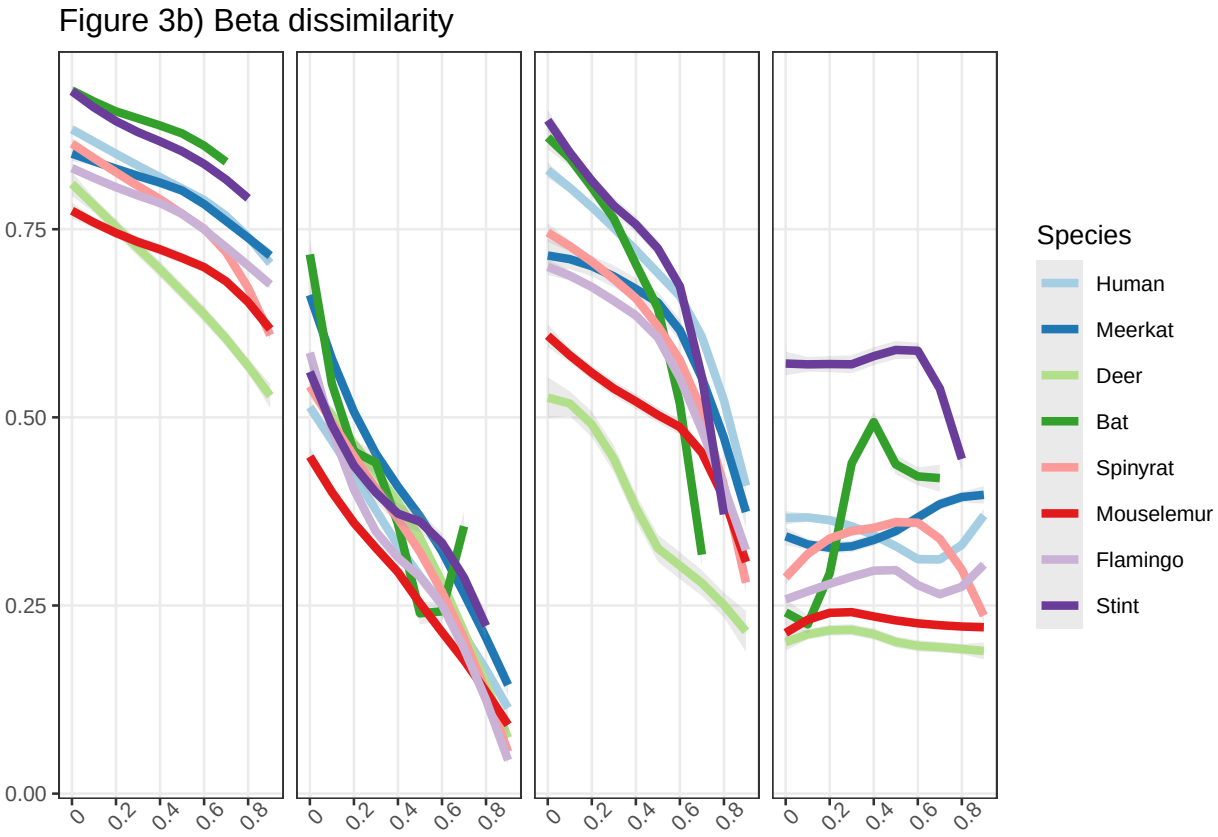
```
##
##  Bartlett test of homogeneity of variances
##
## data:  beta_df_all$wunifrac and beta_df_all$Prevalence
## Bartlett's K-squared = 126.44, df = 9, p-value < 2.2e-16
```

```
## Now lets compare mean values

# Figure 3a

ggplot(alpha_long_scaled, aes(x =Prevalence, y = Distance, group = Species))+
  geom_smooth(aes(col = Species), alpha = 0.2, method = "loess", size = 1.5)+
  scale_color_manual(values = palette)+
  facet_wrap(~Index, ncol = 8)+
  theme_bw(base_size = 10)+
  theme( panel.grid.minor = element_blank())+
 # theme(legend.position="none")+
  theme(strip.background = element_blank(),strip.text.x = element_blank())+
  theme(axis.title = element_blank())+
  theme(axis.text.x = element_text(angle = 45))+
  theme(plot.margin=unit(c(0.2,0.2,0.8,0.2),"cm"))+
  scale_x_discrete(breaks=c(0,0.2,0.4,0.6,0.8))+
  ggtitle("Figure 3a) Alpha diversity")
```




```
# Figure 3b

ggplot(beta_long_normal, aes(x =Prevalence, y = Distance, group = Species))+
  geom_smooth(aes(col = Species), alpha = 0.2, method = "loess", size = 1.5)+
  scale_color_manual(values = palette)+
  facet_wrap(~Index, ncol = 8)+
  theme_bw(base_size = 10)+
```

```
 theme( panel.grid.minor = element_blank())+
# theme(legend.position="none")+
 theme(axis.title = element_blank())+
 theme(strip.background = element_blank(), strip.text.x = element_blank())+
 theme(axis.text.x = element_text(angle = 45))+
 scale_x_discrete(breaks=c(0,0.2,0.4,0.6,0.8))+
 ggtitle("Figure 3b) Beta dissimilarity")
```



END