Vignette for phyloseq: A Bioconductor package for handling and analysis of high-throughput phylogenetic sequence data

Paul J. McMurdie and Susan Holmes*
Statistics Department, Stanford University,
Stanford, CA 94305, USA
*E-mail: mcmurdie@stanford.edu
https://github.com/joey711/phyloseq

March 9, 2012

Contents

1 Summary

There are already several ecology and phylogenetic packages available in R, including the adephylo, vegan, ade4, picante, ape, phangorn, phylobase, and OTUbase packages. These can already take advantage of many of the powerful statistical and graphics tools available in R. However, at present a user must devise their own methods for parsing the output of their favorite OTU clustering application, and, as a consequence, there is also no standard within Bioconductor (or R generally) for storing or sharing the suite of related data objects that describe a phylogenetic sequencing project. The phyloseq package seeks to address these issues by providing a related set of S4 classes that internally manage the handling tasks associated with organizing, linking, storing, and analyzing phylogenetic sequencing data. *phyloseq* additionally provides some convenience wrappers for input from common clustering applications, common analysis pipelines, and native implementation of methods that are not available in other R packages.

2 About this vignette

A separate vignette is included within the phyloseq-package that describes the basics of importing pre-clustered phylogenetic sequencing data, data filtering, as well as some transformations and some additional details about the package and installation. A quick way to load it is:

```
> vignette("phyloseq_basics")
```

By contrast, this vignette is intended to provide functional examples of the analysis tools and wrappers included in phyloseq. All necessary code for performing the analysis and producing graphics will be included with its description, and the focus will be on the use of example data that is included and documented within the phyloseq-package.

Let's start by loading the `phyloseq-package`:

```
> library("phyloseq")
```

3 Simple exploratory graphics

3.1 Easy Richness Estimates

```
> data(GlobalPatterns)
> (p <- plot_richness_estimates(GlobalPatterns, "SampleType", "SampleType"))
```

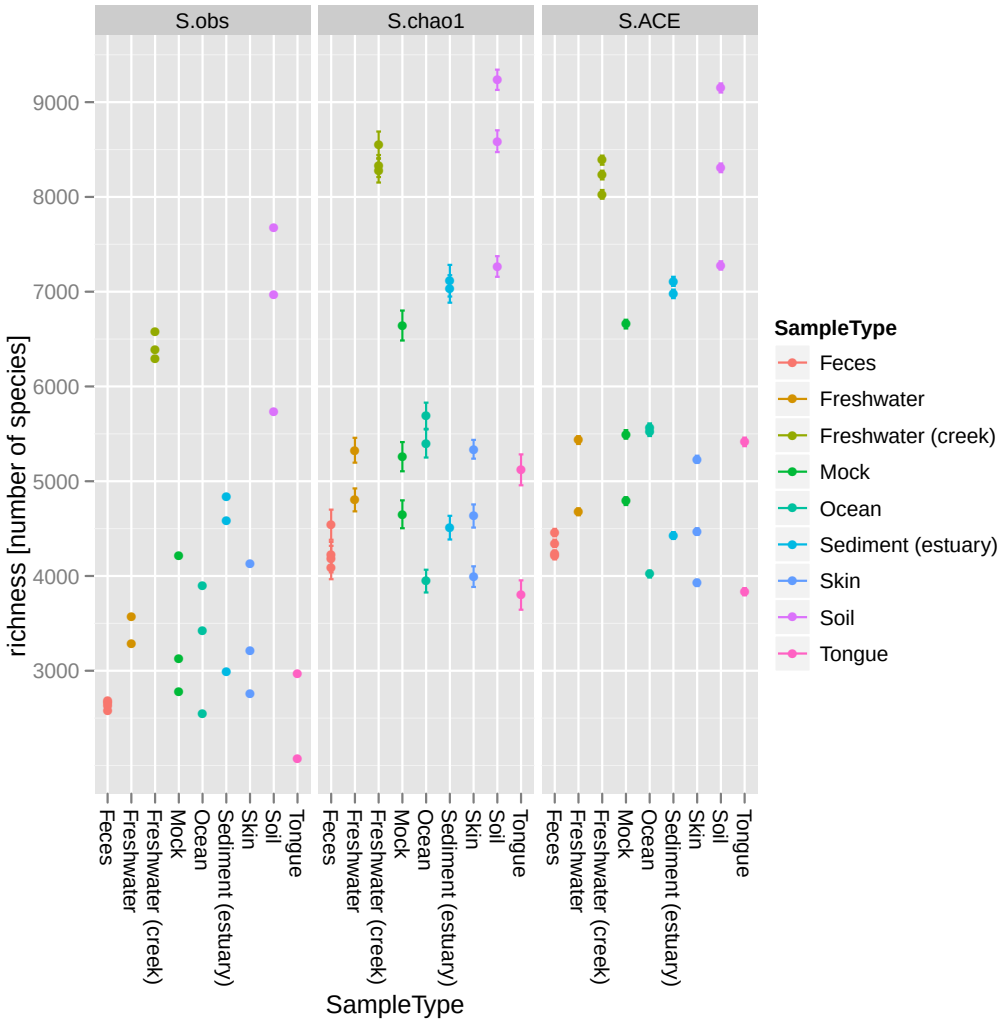


Figure 1: Estimates of the species richness of samples in the "Global Patterns" dataset.

3.2 Exploratory tree plots

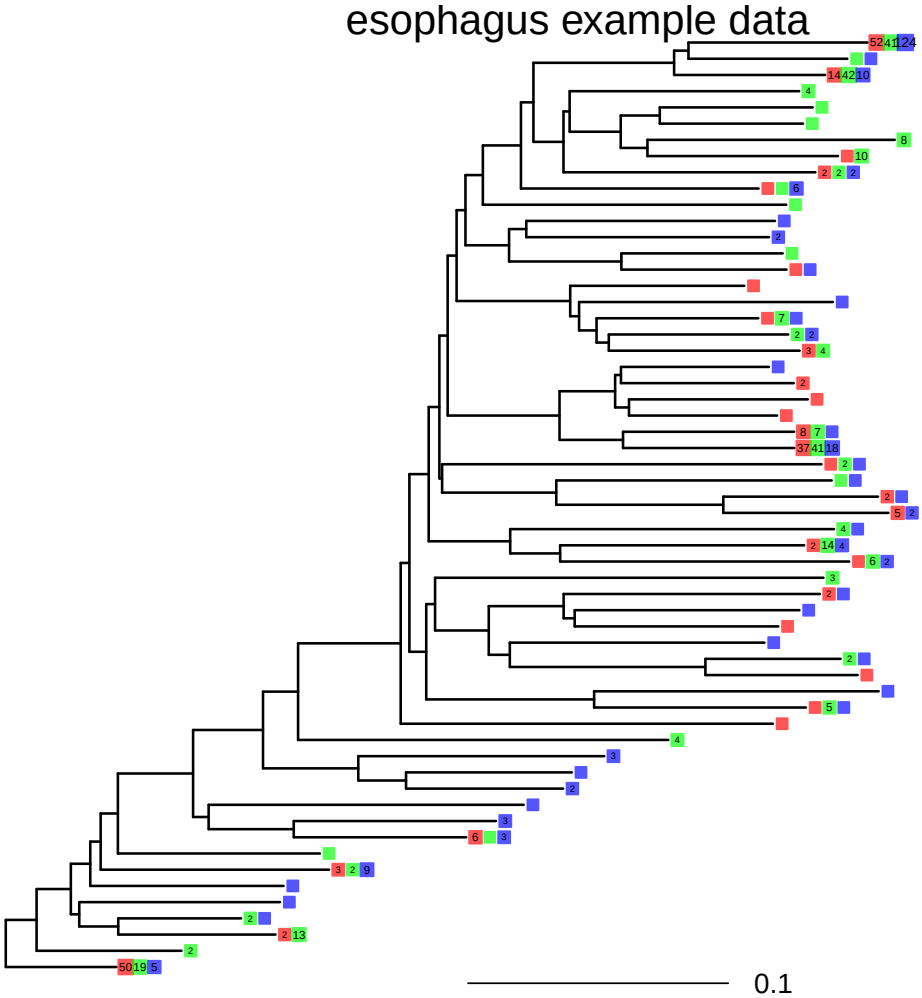
phyloseq also contains a method for easily annotating a phylogenetic tree with information regarding the sample in which a particular taxa was observed, and optionally the numbe of individuals that were observed. In the following example we use the included "esophagus" dataset.

`esophagus` is comprised of only 3 samples and so does not include any `sampleData`. For our tree example we must first make dummy `sampleData` that only refers to the 3 sample names.

```
> data(esophagus)
> es1              <- esophagus
> sn               <- sample.names(esophagus)
> sampleData(es1) <- sampleData(data.frame(sample=sn, row.names=sn))
```

And now we will create the tree graphic, grouping/coloring by our dummy sample-name variable, and also labelling the number of individuals observed in each sample (if at all). The symbols are slightly enlarged as the number of individuals increases.

```
> plot_tree_phyloseq(es1, color_factor="sample",
+                          type_abundance_value=TRUE, treeTitle="esophagus example data")
```



3.3 Exploratory bar plots

In the following example we use the included "enterotype" dataset [1].

```
> data(enterotype)
```

We start with a simple rank abundance barplot, using the cumulative fractional abundance of each Operational Taxonomic Unit (OTU) in the dataset. In this particular example, the available published data are pre-processed/simplified as sample-wise fractional occurrences (rather than counts of individuals; preffered), and OTUs are clustered/labeled at the genus level. For the barplot in Figure 2, we further normalize by the total number of samples (280).

```
> par(mar = c(10, 4, 4, 2) + 0.1) # make more room on bottom margin for genera names
> N <- 30
> barplot(sort(speciesSums(enterotype), TRUE)[1:N]/nsamples(enterotype), las=2)
```

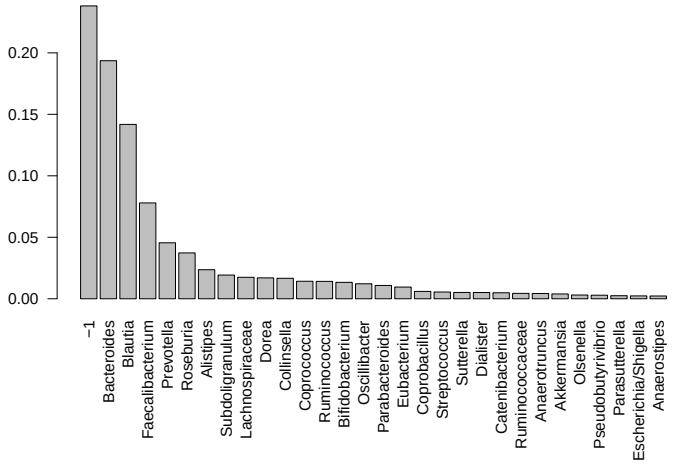


Figure 2: An example exploratory barplot using base `R` graphics and the `speciesSums` and `nsamples` functions.

Note that this first barplot is clipped at the 30th OTU. This was chosen because `nspecies(enterotype)` = 553 OTUs would not be legible on the plot. As you can see, the relative abundances have decreased dramatically by the 10th-ranked OTU.

So what are these OTUs? In the `enterotype` dataset, only a single taxonomic rank type is present:

```
> rank.names(enterotype)
```

```
[1] "Genus"
```

This means the OTUs in this dataset have been grouped at the level of genera, and no other taxonomic grouping/transformation is possible without additional information (like might be present in a phylogenetic tree, or with further taxonomic classification analysis).

We need to know which taxonomic rank classifiers, if any, we have available to specify in the second barplot function in this example, `plot_taxa_bar()`. We have already observed how quickly the abundance decreases with rank, so wo we will subset the enterotype dataset to the most abundant `N` taxa in order to make the barplot legible on this page.

```
> TopNOTUs <- names(sort(speciesSums(enterotype), TRUE)[1:10])
> entTop   <- prune_species(TopNOTUs, enterotype)
> print(entTop)

phyloseq-class experiment-level object
OTU Table:          [10 species and 280 samples]
                     species are rows
Sample Map:         [280 samples by 9 sample variables]:
Taxonomy Table:     [10 species by 1 taxonomic ranks]:
```

Note also that there are 280 samples in this dataset, and so a remaining challenge is to consolidate these samples into meaningful groups. A good place to look is the available sample variables, which in most cases will carry more "meaning" than the sample names alone.

```
> sample.variables(entTop)

[1] "Enterotype"     "Sample_ID"      "SeqTech"        "SampleID"
[5] "Project"        "Nationality"    "Gender"         "Age"
[9] "ClinicalStatus"
```

The parameters to `plot_taxa_bar` in the following code-chunk were chosen after various trials. We suggest that you also try different parameter settings while you're exploring different features of the data. In addition to the variables names of `sampleData`, the `plot_taxa_bar()` function recognizes a special parameter name `"TaxaGroup"`, which is not (should not be) a sample variable name in `sampleData(enterotype)`, but instead indicates that the particular graphic parameter should group values by the taxanomic rank specified in the `taxavec` argument. In this example we have also elected to separate the samples by "facets" (separate, adjacent sub-plots) according to the enterotype to which they have been assigned. Within each enterotype facet, the samples are further separated by sequencing technology, and the genera is indicated by fill color. Multiple samples having the same enterotype designation and sequencing technology are plotted side-by-side as separate bars.

```
> p <- plot_taxa_bar(entTop, taxavec="Genus", x_category="SeqTech", fill_category = "TaxaGroup")
> p <- p + facet_wrap(~Enterotype) # Do a facet_wrap
> print(p)
```

Figure 3 summarizes quantitatively the increased abundances of *Bacteroides* and *Prevotella* in the Enterotypes 1 and 2, respectively. Interestingly, a large relative abundance of *Blautia* was observed for Enterotype 3, but only from 454-pyrosequencing data sets, not the Illumina or Sanger datasets. This suggests the increased *Blautia* might actually be an artifact. Similarly, *Prevotella* appears to be one of the most abundant genera in the Illumina-sequenced samples among Enterotype 3, but this is not reproduced in the 454-pyrosequencing or Sanger sequencing data.

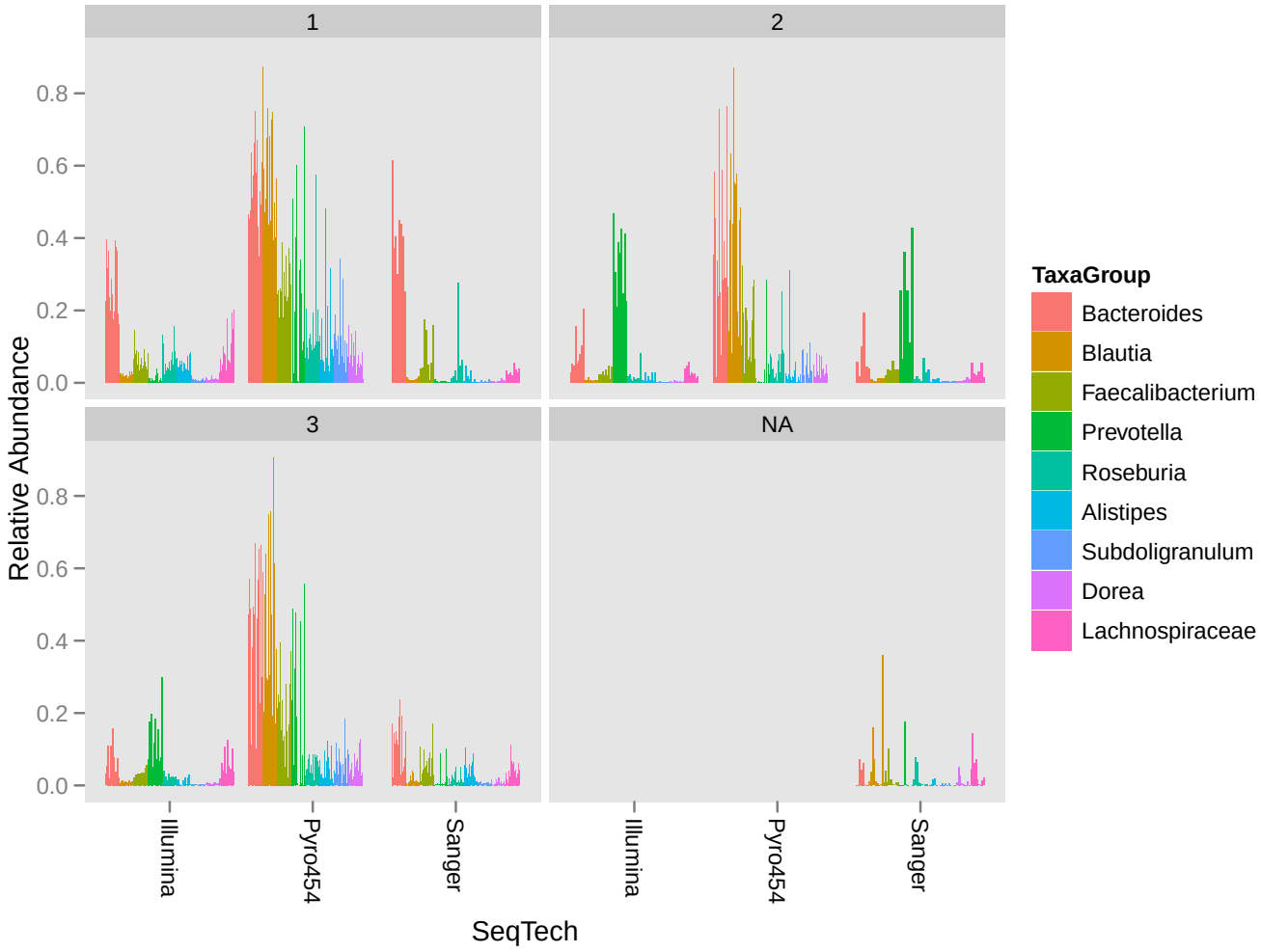


Figure 3: An example exploratory barplot using the `plot_taxa_bar()` function. In this case we have faceted the samples according to their assigned Enterotype. Within each Enterotype facet, the samples are further separated by sequencing technology, and each genera is shaded a different color. Multiple samples from the same Enterotype and sequencing technology are plotted side-by-side as separate bars (dodged).

4 Exploratory analysis and graphics

4.1 Microbiome Graphical/Network Models

Continuing with the `enterotype` dataset, here are some examples for creating custom graphical models of the relationship between microbiome samples in an experiment. This relies heavily on the `igraph` package, and uses `ggplot2` to create a graphical display of the "connectedness" of samples according to some user-provided ecological similarity.

```
> data(enterotype)
> ig <- make_sample_network(enterotype, FALSE, max.dist=0.3)
> (p  <- plot_sample_network(ig, enterotype,
+           color="SeqTech", shape="Enterotype", line_weight=0.3, label=NULL))
```

Interestingly, at this level of analysis and parameter-settings the two major sub-graphs appear to be best explained by the sequencing technology and not the subject enterotype (Figure 4), suggesting that the choice of sequencing technology has a major effect on the microbial community one can observe. This seems to differ somewhat with the inferences described in the "enterotype" article [1]. However, there could be some confounding or hidden variables that might also explain this phenomenon, and the well-known differences in the sequence totals between the technologies may also be an important factor. Furthermore, since this is clearly an experimental artifact (and they were including data from multiple studies that were not orginally planned for this purpose), it may be that the enterotype observation can also be shown in a network analysis of this data after removing the effect of sequencing technology and related sequencing effort. Such an effort would be interesting to show here, but is not yet included.

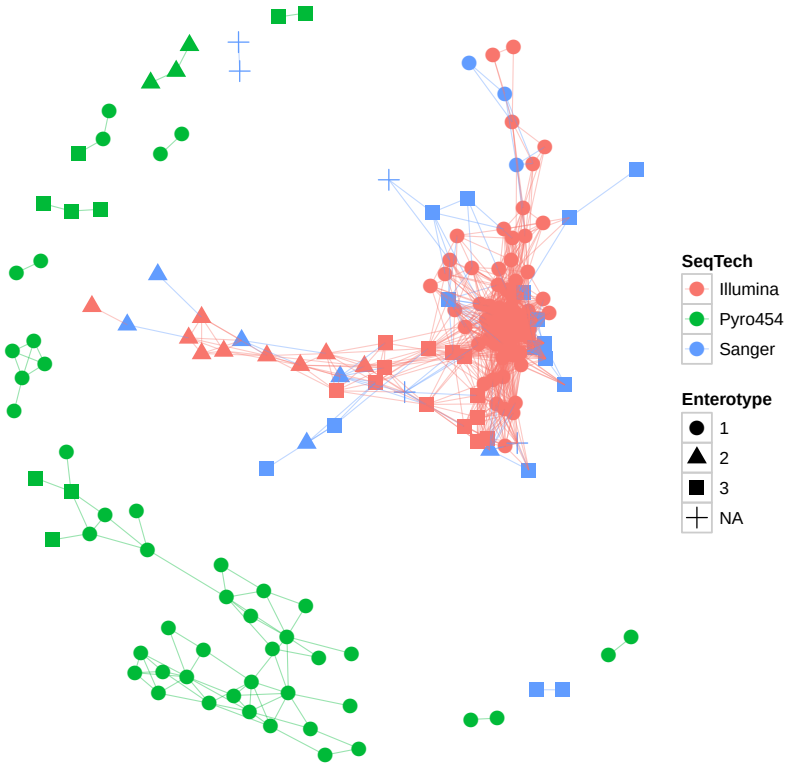


Figure 4: Graphical model of the relationship between microbiome samples in the "Enterotype" dataset [1].

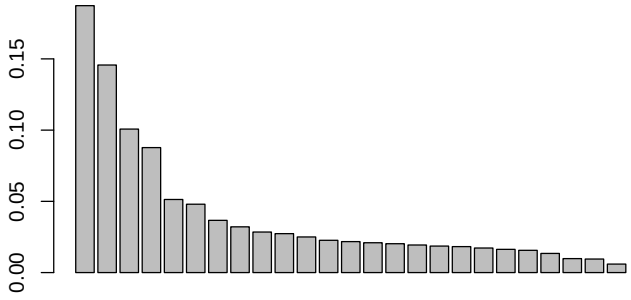


Figure 5: Scree plot of the PCoA used to create Figure 5 from the "Global Patterns" article [2]. The first three axes represent 43% of the total variation in the distances. Interestingly, the fourth axis represents another 9%, and so may warrant exploration as well. A scree plot is an important tool for any ordination method, as the relative importance of axes can vary widely from one dataset to another.

4.2 Ordination Methods

4.2.1 Principal Coordinates Analysis (PCoA)

We take as our first example, a reproduction of Figure 5 from the "Global Patterns" article [2]. The authors show a 3-dimensional representation of the first three axes of a Principal Coordinates Analysis (PCoA) performed on the unweighted-UniFrac distance (see section 4.3.3) using all of the available sequences (their approach included both 5' and 3' sequences). According to the authors, "the first axis [appears to be associated with a] host associated/free living [classification]," and similarly the third axis with "saline/nonsaline environment[s]."

The following reproduces the unweighted UniFrac distance calculation on the full dataset. Note that this calculation can take a long time because of the large number of OTUs. Parallelization is recommended for large datasets, typically if they are as large as `GlobalPatterns`, or larger. For details on parallelization, see the details section and examples in the `UniFrac()` documentation, and also the page dedicated to the topic on the phyloseq-wiki:

`https://github.com/joey711/phyloseq/wiki/Fast-Parallel-UniFrac`

```
> data(GlobalPatterns)

> GPUF <- UniFrac(GlobalPatterns)

> GloPa.pcoa <- pcoa(GPUF)
```

Before we look at the results, let's first investigate how much of the total distance structure we will capture in the first few axes. We can do this graphically with a "scree plot", an ordered barplot of the relative fraction of the total eigenvalues associated with each axis (Fig. 5).

```
> barplot(GloPa.pcoa$values$Relative_eig)
```

Next, we will reproduce Figure 5 from the "Global Patterns" article [2], but separating the three axes into 2 plots using `plot_ordination()` (Fig. 6).

```
> (p12 <- plot_ordination(GlobalPatterns, GloPa.pcoa, "samples", color="SampleType") + geom_line() )
> (p13 <- plot_ordination(GlobalPatterns, GloPa.pcoa, "samples", axes=c(1, 3),
+                                 color="SampleType") + geom_line() )
```

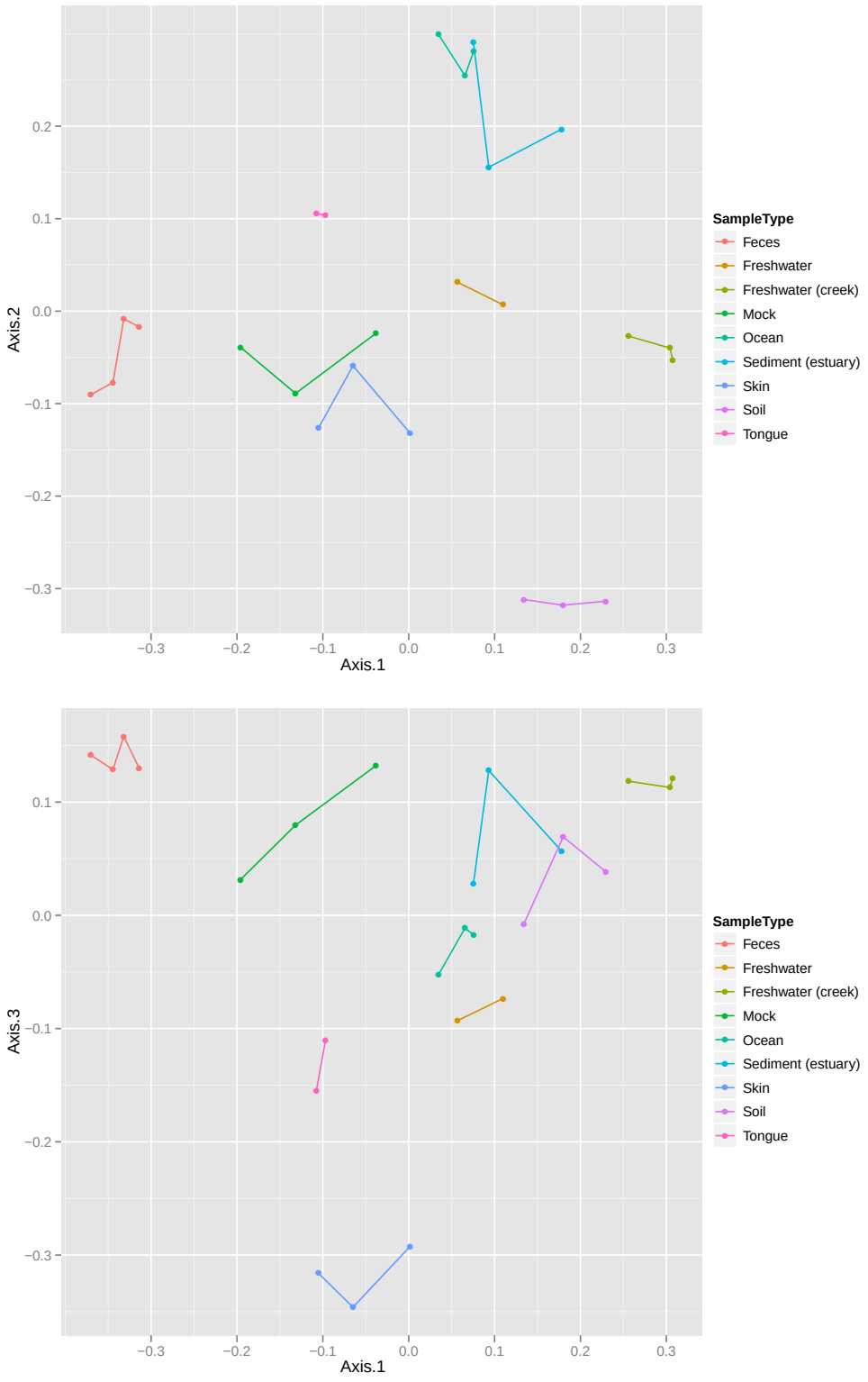


Figure 6: A reproduction in *phyloseq* / `R` of the main panel of Figure 5 from the "Global Patterns" article [2], on two plots. The horizontal axis represents the first axis in the PCoA ordination, while the top and bottom vertical axes represent the second and third axes, respectively. Different points represent different samples within the dataset, and are shaded according to the environment category to which they belong. The color scheme is the default used by *ggplot*.

4.2.2 non-metric Multi-Dimensional Scaling (nmMDS)

We repeat the previous example, but instead using non-metric multidimensional scaling (nmMDS - `metaMDS()`) limited to just two dimensions. This approach limits the amount of residual distance "not shown" in the first two (or three) axes, but forefeits some mathematical properties and does not always converge within the specified number of axes.

```
> # (Re)load UniFrac distance matrix and GlobalPatterns data
> data(GlobalPatterns)
> load("Unweighted_UniFrac.RData") # reloads GPUF variable
> GP.nmMDS <- metaMDS(GPUF, k=2) # perform nmMDS, set to 2 axes

Run 0 stress 0.1432785
Run 1 stress 0.185631
Run 2 stress 0.1670211
Run 3 stress 0.1856305
Run 4 stress 0.1432785
... New best solution
... procrustes: rmse 0.0009642947  max resid 0.003277139
*** Solution reached

> (p <- plot_ordination(GlobalPatterns, GP.nmMDS, "samples", color="SampleType") + geom_line() )
```

Figure 7 nicely shows the relative dissimilarities between microbial communities from different habitats. However, it fails to indicate *what* was different between the communities. For an ordination method that provides information on the taxa that explain differences between samples (or groups of samples), we use Correspondence Analysis (Section 4.2.3).

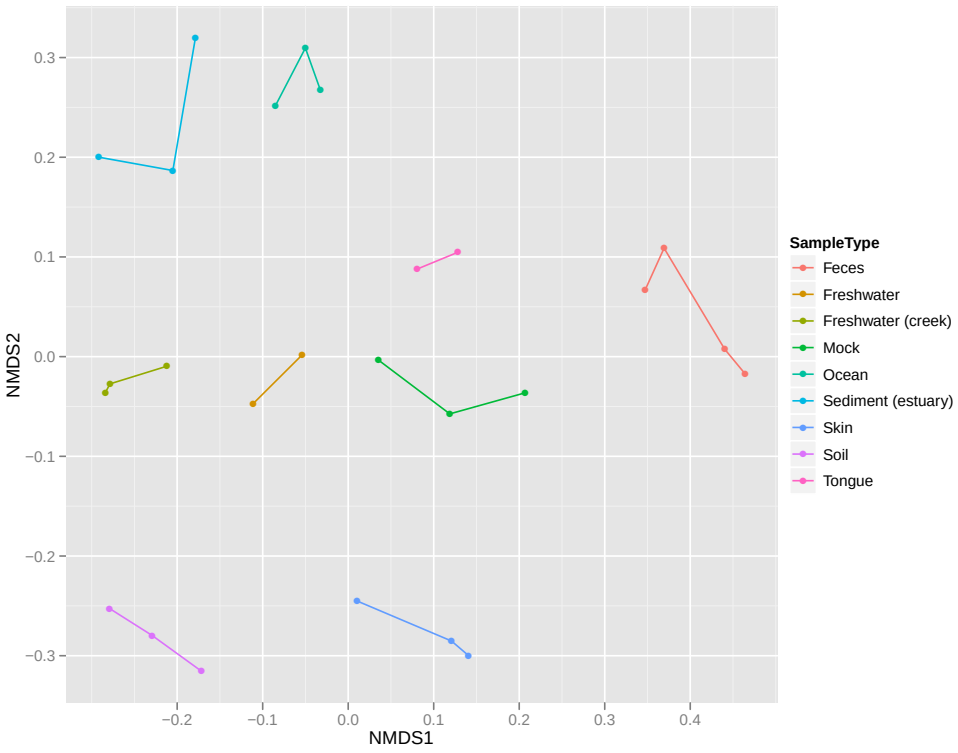


Figure 7: An example exploratory ordination using non-metric multidimensional scaling (nmMDS) on the unweighted UniFrac distance between samples of the "Global Patterns" dataset. Sample points are shaded by environment type, and connected by a line if they belong to the same type. Compare with Figure 5 from the "Global Patterns" article [2].

4.2.3 Correspondence Analysis (CA)

In the following section we will show continue our exploration of the "GlobalPatterns" dataset using various features of an ordination method called Correspondence Analysis. We give special emphasis to exploratory interpretations using the biplot, because it provides additional information that is not available from PCoA or nmMDS.

Let's start by performing a Correspondence Analysis and investigating the scree plot (Figure 8). Both interestingly and challengingly, the scree plot suggests that the `GlobalPatterns` abundance data is quite high-dimensional, with the first two CA axes accounting for not quite 17% of the total (chi-square) variability. Note the absence of a steep decline in eigenvalue fraction as axis number increases. Each additional axis represents only marginally less variability than the previous. It is often more convenient if the first two (or three) axes account for most of the variability.

```
> data(GlobalPatterns)
> # Need to clean the zeros from GlobalPatterns:
> GP <- prune_species(speciesSums(GlobalPatterns)>0, GlobalPatterns)
> # Now do the correspondence analysis
> gpca  <- cca.phyloseq(GP)
> barplot(gpca$CA$eig/sum(gpca$CA$eig), las=2)
```

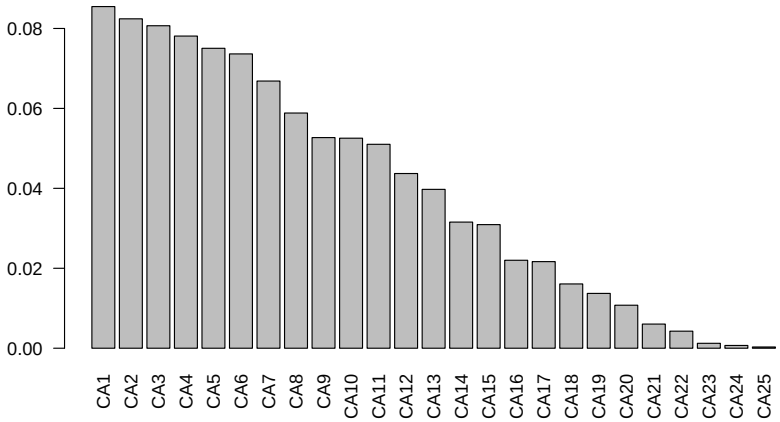
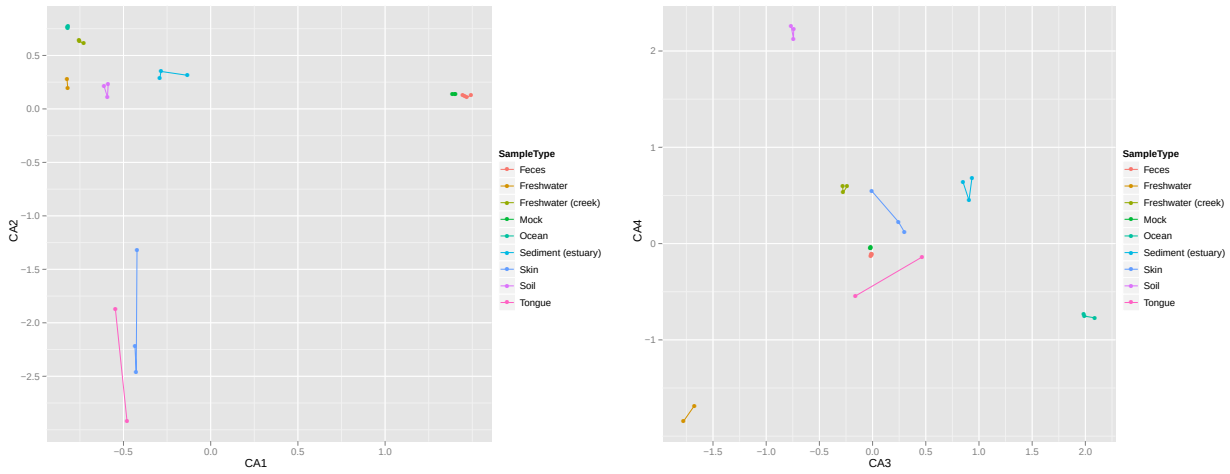


Figure 8: The correspondence analysis (CA) scree plot of the "Global Patterns" dataset [2].

Now let's investigate how the samples behave on the first few CA axes.

```
> (p12 <- plot_ordination(GP, gpca, "samples", color="SampleType") + geom_line() )
> (p34 <- plot_ordination(GP, gpca, "samples", axes=c(3, 4), color="SampleType") + geom_line() )
```

A clear feature of these plots is that the feces and mock communities cluster tightly together, far away from all other samples on the first axis (CA1) in Fig. 9a. The skin and tongue samples separate similarly, but on the second axis. Taken together, it appears that the first two axes are best explained by the separation of human-associated "environments" from the other non-human environments in the dataset, with a secondary separation of tongue and skin samples from feces. Later on we will use this extra categorical designation (human / non-human), so now is a good time to add it as an explicit variable of the 'sampleData':



(a) Axes 1, 2 **(b)** Axes 3, 4

Figure 9: First 4 axes of Correspondence Analysis (CA) of the "Global Patterns" dataset [2].

```
> # Define a human-associated versus non-human categorical variable:
> human.levels <- levels( getVariable(GP, "SampleType") ) %in%
+         c("Feces", "Mock", "Skin", "Tongue")
> human <- human.levels[getVariable(GP, "SampleType")]
> names(human) <- sample.names(GP)
> # Add new human variable to sample data:
> sampleData(GP)$human <- human
```

We will now investigate further this top-level structure of the data, using an additional feature of correspondence analysis that allows us to compare the relative contributions of individual taxa on the same graphical space: the "biplot".

One way to create a biplot in *phyloseq* is to use the included convenience wrapper:

```
> (p <- plot_ordination_biplot(gpca, GP, "", species_color_category=NULL,
+         site_color_category="SampleType", site_shape_category="human") + theme_bw() )
```

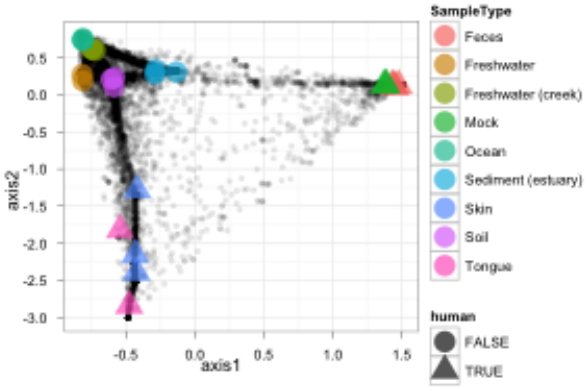


Figure 10: Biplot of the "Global Patterns" CA first two axes, using the dedicated convenience wrapper. The taxa are so diverse (even at the phylum level) that individual shading has been turned off. The darkest black spots are many species-points overlapping one another. In order to keep the file size low, this figure is saved as a low-density raster, but high-quality vector graphics are the default.

While it may be useful to see an overview of how the many thousands of species are ultimately clustering in Fig 10, because there are so many species in this example it is difficult to discern any patterns. For instance, we might want to see how each phylum is represented along the axes. Before showing how to do that, let's first make a custom plot showing just the taxa, while remembering which directions in the coordinate space were important to us (Fig 11)

```
> # Make a data.frame for plotting species, GP CA coordinates
> DF <- data.frame(taxTab(GP), scores(gpca, choices=1:6, display="species"))
> # Sort the phyla names by their total cumulative abundance in the dataset
> top.TaxaGroup <- sort(
+         tapply(speciesSums(GP), taxTab(GP)[, "Phylum"], sum, na.rm = TRUE),
+ decreasing = TRUE)
> # Subset to just those phylum that are among the top-8 most observed in all samples
> DF <- subset(DF, Phylum %in% names(top.TaxaGroup)[1:8])
> # Fix phylum factor issue:
> DF$Phylum <- factor(as(DF$Phylum, "character"))

> (p <- ggplot(DF, aes(x=CA1, y=CA2, color=Phylum)) + geom_point() + theme_bw() )
```

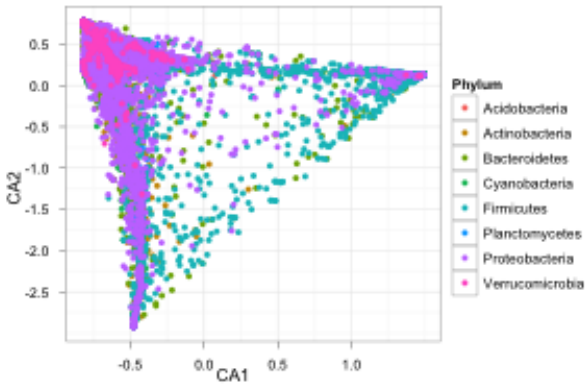


Figure 11: Taxa (species in this case) of the "Global Patterns" CA first two axes, shaded by phylum. Only the top 8 most abundant phyla are included. There are so many individual taxa contributing to this CA that patterns are difficult to discern. In order to keep the file size low, this figure is saved as a low-density raster, but high-quality vector graphics are the default.

Given the large number of points even this phylum-level shading – and only including the most abundant phyla – does not solve the problem of occlusion (Fig 11). This is typical of large datasets. The following is a way to solve the occlusion problem, while still getting useful information about individual species from different phyla that appear to contribute more than others to the separation of human samples from the others. We will need to investigate this further to determine the identities of the outlying species that have a large positive value on `CA1` or negative value on `CA2`.

```
> # Melt the species-data.frame, DF, to facet each CA axis separately
> mdf <- melt(DF[, c("CA1", "CA2", "Phylum", "Family", "Genus")],
+             id=c("Phylum", "Family", "Genus") )
> # Select taxonomic-Family labels of special outliers
> LF <- rbind(
+   subset(mdf, Phylum=="Acidobacteria" & variable=="CA1" & value>1),
+   subset(mdf, Phylum=="Planctomycetes" & variable=="CA1" & value>1),
+   subset(mdf, Phylum=="Cyanobacteria" & variable=="CA2" & value < -2.5)
+   )
> # plot boxplot summaries of each CA-axis, with labels
> p <- ggplot(mdf, aes(Phylum, value, color=Phylum)) + geom_boxplot() +
+   facet_wrap(~variable, 2) + scale_colour_hue(legend = FALSE) +
+   theme_bw() + opts( axis.text.x = theme_text(angle = -90, hjust = 0) )
> # Add the text label layer, and render ggplot graphic
> (p <- p + geom_text(aes(Phylum, value+0.1, color=Phylum, label=Family),
+                     data=LF, vjust=0, size=2) )
```

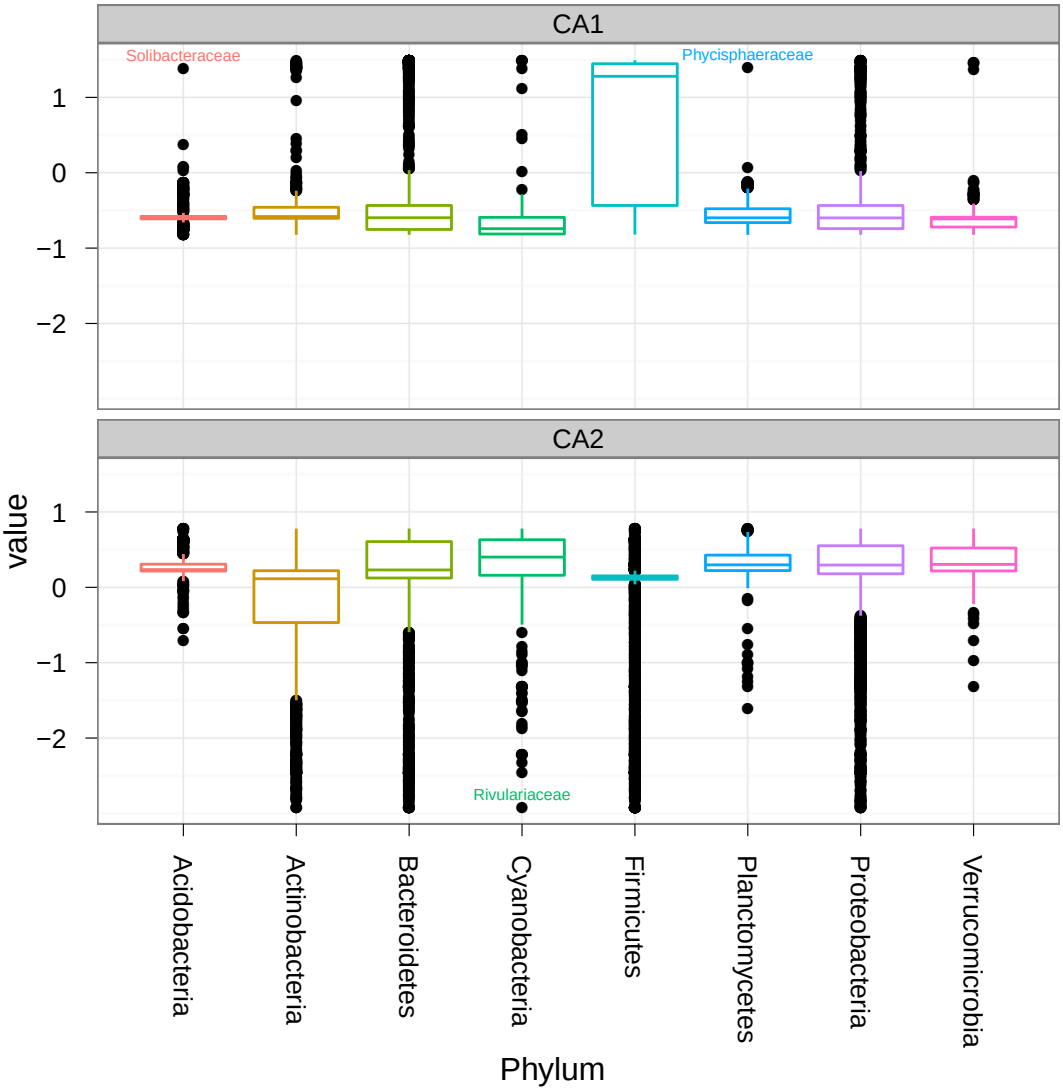


Figure 12: Boxplot of taxa (species in this case) of the "Global Patterns" CA first two axes, shaded by phylum. Only the top 8 most abundant phyla are included. Through this approach it is much easier to see particular species that cluster unusually relative to the rest of their phylum, for example the cyanobacteria (*Rivulariaceae*) that is positioned most in the negative CA2 direction toward the Tongue/Skin samples.

One way to relate some of the high-level patterns we observed from correspondence analysis is to visualize the relative abundances of the relevant phylogenetic groups, to see if this does in fact support / explain the human/environment microbiome differences. Here is an example using the `plot_taxa_bar` function described earlier in Section 3.3.

```
> (p <- plot_taxa_bar(GP, "Phylum", NULL, threshold=0.9, "human", "SampleType",
+                                                    facet_formula= TaxaGroup ~ .) )
```

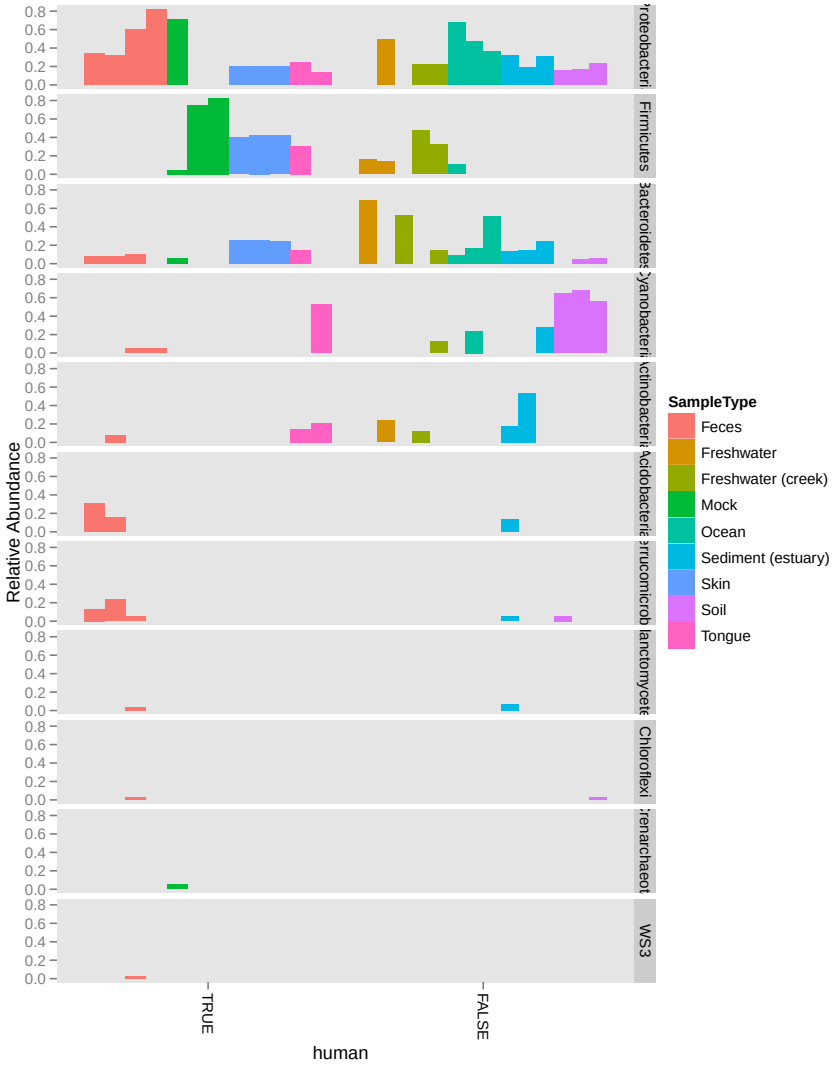


Figure 13: Phylum-level comparison of relative abundance of taxa in samples that are from human microbiomes (or not).

In Fig 13 we've used the `threshold` parameter to omit all but phyla accounting for the top 90% of phyla in any one sample. Some patterns emerging from this display appear to be: (1) Cyanobacteria, Actinobacteria appear under-represented in human samples; (2) conversely, Firmicutes appear over-represented in human samples; (3) Acidobacteria, Verrucomicrobia appear over-represented in the fecal samples; (4) the only Crenarchaeota were observed in the Mock sample, which is not really a community but a simulated

community used as a control. These are not hugely surprising based on previous biological observations from the field, but it is hopefully useful code that can be applied on other datasets that you might have.

4.2.4 Double Principle Coordinate Analysis (DPCoA)

`DPCoA()`

4.3 Distance Methods

4.3.1 `distance()`: Central Distance Function

Many comparisons of microbiome samples, including the graphical model (Section 4.1) and the PCoA analysis (Section 4.2.1), require a calculation for the relative dissimilarity of one microbial community to another, or "distance". Although not fully implemented yet, the phyloseq-package intends to provide a unified ecological distance function for calculating a matrix of microbial community distances between the samples in an experiment. This will surely include a wrapper for the several dozen distance calculations provided via the three distance functions in the `vegan`-package, many of the distance methods supported in the `ade4`-package, as well as the included `UniFrac` distance function (Section 4.3.3), a method for calculating Double Principal Coordinate Analysis (DPCoA), as well as an extension to the vegan interface for arbitrary, user-defined distances.

The function will take a `phyloseq-class` object and an argument indicating the distance type; and it will return a `dist`-class distance matrix.

4.3.2 `vegdist()` extension

The *phyloseq* package includes an extension for the `vegdist()` function from the *vegan* package [3], which in-turn can calculate 14 or so ecologically relevant distances / dissimilarity indices. The primary argument should be a `phyloseq-class` object, and the expected result is a sample-wise distance matrix.

```
> data(esophagus)
> vegdist(esophagus)
```

```
     B    C
C 0.406
D 0.498 0.591
```

The available distances/dissimilarity indices calculated by `vegdist()` currently include the following:

```
 [1] "manhattan"  "euclidean"  "canberra"   "bray"       "kulczynski"
 [6] "jaccard"    "gower"      "altGower"   "morisita"   "horn"
[11] "mountford"  "raup"       "binomial"   "chao"
```

These are specified by the `method=` argument. For example, if one alternatively wants to calculate the Jaccard distance instead of the default (Bray-Curtis), the following command will work:

```
> data(esophagus)
> vegdist(esophagus, "jaccard")
```

```
     B    C
C 0.578
D 0.665 0.743
```

We also have plans to extend the `designdist`, `betadiver`, and `dist` functions, which provide even further options for distance type. `designdist` also provides a way to define a custom distance calculation, while `betadiver` calculates all 24 ecological distances reviewed in Koleff et al. 2003 [4]. We plan to wrap all of these methods into one *central* distance calculator method, say "`sampleDistance()`", that would also include UniFrac, DPCoA, etc. This will be implemented soon.

4.3.3 UniFrac and weighted UniFrac

UniFrac is a recently-defined [5] and popular distance metric to summarize the difference between pairs of ecological communities. All UniFrac variants use a phylogenetic tree of the relationship among taxa as central information to calculating the distance between two samples/communities. An unweighted UniFrac distance matrix only considers the presence/absence of taxa, while weighted UniFrac accounts for the relative

abundance of taxa as well as their phylogenetic distance. Prior to *phyloseq*, a non-parallelized, non-Fast implementation of the unweighted UniFrac was available in R packages (`picante::unifrac` [6]). In the *phyloseq* package we provide optionally-parallelized implementations of Fast UniFrac [7] (both weighted and unweighted, with plans for additional UniFrac variants), all of which return a sample-wise distance matrix from any `phyloseq-class` object that contains a phylogenetic tree component.

The following is an example calculating the UniFrac distance (both weighted and unweighted) matrix using the "esophagus" example dataset:

```
> data(esophagus)
> UniFrac(esophagus, weighted=TRUE)
> UniFrac(esophagus, weighted=FALSE)
```

```
      B     C
C 0.204
D 0.260 0.248
```

```
      B     C
C 0.518
D 0.518 0.542
```

See the wiki-page devoted to details about calculating the UniFrac distances for your experiment. In particular, some example run-times are provided for comparison, as well as details for initializing a parallel "back end" to perform the computation with multiple processor cores simultaneously:

`https://github.com/joey711/phyloseq/wiki/Fast-Parallel-UniFrac`

4.4 Hierarchical Clustering

Another potentially useful and popular way to visualize/decompose sample-distance matrices is through hierarchical clustering (e.g. `hclust()`). In the following example, we reproduce Figure 4 from the "Global Patterns" article [2], using the unweighted UniFrac distance and the UPGMA method (`hclust` parameter `method="average"`). Try `help("hclust")` for alternative clustering methods included in standard `R`.

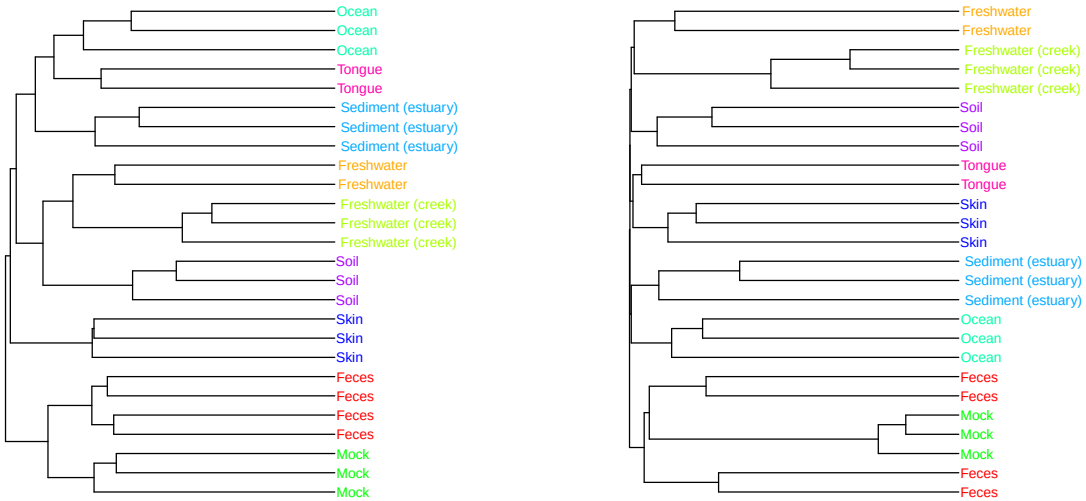
```
> # (Re)load UniFrac distance matrix and GlobalPatterns data
> data(GlobalPatterns)
> load("Unweighted_UniFrac.RData") # reloads GPUF variable
> # Manually define color-shading vector based on sample type.
> colorScale    <- rainbow(length(levels(getVariable(GlobalPatterns, "SampleType"))))
> cols          <- colorScale[getVariable(GlobalPatterns, "SampleType")]
> GP.tip.labels <- as(getVariable(GlobalPatterns, "SampleType"), "character")
> GP.hclust     <- hclust(GPUF, method="average")
```

Plot the hierarchical clustering results as a dendrogram, after first converting the `hclust-class` object to `phylo-class` tree using the `as.phylo()` function.

```
> plot(as.phylo(GP.hclust), show.tip.label=TRUE, tip.color="white")
> tiplabels(GP.tip.labels, col=cols, frame="none", adj=-0.05, cex=0.7)
```

Create an alternative plot, using the Jaccard distance and complete-linkage clustering (the default) instead of UPGMA.

```
> jaccCLC <- hclust(vegdist(GlobalPatterns, "jaccard"))
> plot( as.phylo(jaccCLC), show.tip.label=TRUE, tip.color="white" )
> tiplabels(GP.tip.labels, col=cols, frame="none", adj=-0.05, cex=0.7)
```



(a) UniFrac / UPGMA

(b) Jaccard / CLC

Figure 14: An alternative means of summarizing a distance matrix via hierarchical clustering and plotting as an annotated dendrogram. Compare with Figure 4 from the "Global Patterns" article [2]. Panel 14a represents a faithful reproduction of the original approach from the article using `R` utilities, while Panel 14b is an illustration of slightly different results with different choices of distance measure and clustering algorithm. Some differences in Panel 14a from the original article might be explained by the `GlobalPatterns` dataset in *phyloseq* includes the summed observations from both directions (5' and 3'), while in the article they show the results separately. Furthermore, in the article the "mock" community is not included in the dataset, but an extra fecal sample is included.

5 Validation

5.1 Multiple Inference Correction

The *phyloseq* package includes support for significance testing with correction for multiple inference. This is particularly important when testing for significance of the abundance patterns among thousands of microbes (OTUs). This is a common question of phylogenetic sequence data, that is, "what is the subset of microbes that significantly correlate with a scientifically-interesting sample variable". Although we plan to include support for other types of multiple-inference corrected tests, this "which taxa?" test is the only directly supported test at the moment.

Our initial implementation of this support is via an extension to the `mt.maxT()` and `mt.minP` functions in the *multtest* package [8] (Bioconductor repo). This uses permutation-adjusted p-values in a multiple testing procedure that provides strong control of the Family-Wise Error Rate (FWER) among the taxa being tested. The user specifies a sample-variable among the `sampleData` component, or alternatively provides a sample-wise vector or factor that classifies the samples into groups. Additional optional parameters can be provided that specify the type of test ("`test=`"), the sidedness of the test ("`side=`"), as well as some additional technical/computational parameters.

In the following example we test whether a particular genera correlates with the Enterotype classification of each sample. Note that we have to specify an alternate test, `test="f"`, because the default test (t-test) can only handle up to 2 classes, and there are three enterotype classes.

```
> data(enterotype)
> # Filter samples that don't have Enterotype classification.
> x <- subset_samples(enterotype, !is.na(Enterotype))
> # Calculate the multiple-inference-adjusted P-values
> ent.p.table <- mt(x, "Enterotype", test="f")
> print(head(ent.p.table, 10))
```

genera	index	teststat	rawp	adjp	plower
Prevotella	207	344.73	0.0001	0.0158	0.0001
Bacteroides	203	85.01	0.0001	0.0158	0.0001
Blautia	187	19.52	0.0001	0.0158	0.0001
Bryantella	503	16.38	0.0001	0.0158	0.0001
Parabacteroides	205	12.89	0.0001	0.0158	0.0001
Alistipes	208	8.71	0.0002	0.0301	0.0158
Bifidobacterium	240	9.29	0.0004	0.0560	0.0430
Holdemania	201	7.64	0.0009	0.1146	0.1031
Dorea	182	7.44	0.0009	0.1146	0.1031
Phascolarctobacterium	513	7.01	0.0014	0.1695	0.1585

Table 1: For computational efficiency this calculation was run separately, and results embedded here.

Not surprisingly, *Prevotella* and *Bacteroides* top the list, since they were major components of the "Enterotype" classification.

Please also note that we are planning to incorporate other tools from *multtest* that would allow for other types of multiple-inference correction procedures, for instance, strong control of the False Discovery Rate (FDR). These additional options will be made available shortly.

6 Further Examples

This vignette is limited in size and scope because of constraints on the cumulative-size of packages allowed in Bioconductor.

6.1 phyloseq Wiki

For further examples presented in html/wiki format, please see:
`https://github.com/joey711/phyloseq/wiki/Graphics-Examples`

6.2 phyloseq Vignette Gallery

For additional and/or updated vignettes not present in the official Bioconductor release, please see:
`https://github.com/joey711/phyloseq/wiki/Vignettes`

6.3 phyloseq Feedback

For feature requests, bug reports, and other suggestions and issues, please go to:
`https://github.com/joey711/phyloseq/issues`

References

[1] Arumugam, M. *et al.* Enterotypes of the human gut microbiome. *Nature* **473**, 174–180 (2011). URL http://dx.doi.org/10.1038/nature09944.

[2] Caporaso, J. G. *et al.* Global patterns of 16s rrna diversity at a depth of millions of sequences per sample. *Proceedings of the National Academy of Sciences* **108**, 4516–4522 (2011). URL http://www.pnas.org/content/108/suppl.1/4516.abstract. http://www.pnas.org/content/108/suppl.1/4516.full.pdf+html.

[3] Oksanen, J. *et al. vegan: Community Ecology Package* (2011). URL http://CRAN.R-project.org/package=vegan. R package version 1.17-10.

[4] Koleff, P., Gaston, K. J. & Lennon, J. J. Measuring beta diversity for presence–absence data. *Journal of Animal Ecology* **72**, 367–382 (2003). URL http://dx.doi.org/10.1046/j.1365-2656.2003.00710.x.

[5] Lozupone, C. & Knight, R. UniFrac: a new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology* **71**, 8228–8235 (2005).

[6] Kembel, S. W. *et al.* Picante: R tools for integrating phylogenies and ecology. *Bioinformatics (Oxford, England)* **26**, 1463–1464 (2010).

[7] Hamady, M., Lozupone, C. & Knight, R. Fast unifrac: facilitating high-throughput phylogenetic analyses of microbial communities including analysis of pyrosequencing and phylochip data. *ISME J* **4**, 17–27 (2009). URL http://dx.doi.org/10.1038/ismej.2009.97.

[8] Pollard, K. S., Gilbert, H. N., Ge, Y., Taylor, S. & Dudoit, S. *multtest: Resampling-based multiple hypothesis testing.* R package version 2.10.0.