microbiomeSeq: An R package for microbial community analysis in an environmental context

*Alfred Ssekagiri, William T. Sloan, * Umer Zeeshan Ijaz (* Correspondence: Umer.Ijaz@glasgow.ac.uk)*

August 20, 2017

Contents

Introduction to microbiomeSeq

This package is developed to enhance the available statistical analysis procedures in R and providing more informative visualisation of analysis results for microbial communities obtained from 16S ribosomal RNA gene sequencing studies. It has been designed under the supervision of Dr. Umer Zeeshan Ijaz (primary) and Prof. William T. Sloan (secondary) by Alfred Ssekagiri as part of his Msc Bioinformatics project. Source code is available at http://www.github.com/umerijaz/microbiomeSeq. Here we present a tutorial with minimum working examples to demonstrate usage and dependencies.

Disclaimer:

microbiomeSeq is still in development phase and **we do not recommend you** to use it until a stable version is available and when this message disappears.

Installation

Install the package with its dependencies and load it for usage in R.

```
library(devtools) # Load the devtools package
install_github("umerijaz/microbiomeSeq") # Install the package
library(microbiomeSeq)  #load the package
```

Data format/requirement

The data is required to be a phyloseq object (`phloseq-class`) comprising taxa abundance information, taxonomy assignment, sample data which is a combination of measured environmental variables together with any categorical variables present in the samples. If the phylogenetic tree is available, it can also be part but not so relevant for most of the functionality implemented here so far. We choose to use this format since we can have enormous options for manipulating the data as we progress with the analysis and visualisations. Details of format and comprehensive manipulations of phyloseq objects are available at https://github.com/joey711/phyloseq.

Example data

To test the functionality, we use a pitlatrine dataset which was generated by 16S rRNA sequencing of various latrines from Tanzania and Vietnam at different depths. The data files are available at http://userweb.eng.gla.ac.uk/umer.ijaz/bioinformatics/ecological.html and the associated paper is B Torondel, JHJ Ensink, O Gundogdu, UZ Ijaz, J Parkhill, F Abdelahi, V-A Nguyen, S Sudgen, W Gibson, AW Walker, and C Quince. **Assessment of the influence of intrinsic environmental and geographical factors on the bacterial ecology of pit latrines** Microbial Biotechnology, 9(2):209-223, 2016. It is also freely accessible here.

To get the test data in phyloseq format,

```
library(microbiomeSeq)
data(pitlatrine)
```

To check the components of the data, print out the data to find out the structure.

```
print(pitlatrine)
phyloseq-class experiment-level object
otu_table()   OTU Table:         [ 8883 taxa and 81 samples ]
sample_data() Sample Data:       [ 81 samples by 14 sample variables ]
tax_table()   Taxonomy Table:    [ 8883 taxa by 6 taxonomic ranks ]
phy_tree()    Phylogenetic Tree: [ 8883 tips and 8881 internal nodes ]
```

Generate a phyloseq object

To generate a phyloseq object to be used for analysis, a phyloseq function `merge_phyloseq` can be used to combine the taxa abundance information (`OTU`), taxa assignment (`TAX`), sample data (`SAM`) and phylogenetic tree (`OTU_tree`) in Newick format as follows; More details on how to construct a phyloseq object can be obtained from the phyloseq site cited earlier.

```
OTU = otu_table(as.matrix(abund_table), taxa_are_rows = FALSE)
TAX = tax_table(as.matrix(OTU_taxonomy))
SAM = sample_data(meta_table)
OTU_tree<-compute.brlen(OTU_tree,method="Grafen")
physeq<-merge_phyloseq(phyloseq(OTU, TAX),SAM,OTU_tree)
```

Data normalisation

Microbial community data is mainly OTU/taxa abundance (counts) and corresponding environmental data. The statistical methods have different requirements regarding the distribution and kind of data (for example counts, binary, fractional e.t.c), therefore, it is usually necessary that data is transformed by a suitable normalisation method.

Normalising OTU abundance

We implement different methods including; "relative", "log-relative", random sub sampling ("randomsubsample"), edgeR ("edgernorm") and variance stabilisation ("varstab") for normalisation of taxa abundance. The function takes a phyloseq object `physeq` and returns a similar object whose `otu-table` component is normalised by a selected method as shown in the following examples.

```
physeq<-normalise_data(physeq,norm.method = "randomsubsample")
physeq <- normalise_data(physeq, norm.method = "varstab" ,fitType="local")
```

Normalising sample data

In order to transform the `sample_data` component of `phyloseq` object, a logical value `norm.meta` is set to `TRUE` in additon to a suitable normalisation method. Note that amongst the above mentioned methods, this option(`norm.meta`) is currently available for relative and log-relative only.

```
physeq <- normalise_data(physeq, norm.method = "relative", norm.meta=T)
```

To scale `sample data`, "scale" is the selected as the `norm.method`. This function can also be used to perform log2 and square root transformation of `sample data` which is specified using the `type` argument as illustrated in the example below.

```
physeq <- normalise_data(physeq, norm.method = "scale", type="log")
physeq <- normalise_data(physeq, norm.method = "scale", type="sqrt")
```

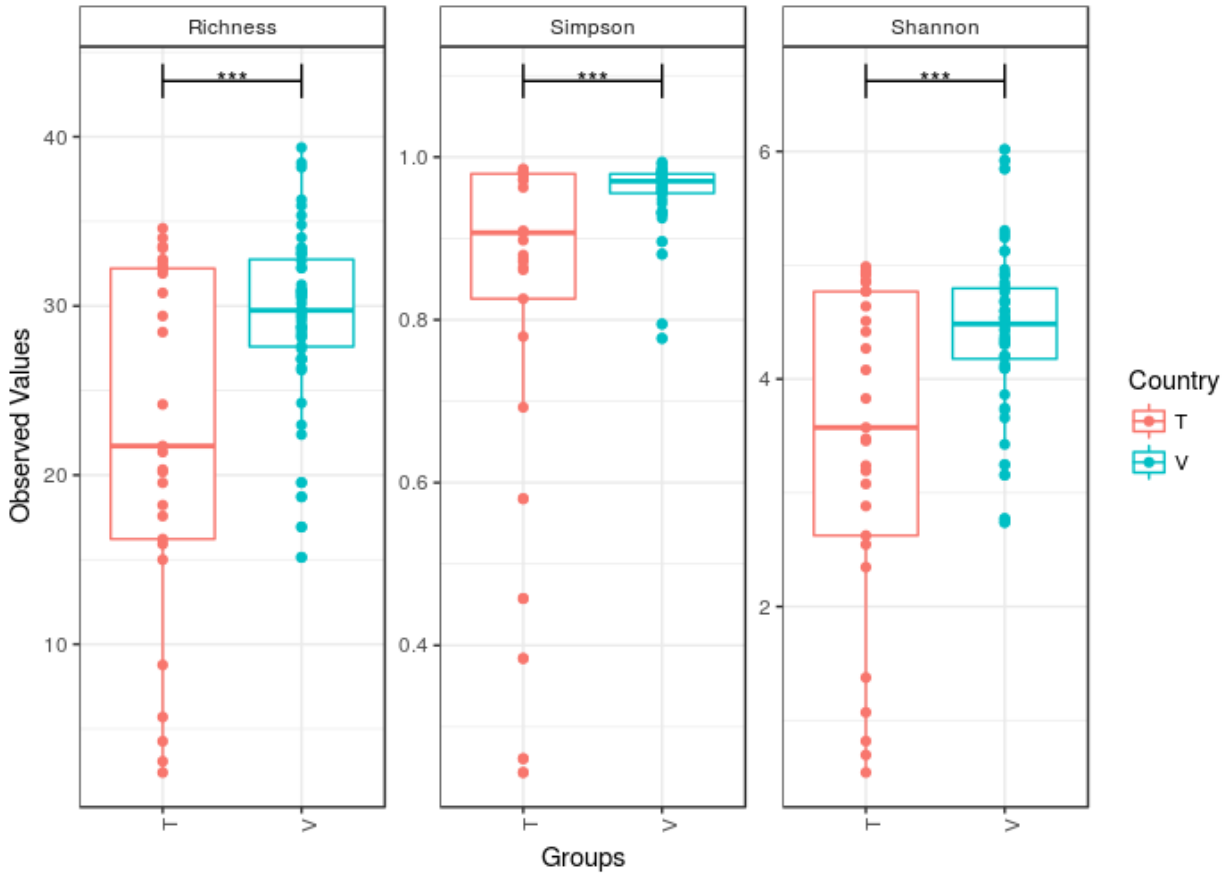
Alpha diversity with ANOVA

This function calculates alpha diversity of provided community data using selected indices/method(s). It performs pair-wise ANOVA of diversity measures between groups and outputs a plot for each of the selected methods(indices) annotated with significance labels.

`method`, options include:"richness", "fisher", "simpson", "shannon" and "evenness". It performs pairwise analysis of variance in diversity between groups and its significance annotated as on the plots. `grouping_column` is a categorical variable for which the grouping should be based on during the analysis. `pValueCutoff` specifies the p-value threshold for significance in `ANOVA`, default is set to 0.05. For the following examples, we use simpson, richness and shannon indices for calculating diversity.

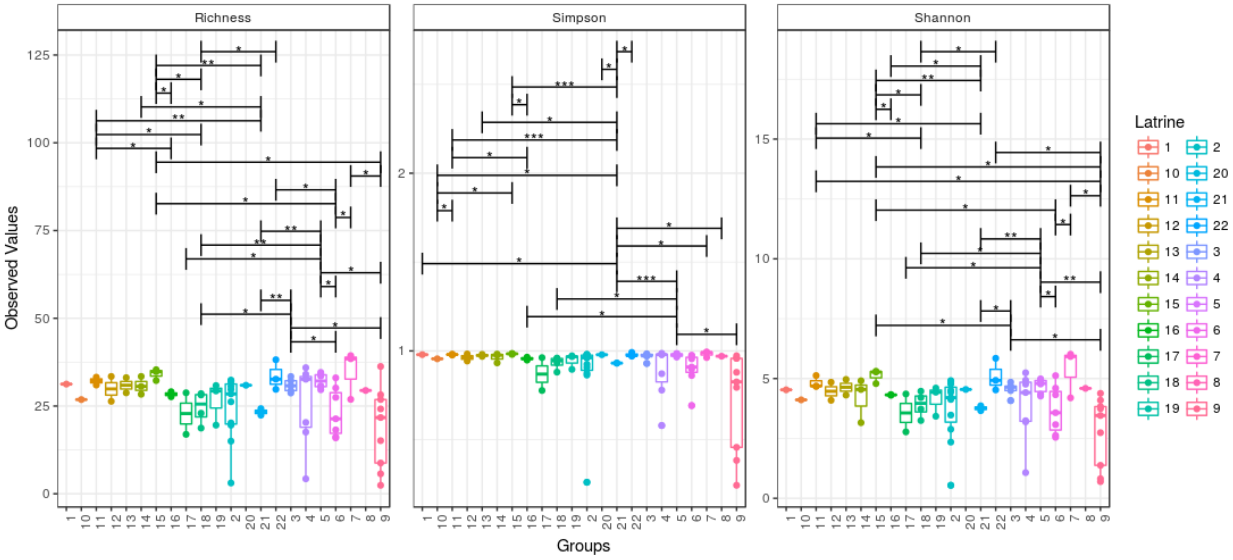
Grouping by Country categorical variable, we obtain the following plot.

```
p<-plot_anova_diversity(physeq, method = c("richness","simpson", "shannon"),
                        grouping_column =  "Country",pValueCutoff=0.05)
print(p)
```



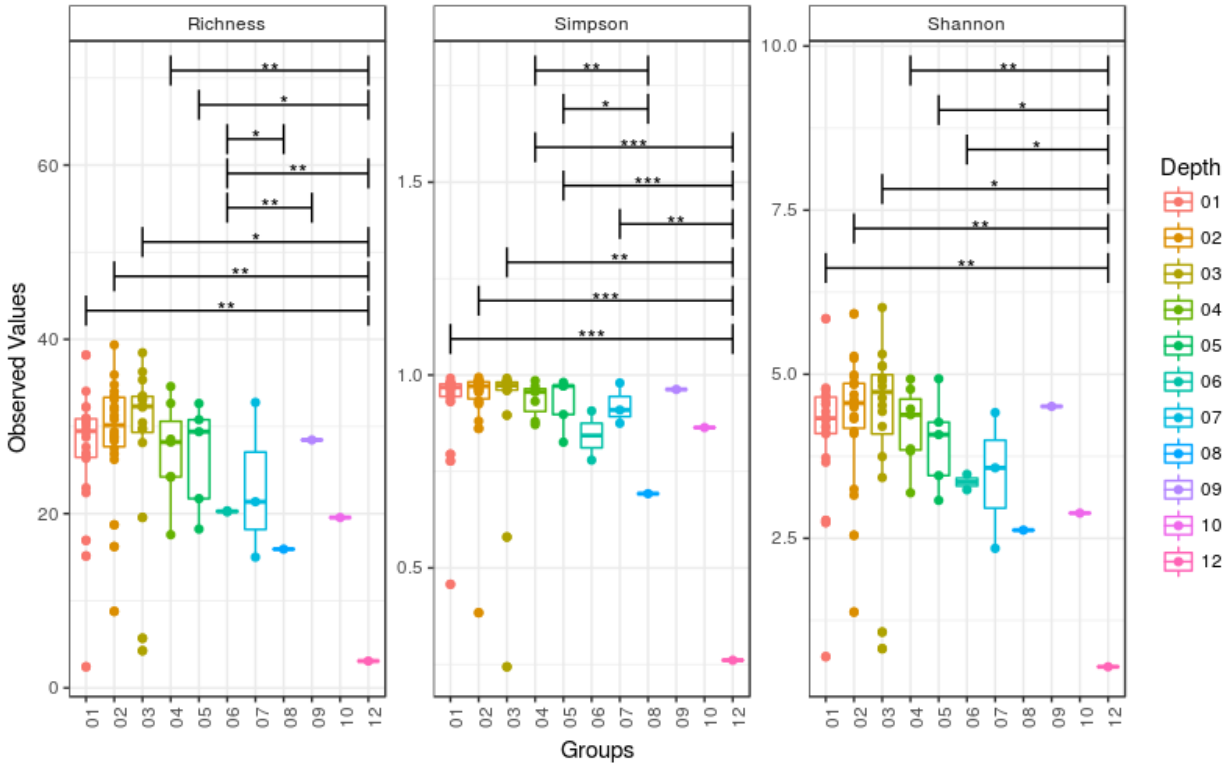
Grouping by Latrine categorical variable, we obtain the following plot.

```
p<-plot_anova_diversity(physeq, method = c("richness","simpson", "shannon"),
                        grouping_column =  "Latrine", pValueCutoff=0.05)
print(p)
```



Grouping by Depth categorical variable, we obtain the following plot.

```
p<-plot_anova_diversity(physeq, method = c("richness","simpson", "shannon"),
                        grouping_column =  "Depth", pValueCutoff=0.05)
print(p)
```



Beta diversity

Local Contribution to Beta diversity

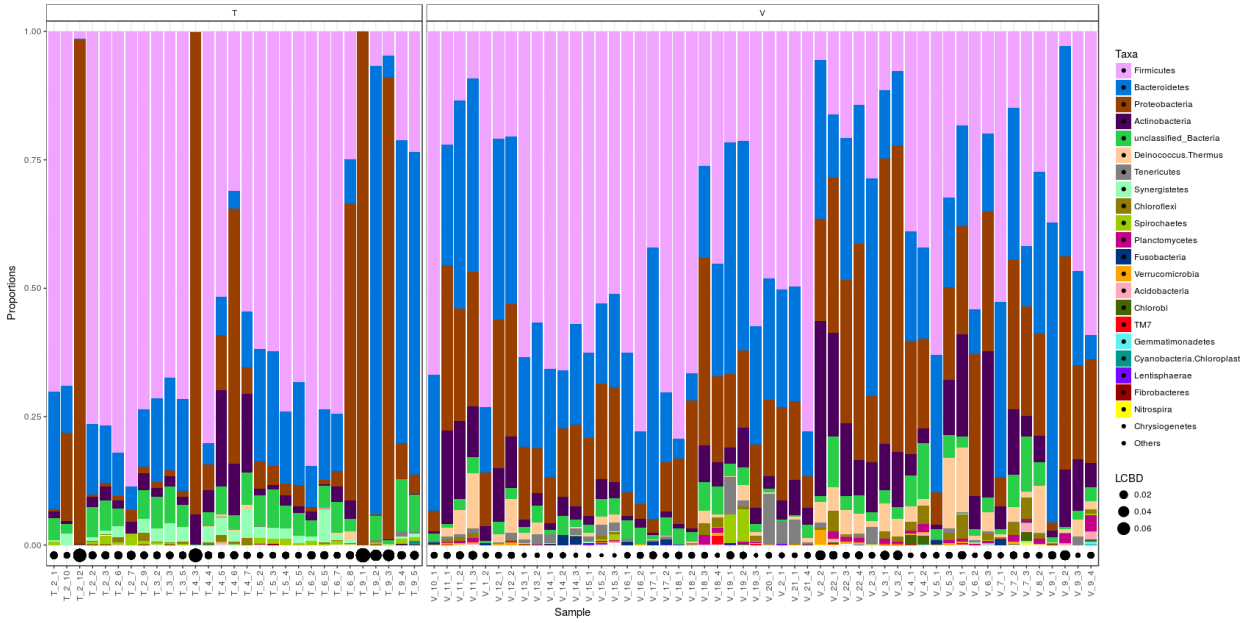
To measure degree of uniqueness of a given sample to the variation in community composition, LCBD is calculated according to the procedure provided by (Legendre and Cáceres 2013). In the example provided below, we relative normalised taxa abundance to obtain the proportion of most abundant taxa per sample. This shows the features which are responsible for observed values of LCBD for a given sample.

The plot produced has points at the bottom whose diameter corresponds to magnitude of LCBD value coresponding to a particular sample, the bars correspond to taxa that are most abundant with the top taxa sharing a bigger portion of the bar for each sample.

A character string for the variable to be used for grouping is be specified as `grouping_column`. Dissimilarity coefficients method to be used specified as a character string `method`, default is set to "hellinger", other options for this include: "chord", "chisquare","profiles", "percentdiff", "ruzicka", "divergence", "canberra", "whittaker", "wishart", "kulczynski", "jaccard", "sorensen","ochiai", "ab.jaccard", "ab.sorensen","ab.ochiai", "ab.simpson" and "euclidean". Supplying a filename writes a file containing values for local contribution to beta diversity, corresponding p-value for each sample.

```
physeq <- normalise_data(physeq, norm.method = "relative")
```

```
p <- plot_taxa(physeq,grouping_column="Country",method="hellinger",number.taxa=21,filename=NULL)
print(p)
```



A file containing details of local contribution to beta diversity can be generated by setting supplying a filename. It contains LCBD values, associated p-values and group for each of the samples.

```
      Sample        LCBD p.LCBD Country
T_2_1   T_2_1 0.011716826  0.499       T
T_2_2   T_2_2 0.012600929  0.431       T
T_2_3   T_2_3 0.012908548  0.454       T
T_2_6   T_2_6 0.013977118  0.389       T
T_2_7   T_2_7 0.017756062  0.211       T
```

```
V_3_1   V_3_1 0.018815943  0.167       V
V_3_2   V_3_2 0.021974763  0.097       V
V_4_1   V_4_1 0.003868666  0.937       V
V_4_2   V_4_2 0.005543190  0.836       V
V_5_1   V_5_1 0.005736017  0.833       V
V_5_3   V_5_3 0.008149469  0.680       V
V_6_1   V_6_1 0.015033956  0.276       V
```

Ordination and beta dispersion

Ordination: This is the clustering procedure of samples to detect features that are more like each other in the dataset. We implement Non-metric multidimensional Scaling (NMDS) which is a rank based approach and PCoA also known as metric/classical multidimensional scaling which uses simmilarity or dissimilarity measure to group samples and provide a representation of original dataset in a lower dimension.

Beta-dispersion: This measures variances in abundance for a group of samples by computing average distance of individual groups to the group centroid, these distances are subjected to ANOVA to test whether they are different or not.The most significantly dispersed groups are annotated on the plot with corresponding significance labels.

We also implement permutation analysis of variance (PERMANOVA) and corresponding r-squared and p-values are anotated on NMDS plot, beta dispersion between all posible pairwise combinations of levels in the grouping variable is calculated and results presented as desired by using provided parameters.

The arguments include: `physeq` which a required phyloseq object, `distance` which is a dissimilarity distance measure with otions of "bray" (default), "wunifrac"and "unifrac". `grouping_column` is a character string specifying a variable whose levels are the groups in the data. `pvalue.cutoff` is threshold p-value for beta dispersion significance (default is 0.05). `show.pvalues` a logical variable for whether to show p-values in beta dispersion results or not, setting it to `FALSE` shows only the significance labels.`num.signi.groups` (optional): An integer for the number of signicant beta dispersion results to report, this is could be necessary in case of `grouping_column` variables has many levels to avoid overcrowding the plot area.`method` is a character string for ordination method, "NMDS" is the only available method so far.

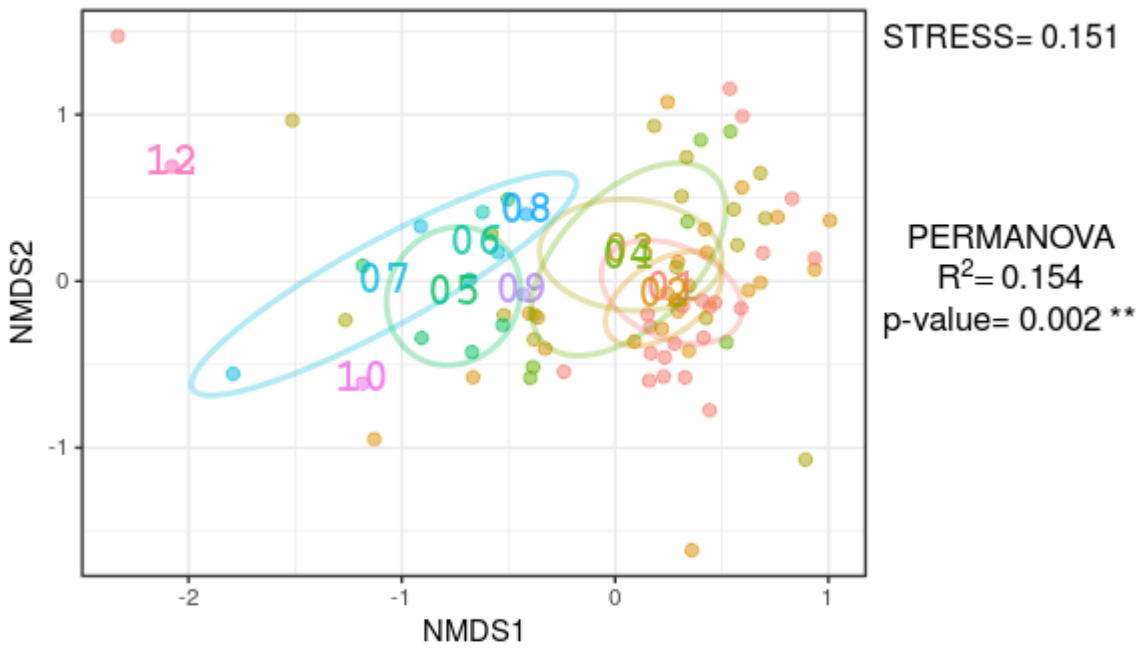
To produce ordination of the data;

```
ord.res <- ordination(physeq,distance="bray",method="NMDS",grouping_column="Depth",
                      pvalue.cutoff=0.05)
```

To plot the ordination results, `plot_ordination` function is used. It takes result of function `ordination`.

```
p <- plot_ordination(ord.res , method="NMDS", pvalue.cutoff=0.05,
                     show.pvalues=T, num.signi.groups=NULL)
print(p)
```

An example of a plot produced by NMDS ordination method with a Depth as grouping variable.



BETA-DISPERSION

06 Vs 08 p-value=1e-16 ***

06 Vs 10 p-value=1e-16 ***

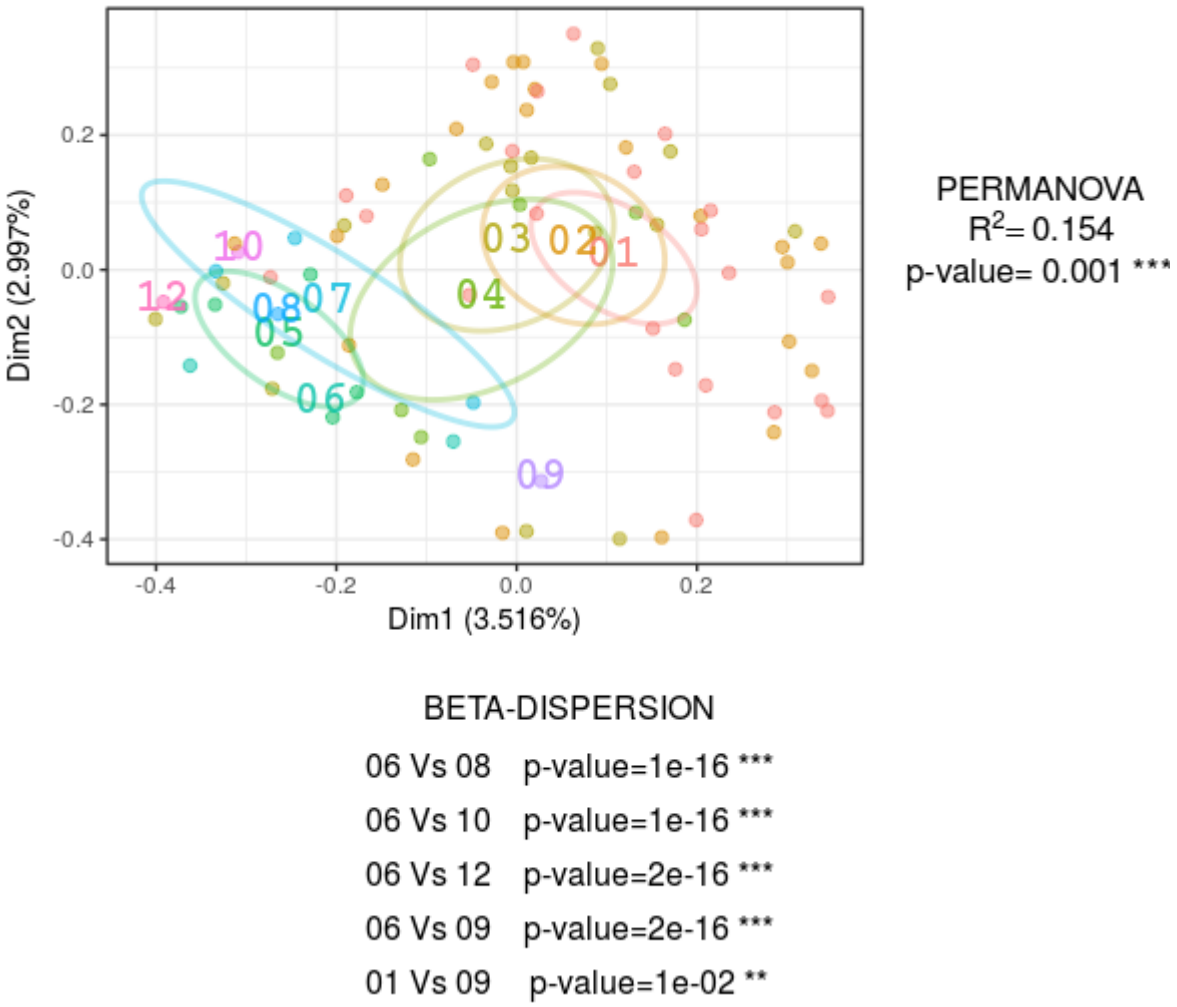
06 Vs 12 p-value=2e-16 ***

06 Vs 09 p-value=2e-16 ***

01 Vs 09 p-value=1e-02 **

Selecting PCoA as the ordination method reports the variance in original dataset explained by the first and second dimensions on the axes labels as percentages.

```
p <- plot_ordination(ord.res, method="PCoA" ,pvalue.cutoff=0.05,
                     show.pvalues=T,num.signi.groups=NULL)
print(p)
```

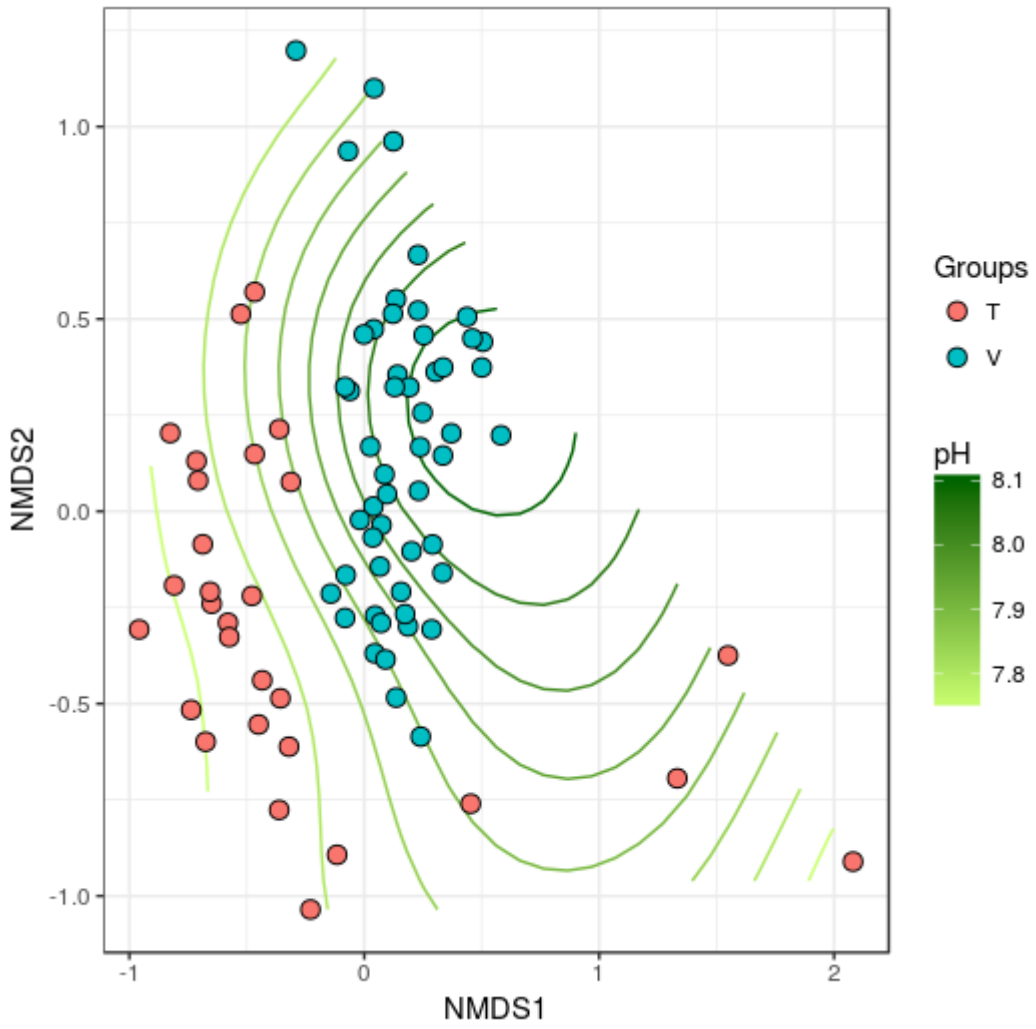


Ordisurf

This function plots a surface of specified variable on ordination plot using `ordisurf` function. It takes ordination results from `ordination`, a `data.frame` of environmental variables and character string for name of environmental variable whose values should be used to generate the surface.

In this example, contours show the distribution of pH in the samples on the NMDS plot.

```
p <- plot_ordisurf(ord.res, meta_table, variable = "pH")
print(p)
```



Fuzzy set ordination of environmental variables

Fuzzy set ordination is used to test effects of pertubation in environmental avariables to community structure. For each of the specified variables, a fuzzy set ordination is calculated and the correlation between the original variable and the fuzzy set is reported. The significance of a particular variable is assessed by comparing a specified threshold p-value and the probability of obtaining a correlation between the data and fuzzy set.

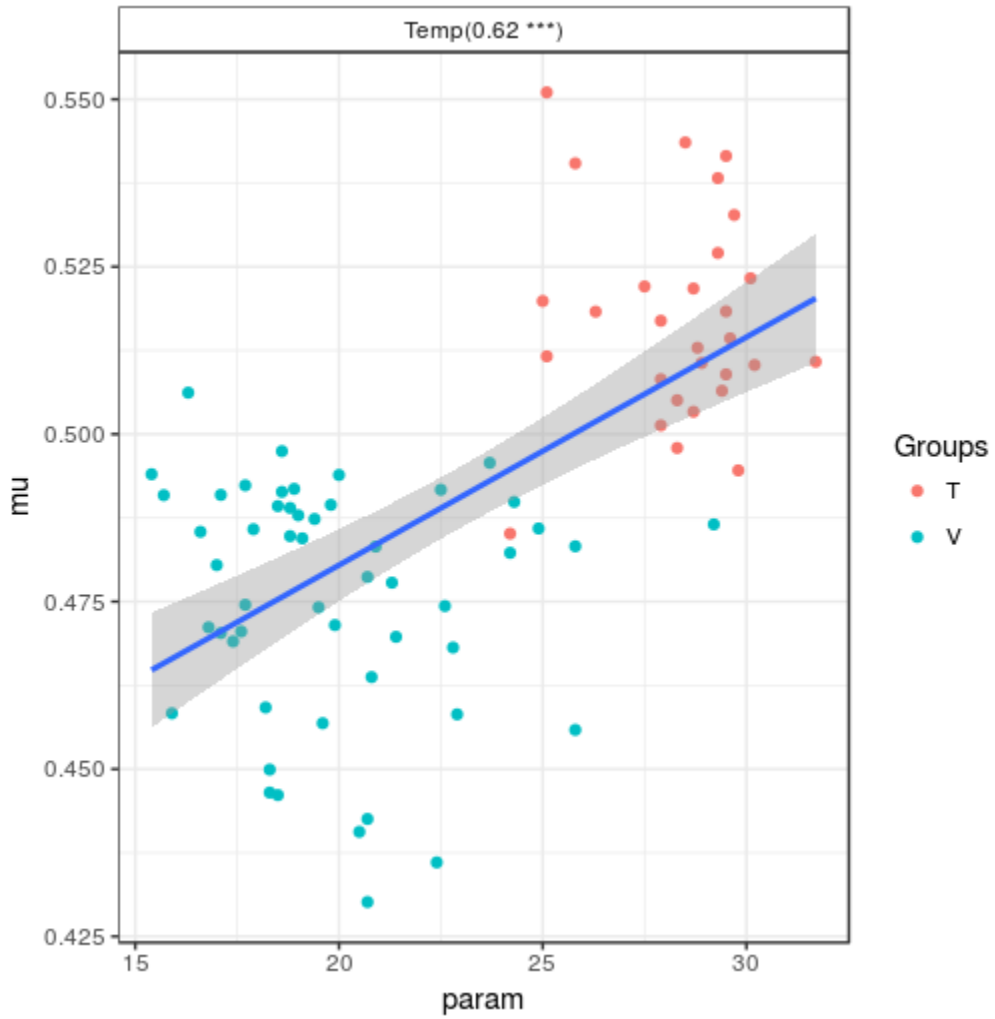
The results are visualised by producing a plot of fuzzy set against original values which is annotated with a correlation between them and a significance label.

`physeq` is a phyloseq object containing taxa abundance and meta data information. `grouping_column` is a character string for variable with respect to which the data should be grouped. `method` is an integer specifying method for computing similarity indices options include:

- 1 = Baroni-Urbani & Buser
- 2 = Horn
- 3 = Yule

`indices` an integer for column number corresponding to environmental variable of interest. The default is set for all variables. `filename` creates a file of fuzzy set correlation with a provided `filename`. In this example, we have selected a variable Temp for illustration.

```
p <- generateFSO(physeq, grouping_column = "Country", method = 2, indices = 2,
    filename = NULL)
print(p)
```

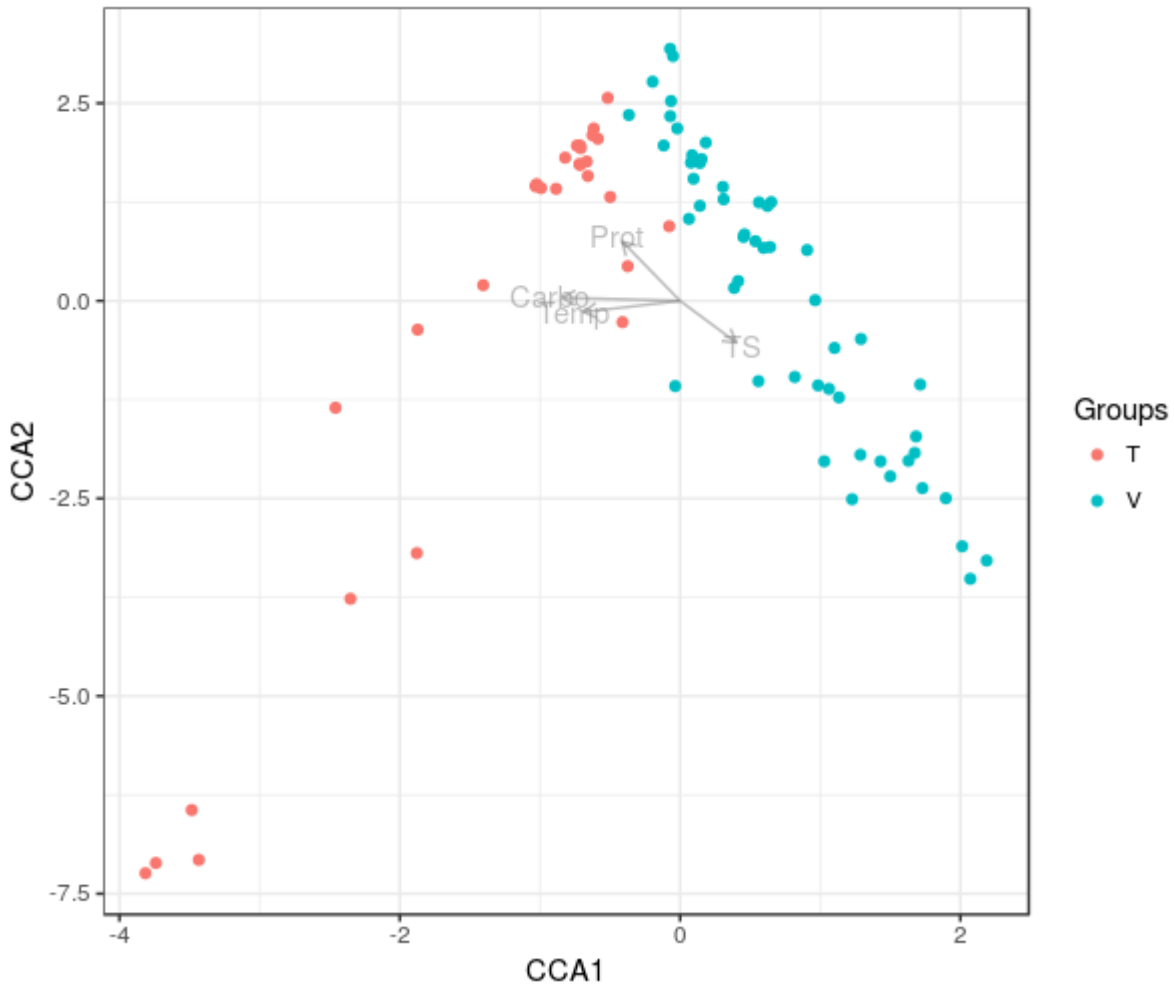


Canonical Correspondence Analysis

This function finds a set of best environmental variables that describe community structure.

`physeq` is a required phyloseq object containing taxa abundance and meta data. `grouping_column` is the variable in the meta data with respect to which the data should be grouped, `pvalueCutoff` the threshold p-value in `anova` of distance matrices, default set to `0.05`. `env.variables` is a list of variables prefered to be on the cca plot. `exclude.variables` a list of variables to be excluded from the cca plot. `num.env.variables` is an integer specifying the number of variables to show on the cca plot. This could be helpful to avoid over crowding of the plot.

```
plot_cca(physeq = physeq, grouping_column = "Country", pvalueCutoff = 0.01,
    env.variables = NULL, num.env.variables = NULL, exclude.variables = "Country",
    draw_species = F)
```



Differential expression

Here we implement two functions to find features that are up or down regulated in the compared groups using `DESeq2`package and Kruskal-Wallis test . Functions produce plots of the top most features annotated with corresponding adjusted p-values and abundance distribution. In addition we implement classification using `random forest` classifier to find most import features amongst the differentially expressed features.

DESeq2 significance

Deseq (Love, Huber, and Anders 2014) procedure which was originally developed for differential expression analysis of RNA-seq data is used. In this case, it is supposed that abundance of a feature in a given sample is modelled as a negative binomial distribution, whose mean depends on sample specific size factor and concentration of that feature in a sample. The wald test is used to test significance of coefficients in a Negative Binomial GLM based on sample estimates.

`physeq` is a required phyloseq object containing taxa abundance and meta data. `grouping_column` is character string specifying the variable in the meta data with respect to which the data should be grouped, `pvalue.threshold` and `lfc.threshold` are thresholds for p-values and log2 fold change,normalisation method `output_norm`, is a character string specifying normalisation method for the output data to be plotted.

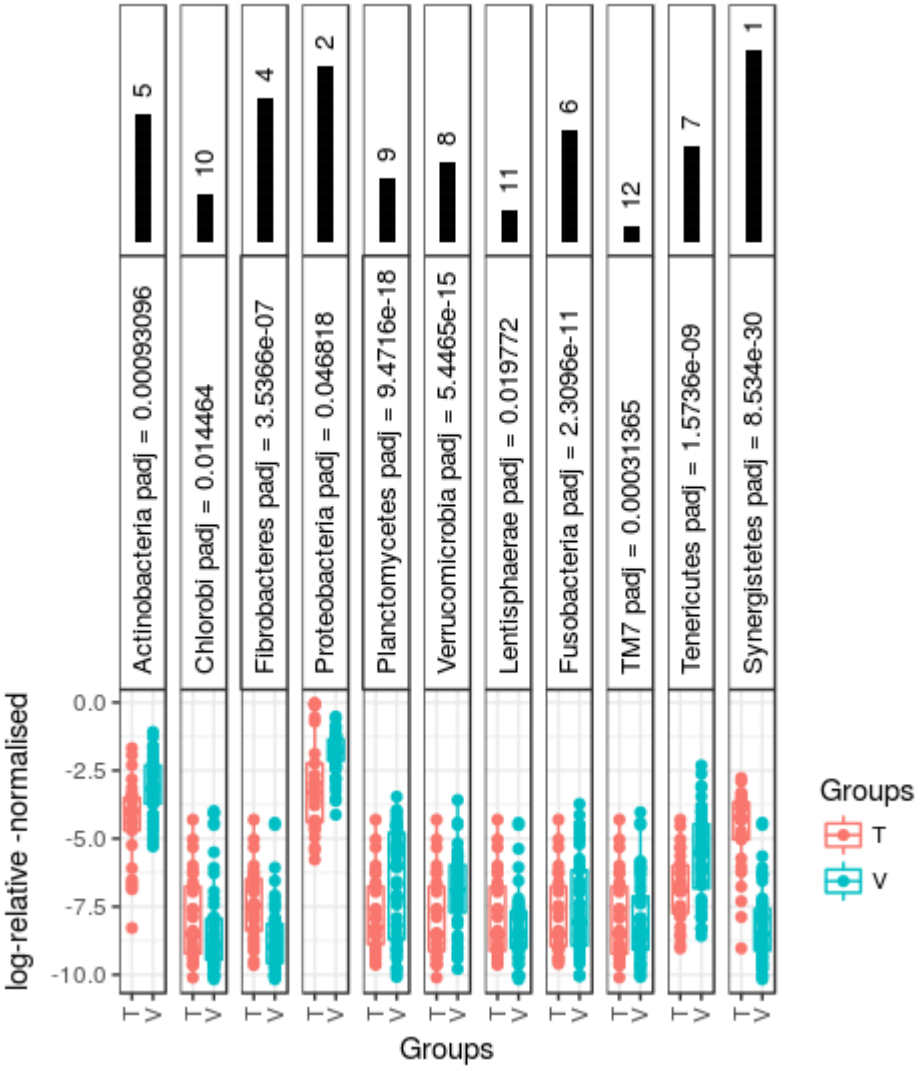
To find differentially expressed features between Tanzania and Vietnam, we use this example.

```
NB_sig <- differential_expression(physeq, grouping_column = "Country",output_norm=NULL,
                                  pvalue.threshold=0.05,lfc.threshold=0,filename=F)
```

The function `plot_signif` is used to produce the plots depending on supplied arguments as illustrated below.

To generate a plot showing differentially expressed (up or downregulated features in compared groups), corresponding adjusted p-values and rank of importance as detected by `random forest` classifier.

```
p<-plot_signif(NB_sig$plotdata)
print(p$SignfeaturesPlot)
```



Supplying a `filename` generates a file containing base mean, log2 fold change, raw and adjusted p-values and a group in which a certain feature is upregulated.

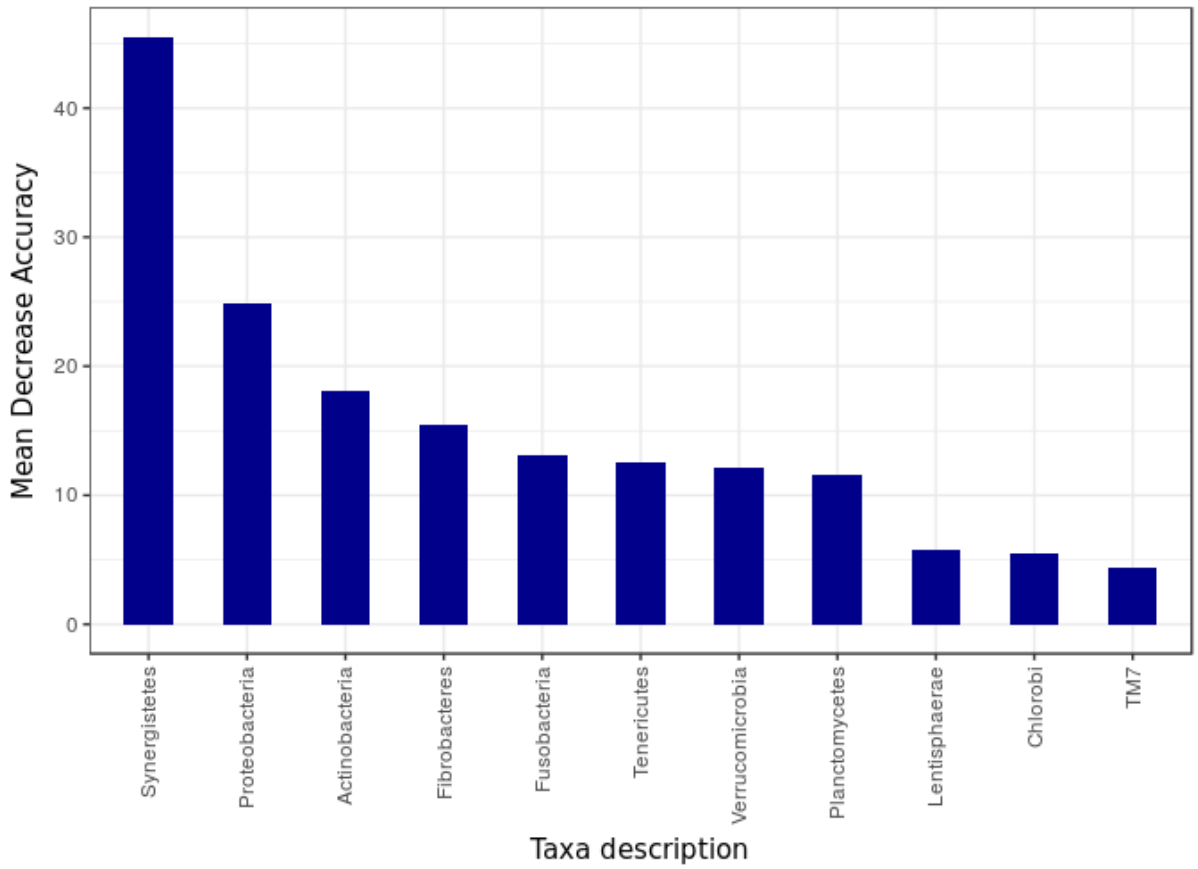
	baseMean	log2FoldChange	pvalue	padj	Upregulated
Synergistetes	40.480500	-4.9153857	2.844673e-31	8.534020e-30	T
Deinococcus-Thermus	145.904957	4.7731870	1.112263e-23	1.668394e-22	V
Planctomycetes	22.530003	4.2313944	9.471601e-19	9.471601e-18	V

Verrucomicrobia	9.793022	3.4771679	7.261986e-16	5.446490e-15	V
Fusobacteria	12.441201	3.5526292	3.849406e-12	2.309644e-11	V
Tenericutes	42.907190	2.7828493	3.147201e-10	1.573600e-09	V
Fibrobacteres	1.983949	-1.5982656	8.252045e-08	3.536591e-07	T
TM7	2.617610	1.4901325	8.363945e-05	3.136480e-04	V
Actinobacteria	475.873102	1.5888616	2.792890e-04	9.309632e-04	V
Chlorobi	2.134781	1.1525631	4.821450e-03	1.446435e-02	V
Lentisphaerae	2.025802	0.9546393	7.249707e-03	1.977193e-02	V
Proteobacteria	1576.418806	1.0319580	1.872728e-02	4.681820e-02	V

Random forest classifier (Liaw and Wiener 2002) is used to determine the importance of differentially expressed features/taxa to the microbial community. The measure used in this case is Mean Descrease in Accuracy which is reported for each of the features. This is obtained by removing the relationship of a feature and measuring increase in error. Consequently, the feature with high mean decrease in accuracy is considered most important.

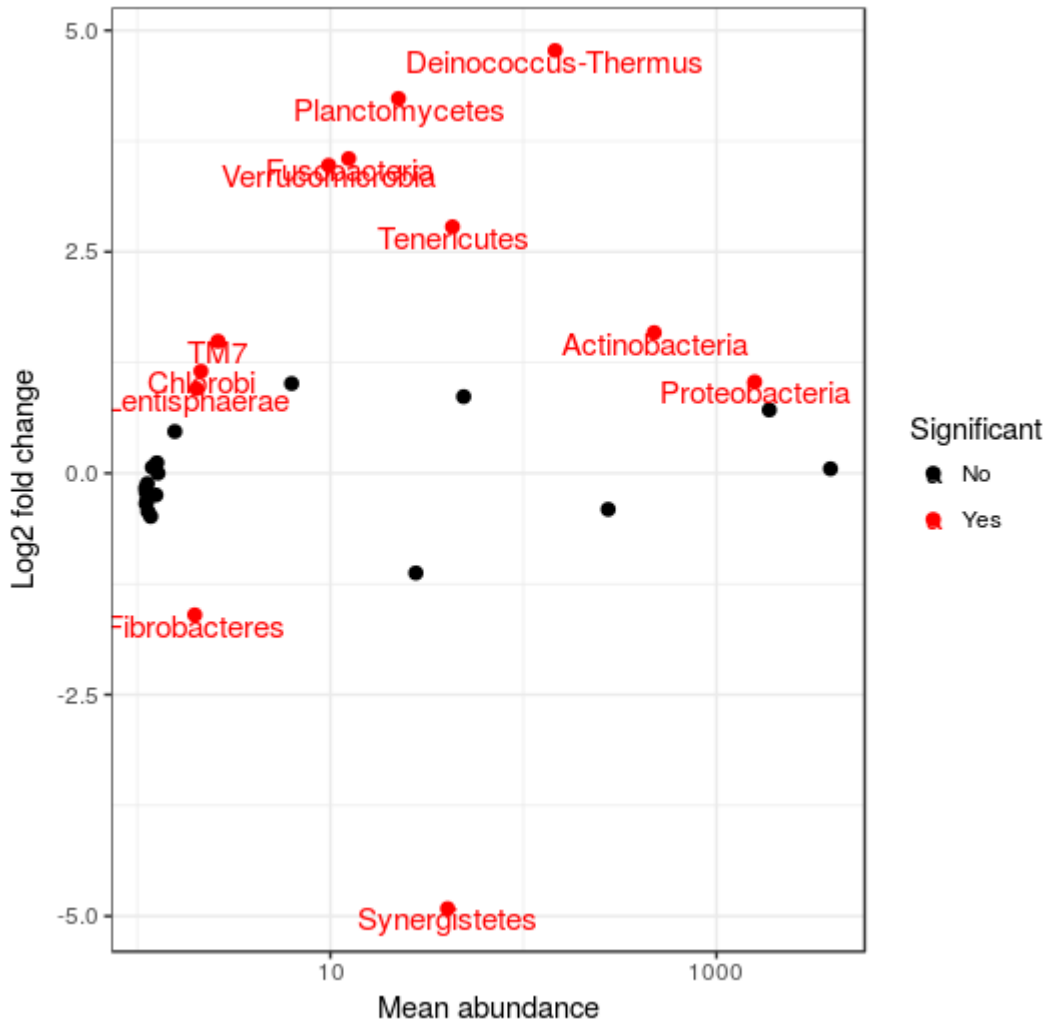
To get a stand alone visual representation of important features as obtained by random forest classifer, The plot shows Taxa description and corresponding Mean Decrease Accuracy values. This shows a bit more details since in the significant features plot we show only the ranks of these features but in this, it is the measured values of Mean Decrease Accuracy.

```
print(p$MDA)
```



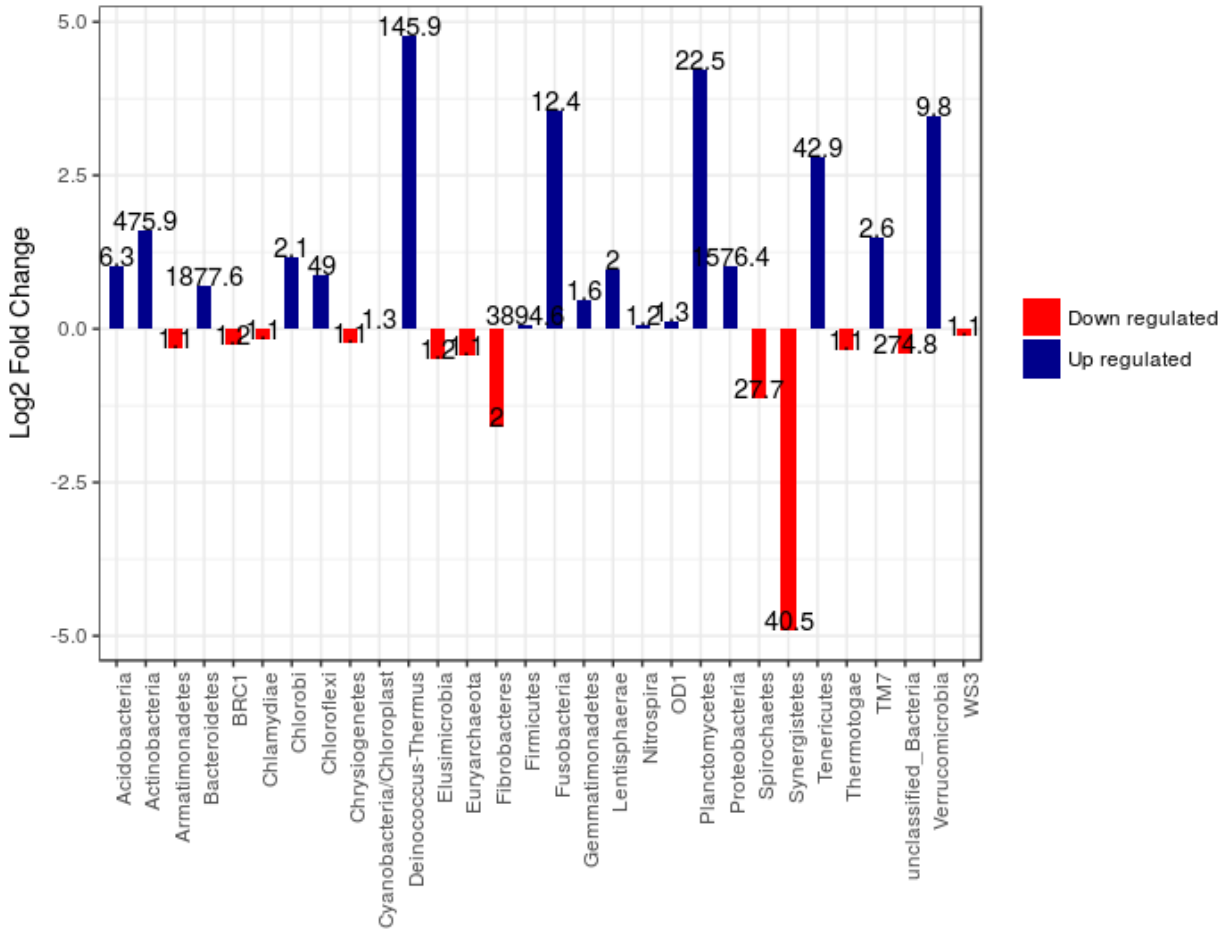
To get the Mean abundance plot (MA plot). This shows features that are detected as significant considering the threshold log fold supplied (`lfc`) against the mean abundance of the features.

```
print(p$MAplot)
```



An optional visual representation of features that are either up or down regulated. It is generated by setting `lfcplot=T`, it is off by default. The information provided here is strictly limited to log fold change and basemean values annotated for each taxa/feature, it help with identification of features with extreme log folds more easily. The height of the bars correspond to log fold change, positive and negative values show up and down regulation respectively.

```
print(p$lfcplot)
```



Kruskal-Wallis significant features

We perform kruskal-wallis test on feature abundance under different groups/conditions. Kruskal-Wallis is a non parametric method for testing whether samples originate from the same distribution. The p-values values generated are corrected for multiple testing using family wise error rate. Significance is based on the corrected pvalue threshold which is specified via arguments. Significant features are assigned importance using random forest classifier. The measure of importance used in this case is mean decrese accuracy.

`physeq` is a required phyloseq object containing taxa abundance and meta data. `grouping_column` is character string specifying the variable in the meta data with respect to which the data should be grouped, `pvalue.threshold` is the significance threshold for p-values. To writes the information of up and down regulated features to a file, a `filename` should be supplied.

This examples shows how to find differentially expressed features between Tanzania and Vietnam using Kruskal-Wallis test. Firstly, we transform taxa abundance data using log-relative transformation.

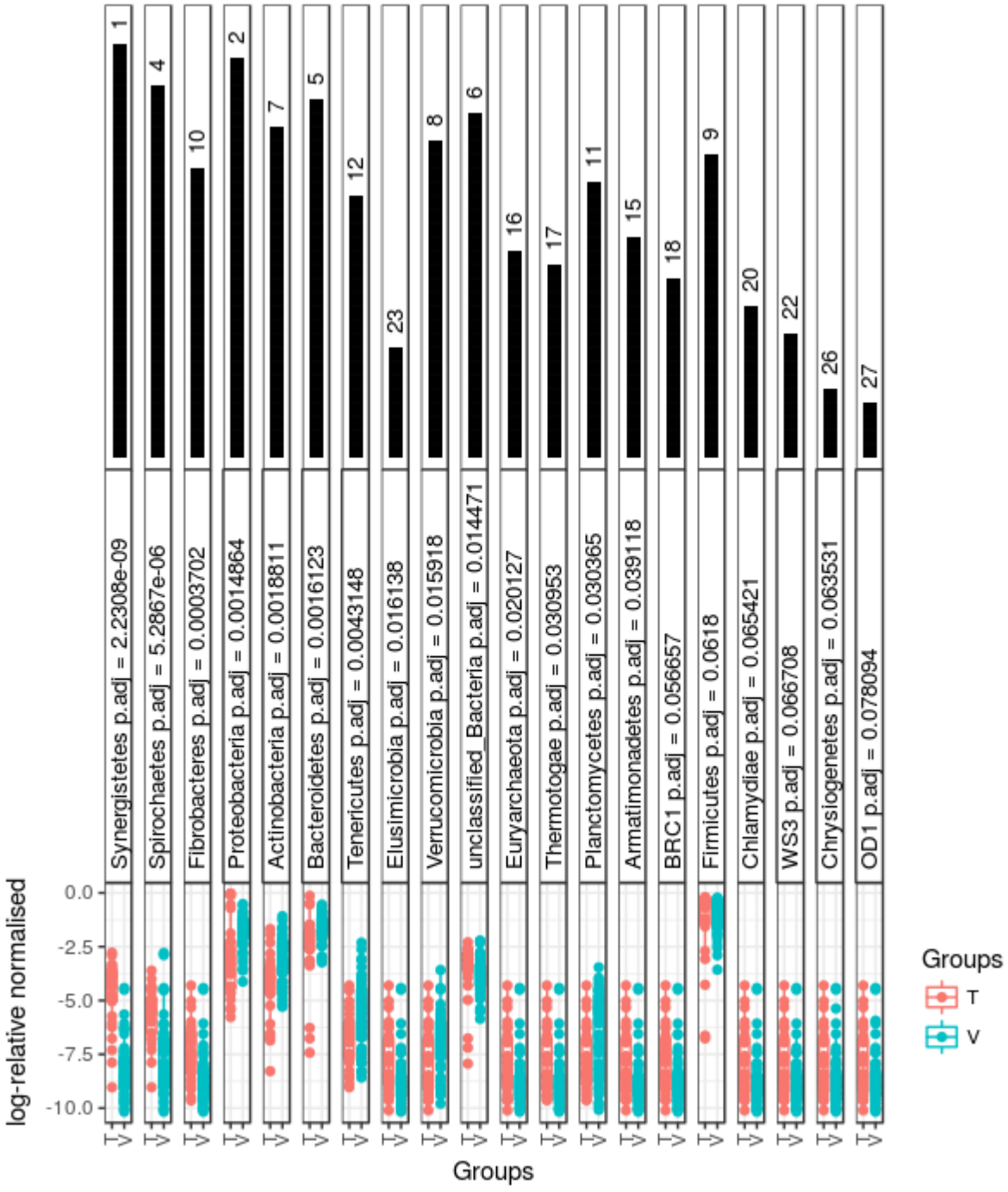
```
physeq <- normalise_data(physeq, norm.method = "log-relative")
```

```
kw_sig <- kruskal_expression(physeq, grouping_column = "Country")
```

The function `plot_signif` is used to produce the plots.

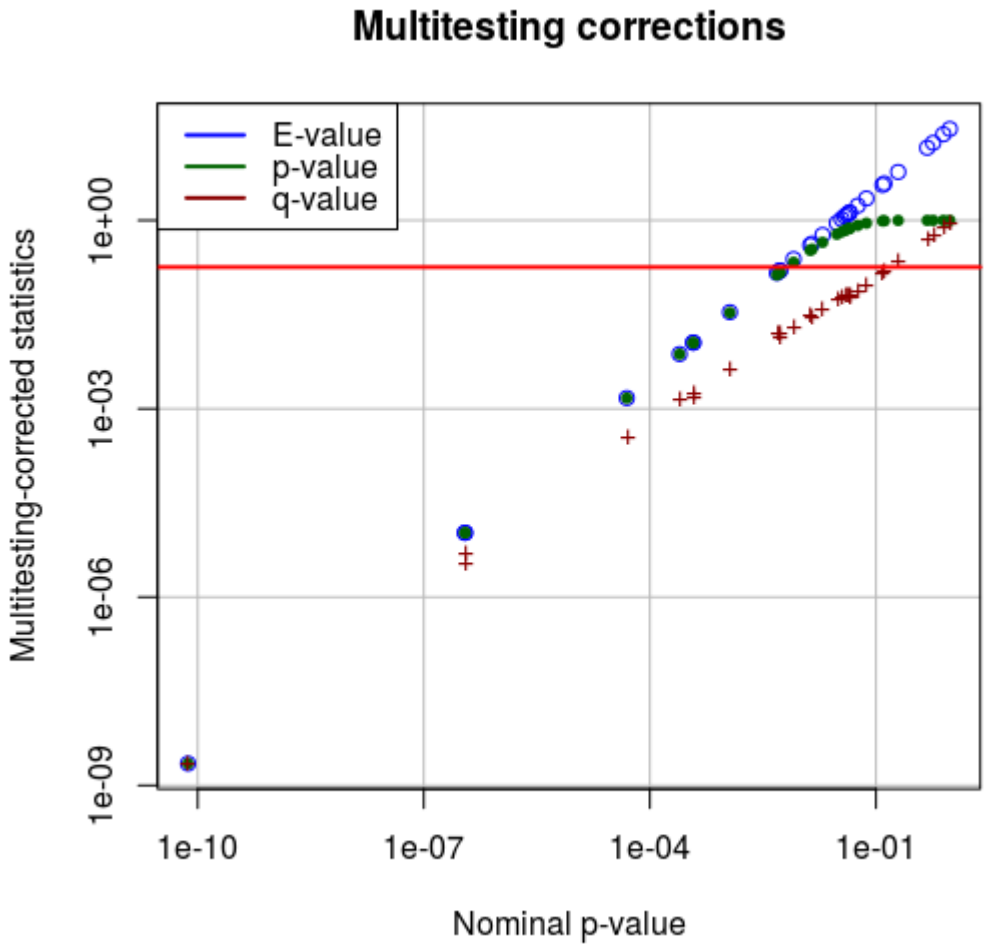
To plot features that are most significantly up or down regulated between the two groups with corresponding p-values and ranks of importance as detected by random forest classifier.

```
print(p$SignfeaturesPlot)
```



To generate plot for multiple testing corrections.

```
print(p$corrections)
```



Supplying a `filename` generates a file containing detailed information about significantly differentially expressed features.

```
      id      p.value      E.value         FWER q.value.factor      q.value
Synergistetes 7.436027e-11 2.230808e-09 2.230808e-09      30.000000 2.230808e-09
Spirochaetes 3.524440e-07 1.057332e-05 1.057327e-05      15.000000 5.286660e-06
Deinococcus 3.524440e-07 1.057332e-05 1.057327e-05      10.000000 3.524440e-06
Fibrobacteres 4.935991e-05 1.480797e-03 1.479738e-03       7.500000 3.701993e-04
Proteobacteria 2.477304e-04 7.431913e-03 7.405278e-03       6.000000 1.486383e-03
Bacteroidetes 3.762102e-04 1.128631e-02 1.122495e-02       5.000000 1.881051e-03
Actinobacteria 3.762102e-04 1.128631e-02 1.122495e-02       4.285714 1.612329e-03
Tenericutes 1.150604e-03 3.451813e-02 3.394838e-02       3.750000 4.314767e-03
Elusimicrobia 4.841365e-03 1.452410e-01 1.354911e-01       3.333333 1.613788e-02
Verrucomicrobia 5.305904e-03 1.591771e-01 1.475161e-01       3.000000 1.591771e-02
```

To get a stand alone visual representation of important features as obtained by random forest classifer, The plot shows Taxa description and corresponding Mean Decrease Accuracy values. This shows a bit more details since in the significant features plot we show only the ranks of these features but in this, it is the measured values of Mean Decrease Accuracy.

```
print(p$MDA)
```

Kernel-based differential analysis

Here we implement differential analysis using a distance-based kernel score test. It allows us to obtain a set(s) of differentially expresssed features (OTUs) or measured environmetal variables. The sets are obtained by grouping based on correlation of feature abundance/ measure of the environmental variables. The function returns a ggplot object showing significant feature/environmental variable sets annotated with corresponding adjusted p-values.

`physeq` is a phyloseq object containing taxa abundance and meta data information. `grouping_column` is a character string for variable with respect to which the data should be grouped. `analyse` is character string specifying whether to analyse taxa abundance ("abundance") or sample data ("meta"). Default is set to analyse taxa abundance. `p.adjust.method` character string for a method to be used for adjusting p-values for multitesting corrections, default is "BH" with options which include: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none". `adjusted.pvalue.threshold`is the threshold for the adjusted p-values. `...` others arguments available to `dscore` and `sscore` functions.

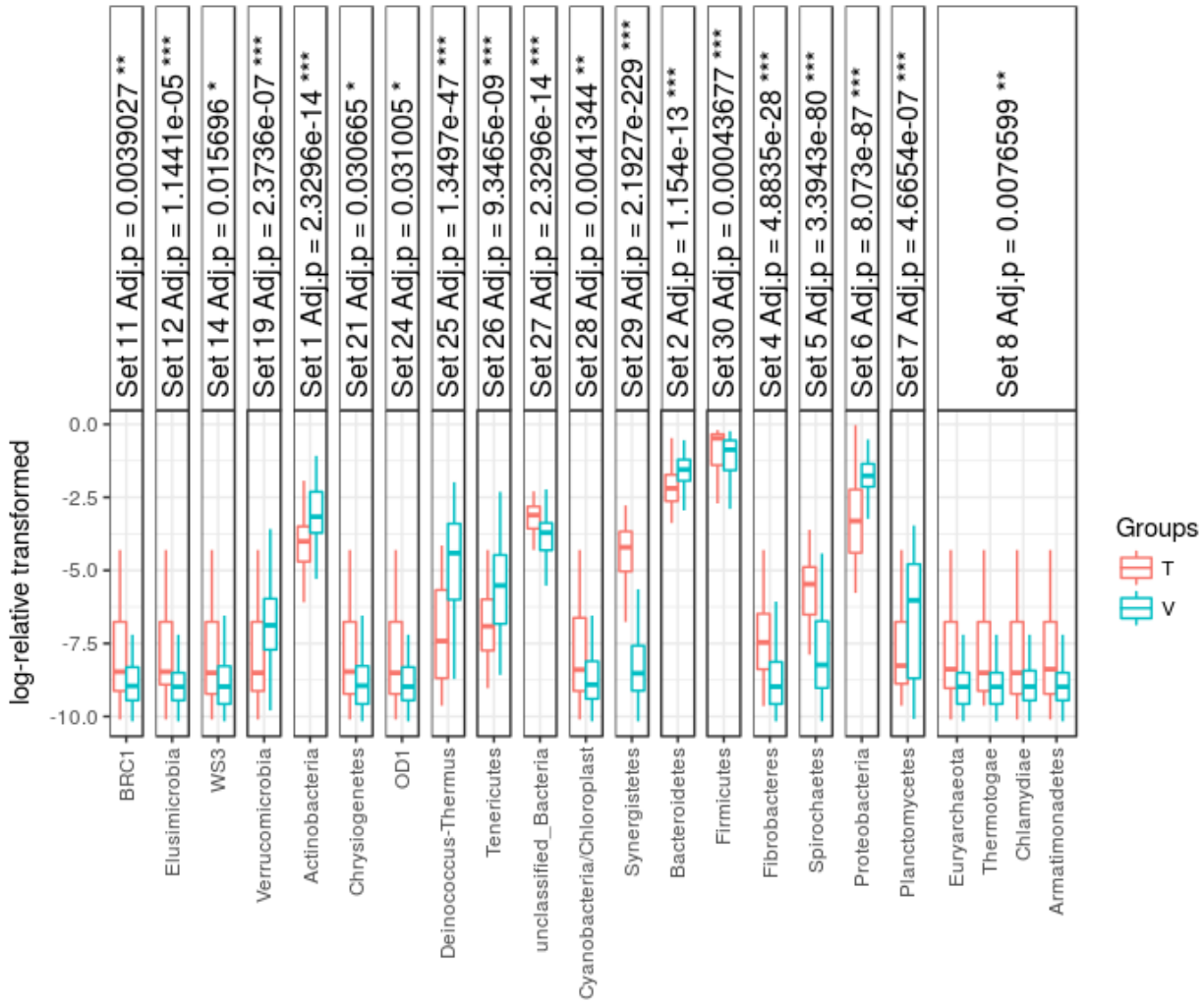
The example below illustrates how to obtain sets of features that are differentially expressed between Tanzania and Vietnam. First, we apply log-relative transformation to the abundance data because this method is designed to handle fractional data and not counts. Therefore, this transformation avails the data in a type required by the method.

```
physeq <- normalise_data(physeq, norm.method = "log-relative")
```

```
kda_sig <- KDA(physeq, grouping_column="Country", analyse="abundance", method="dscore",p.adjust.method=
```

Then use `kda_plot` function to visualise the results.

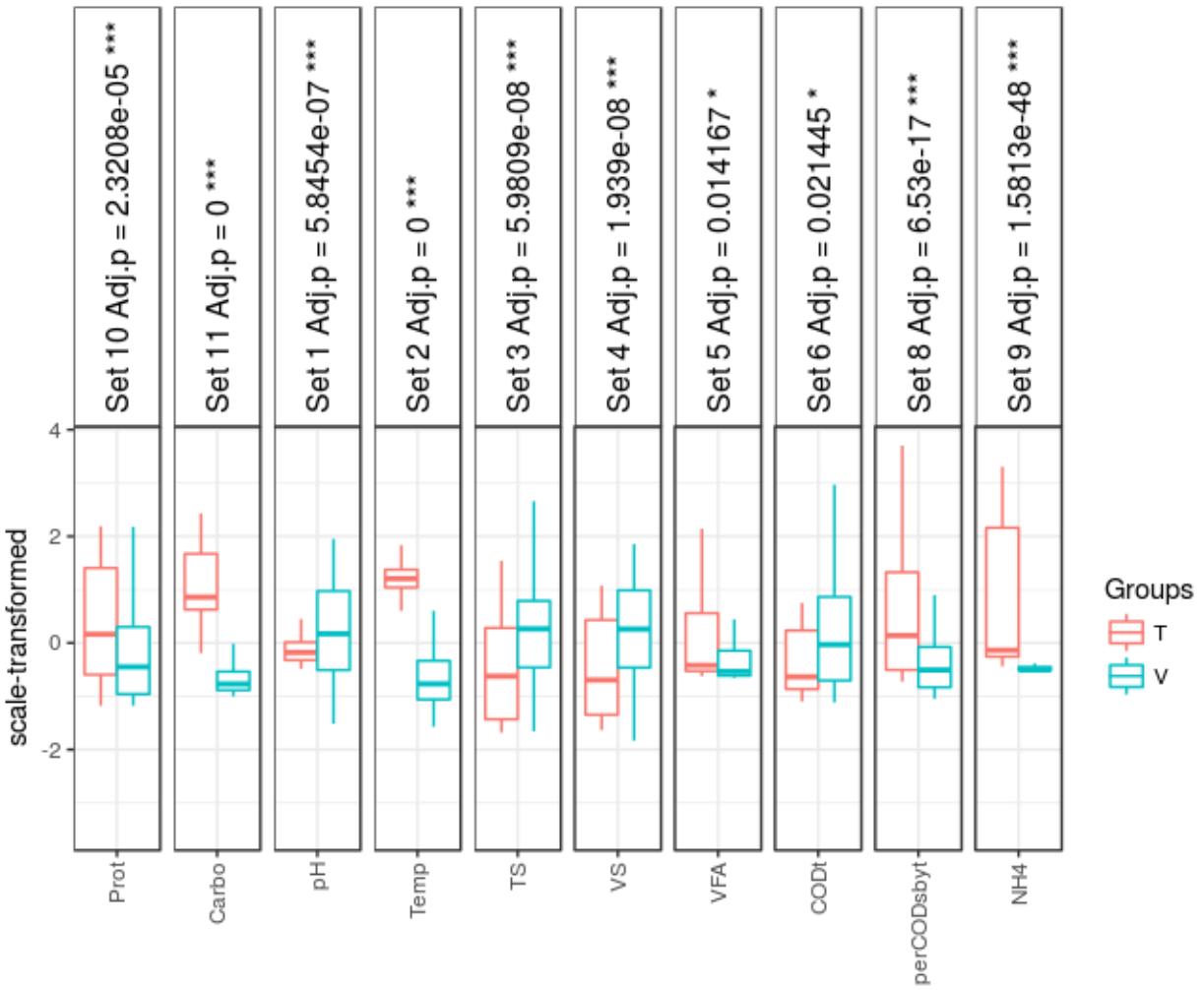
```
p <- kda_plot(kda_sig$plotdata)
print(p)
```



To use this method on meta data, a suitable normalisation should be applied. In this case, scale transformation is applied on the sample data. In addition, a list of variables of interest can be supplied via the argument `select.variables`. If not given, all the numerical variables in meta data are analysed as illustrated in the following example.

```
physeq <- normalise_data(physeq, norm.method = "scale")
```

```
kernel.meta <- KDA(physeq, "Country",analyse = "meta", select.variables=NULL)
p <- plot_kda(kernel.meta$plotdata)
print(p)
```



Co-occurrence pattern analysis

Co-occurrence pattern analysis is used to identify co-occurring features/taxa in community data under specified environmental conditions. Co-occurrence is measured as positive correlation whose threshold(s) can be specified as indicated in arguments section. Amongst these features, pairwise co-occurrences which are outstanding within sub communities are detected. p-values generated during pairwise correlations tests are adjusted for multiple comparisons by false discovery rate. The network statistics used to assign importance of taxa/features include betweenness, closeness and eigenvector centrality. This implementation follows the procedure presented by (Williams, Howe, and Hofmockel 2014)

To generate network of taxa under different conditions as specified by the grouping variable, the function `co_occurence_network` is used. In addition to a phyloseq object and grouping variable character string, other arguments include: `rhos` which a list of threshold correlations. The default is set to c(0.5, -0.5, 0.75 and -0.75). `select.condition` is an optional list of conditions which should be among the levels of grouping column. `scale.vertex.size` and `scale.edge.width` are numbers to adjust the size and width of vertices and edges respectively. `method` is a character string that specifies correlation method used in computing correlation between taxa. `cor` is the default with an option of `bicor`. `...` is for other arguments parsable to network plot for example layout, label size among others. A plot showing relationship between eigen value and betweenness centality is obtained by setting `plotBetweennessEeigenvalue=T`.

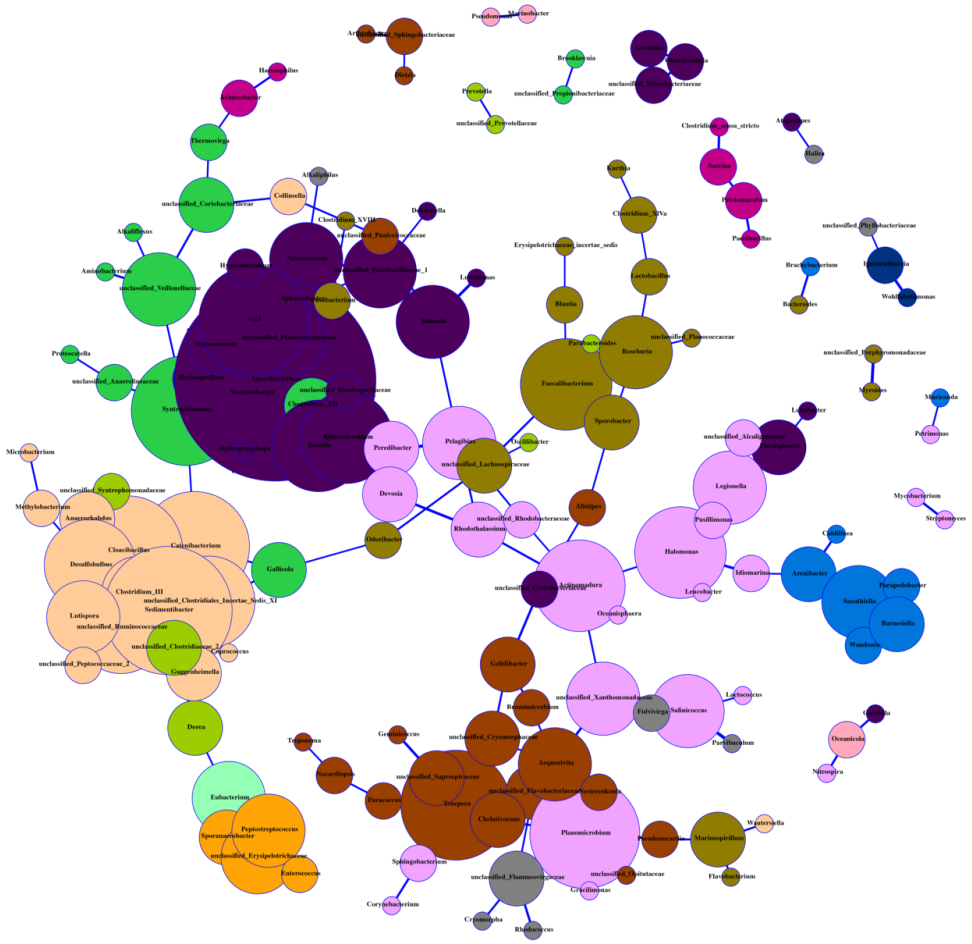
We illustate this at Genus taxonomic level with a threshold correlation of 0.35 for Vietnam as specified in the

selection of condition argument.

```
physeq <- taxa_level(physeq, which_level = "Genus")
```

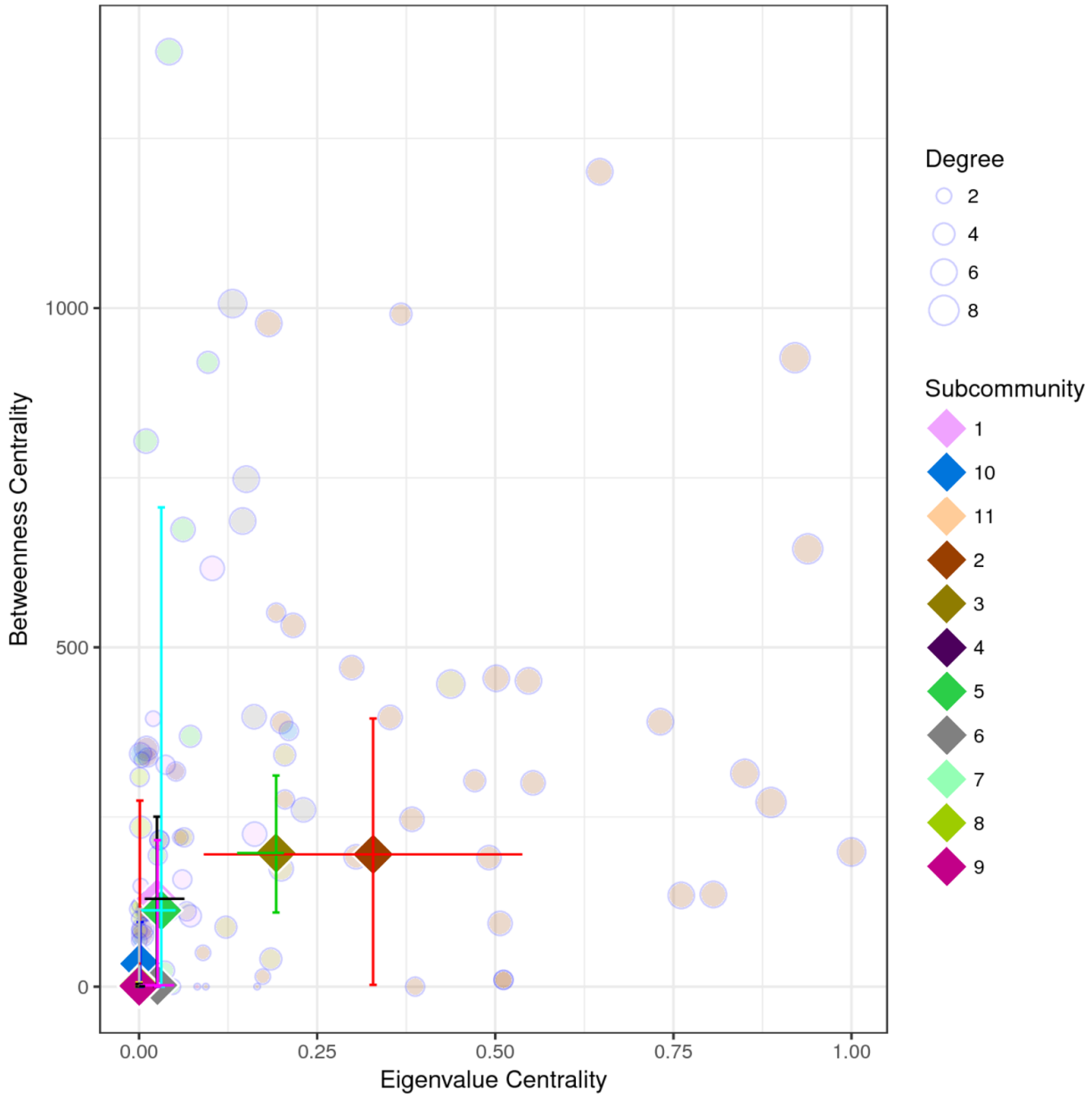
```
co_occr <- co_occurence_network(physeq, grouping_column = "Country", rhos = 0.35, method="cor", qval_thr
```

Note: Nodes are colored as per corresponding sub community. The size of the nodes is proportional to its own total degree. The width of the edges is proportional to the correlation between the two nodes to which it corresponds. Positive and negative correlations between taxa(nodes) are indicated by blue and red color of the edges respectively.



Setting `plotBetweennessEeigenvalue=T` produces plot(s) of betweeness versus eigenvector centrality at each of the specified correlations and conditions. As noted earlier, these are measures of importance of taxa in the network. Betweenness of taxa in this case is a measure of taxa's control in the network. High

betweenness centrality implies that a corresponding node has more influence in the network and viceversa. Eigen vector centrality measures taxa's linkage to others in the network taking into account how connected they are.Therefore, taxa with high eigenvector centrality is linked to highly linked taxa.



Roles of taxa

Features identified sub communities are assigned roles in the network using a procedure provided by (Guimera and Amaral 2005). The metrics used include: within-module degree which measures how well a particular feature is connected to others in the same subcommunity (module) and among-module connectivity which measures how a feature is linked to other modules in the network. Features are classified as ultra peripherals, peripherals, provincial, connectors, kinless, module hubs, or non hubs.

The function takes a graph object returned from `co_occurence_network` function as an argument and assigns roles to each of the features in the network.

We illustrate this using the graph obtained above.

```
taxa.roles <- module.roles(co_occur$net$graph)
```

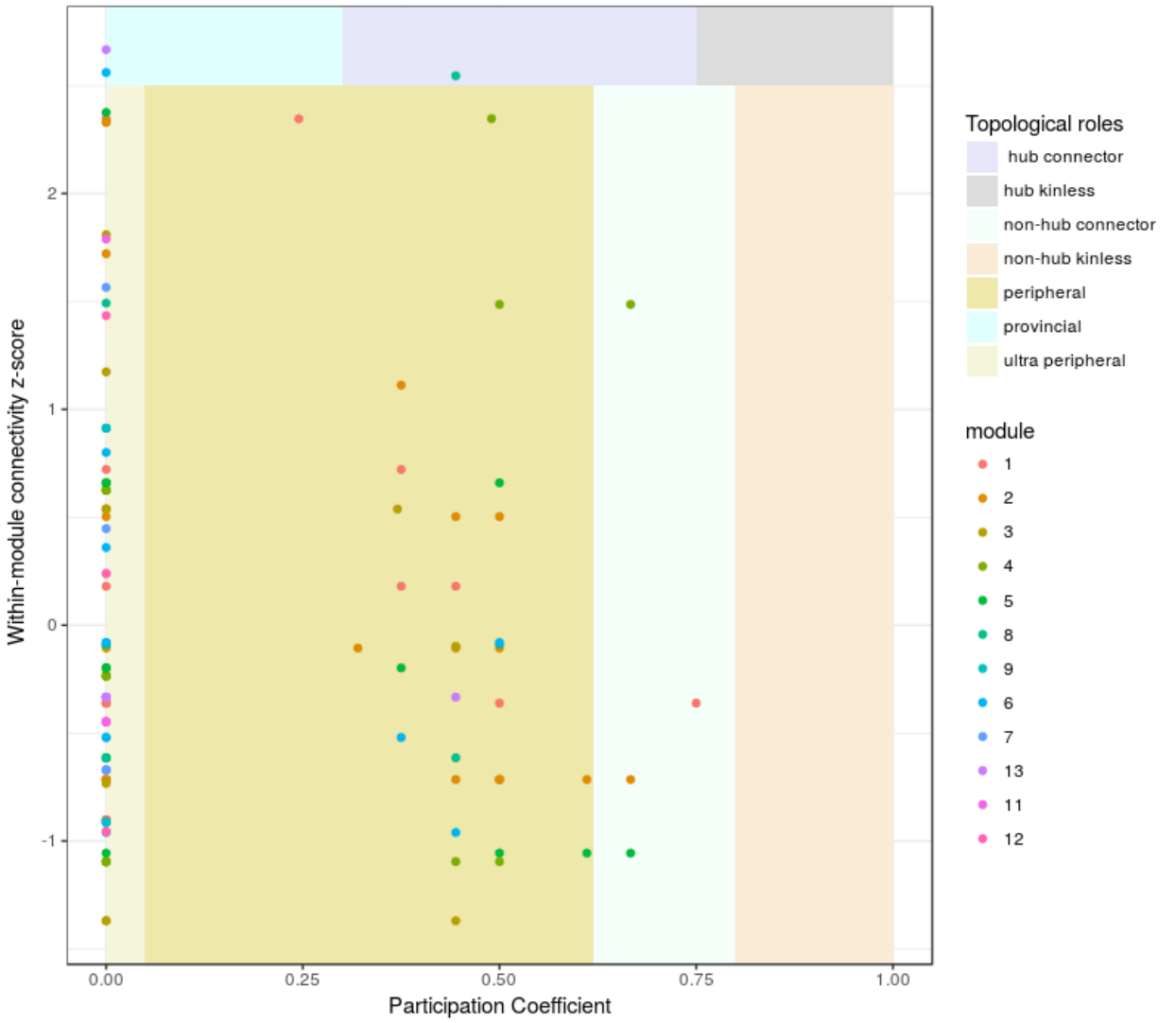
```
head(taxa.roles)
```

```
                                        z         p          roles
Dietzia                         -0.9114654 0.0000000 provincial hub
Streptomyces                    -0.9114654 0.0000000 provincial hub
Brumimicrobium                  -0.9114654 0.0000000 provincial hub
Nitrosomonas                     0.6076436 0.0000000 provincial hub
Sphingobacterium                 0.6076436 0.0000000 provincial hub
unclassified_Flavobacteriaceae -0.1519109 0.4444444  connector hub
```

To produce a visualisation of the results, use function `plot_roles` which takes a result of `module.roles`.

```
p <- plot_roles(taxa.roles)

print(p)
```



Correlations between subcommunities and environmental variables

To explore how the sub communities respond to environmental traits, we consider the correlation betweeen the taxa with maximum betweenness within a sub community and the enviromental variables. This is chosen because it is a good representation of the subcommunity. This is implemented in accordance to (Deng et al. 2012) where correlations between module-based eigengenes and environmental factors are used to detect the modules response to environmental change.

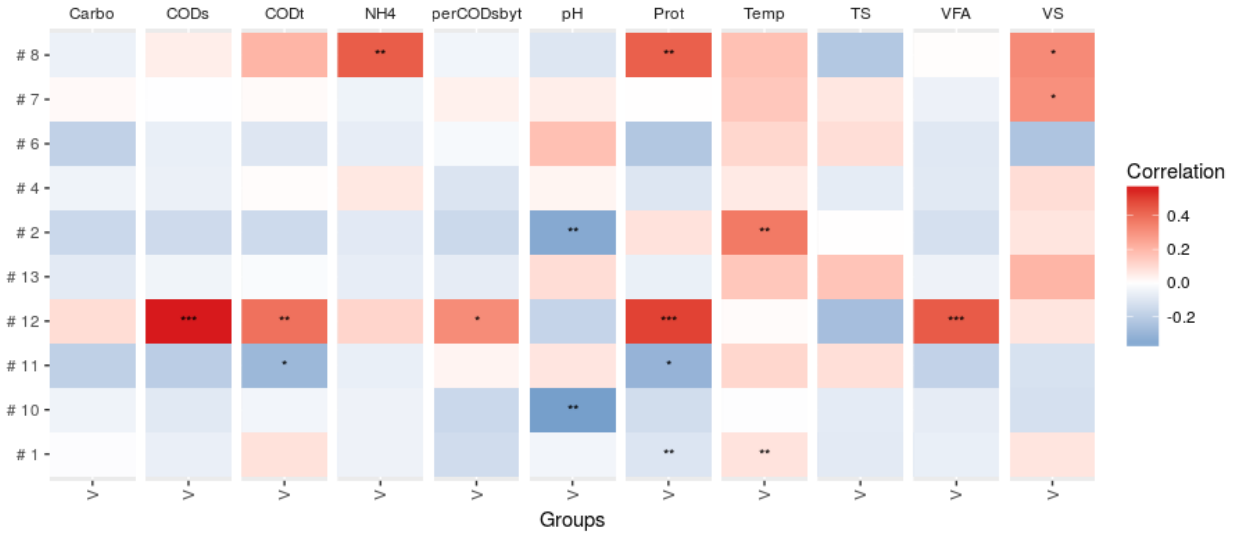
The function to perform this takes a result from co-occurence network. Other arguments include: `select.variables`, `method`,`padjust.method`, `adjustment` as they are available to function `env_taxa_correlation` (see this function for explanation of arguments). It returns a data frame with correlations between each sub community (module) and environmental variables for all conditions. The example below illustrates this by using the co occurence network obtained above.

```
mod.env.cor <- module_env_correlation(co_occur)
```

A visualisation is produced using the function `plot_tax_env` as illustrated below. Significant correlations are annotated with significant labels.

```
p <- plot_taxa_env(mod.env.cor)
print(p)
```

The plot shows correlation between module number (indicated by #number) and environmental traits.



Correlation between numerical environmental variables and most abundant taxa

This function shows the relationship between most abundant taxa and numerical environmental variables based on correlation.The abundance of each feature/taxa is correlated with each of the environmental variables. A correlation test is performed and associated p-values are adjusted for multiple testing. The scheme of adjustment is elaborated in the arguments section. The function returns a data.frame with raw p-values, corrected p-values, and correlation results.

`physeq` is a phyloseq object containing taxa abundance and meta data information. `grouping_column` is a character string for variable with respect to which the data should be grouped. `method` a character string indicating which correlation coefficient is to be computed, available options are `"pearson"` which is also the default, `"kendall"` and `"spearman"`. `adjustment` is an integer with options `1,2,3,4,5` which indicate a way for adjusting p-values for multiple comparisions using Benjamin and Hochberg. These options have the following implications.

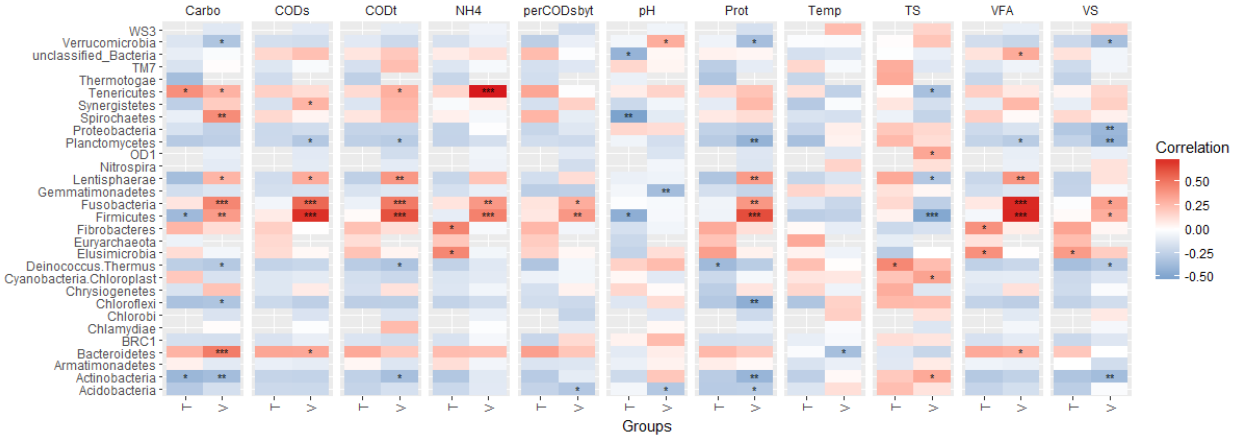
- 1 - donot adjust
- 2 - adjust environmental variables + Groups (column on the correlation plot)
- 3 - adjust Taxa + Groups (row on the correlation plot for each Groups)
- 4 - adjust Taxa (row on the correlation plot)
- 5 - adjust environmental variables (panel on the correlation plot)

`num.taxa` is an integer indicating the number of taxa to be used in the correlation plot, default is 50. `select.variables` is a list of environmental variables to be used in the correlation computation. If not specified, all numerical variables are used as shown in the example below.

```
env.taxa.cor <- env_taxa_correlation(physeq, grouping_column="Country", method="pearson", pvalue.thresh
                                padjust.method="BH", adjustment=5, num.taxa=50, select.variables=NULL)
```

Then visualise the correlation results using `plot_taxa_env` function.

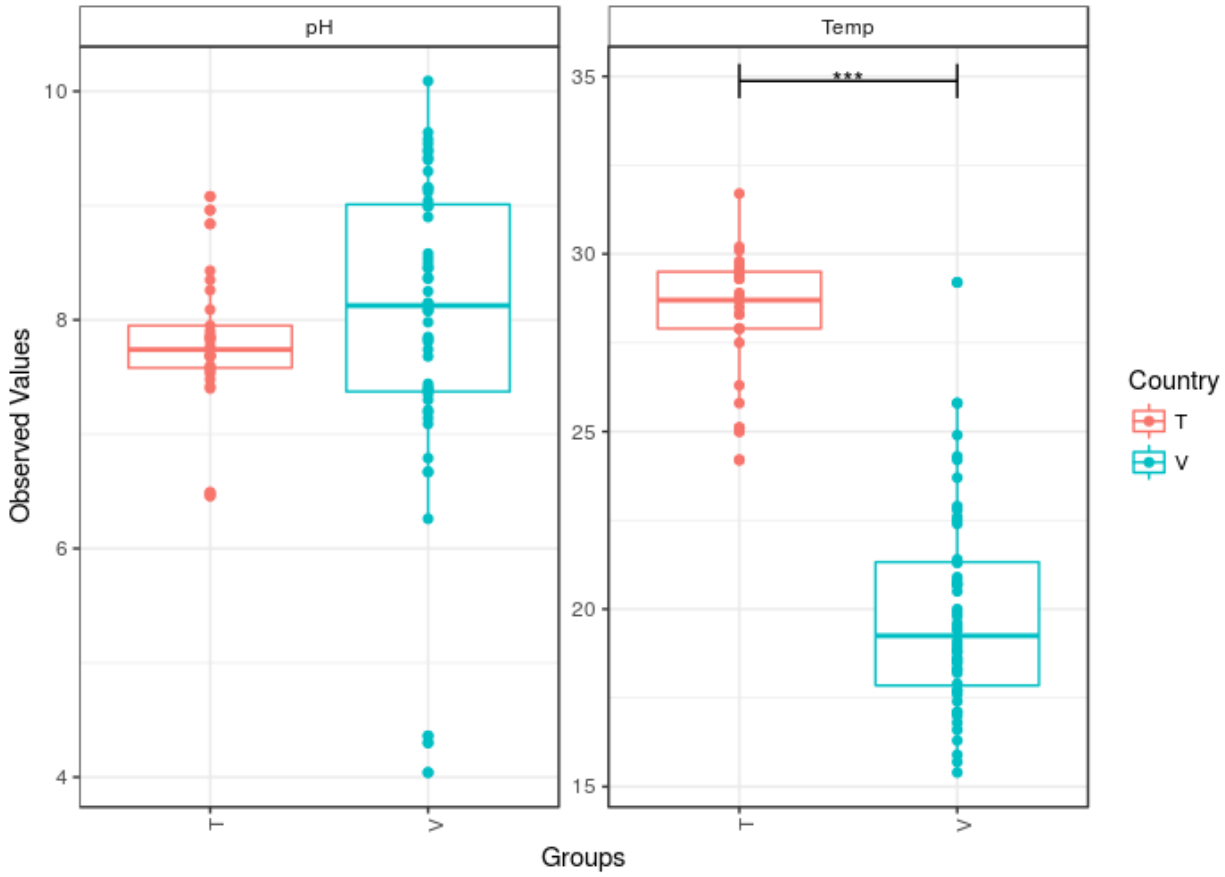
```
p <- plot_taxa_env(env.taxa.cor)
print(p)
```



ANOVA of environmental variables

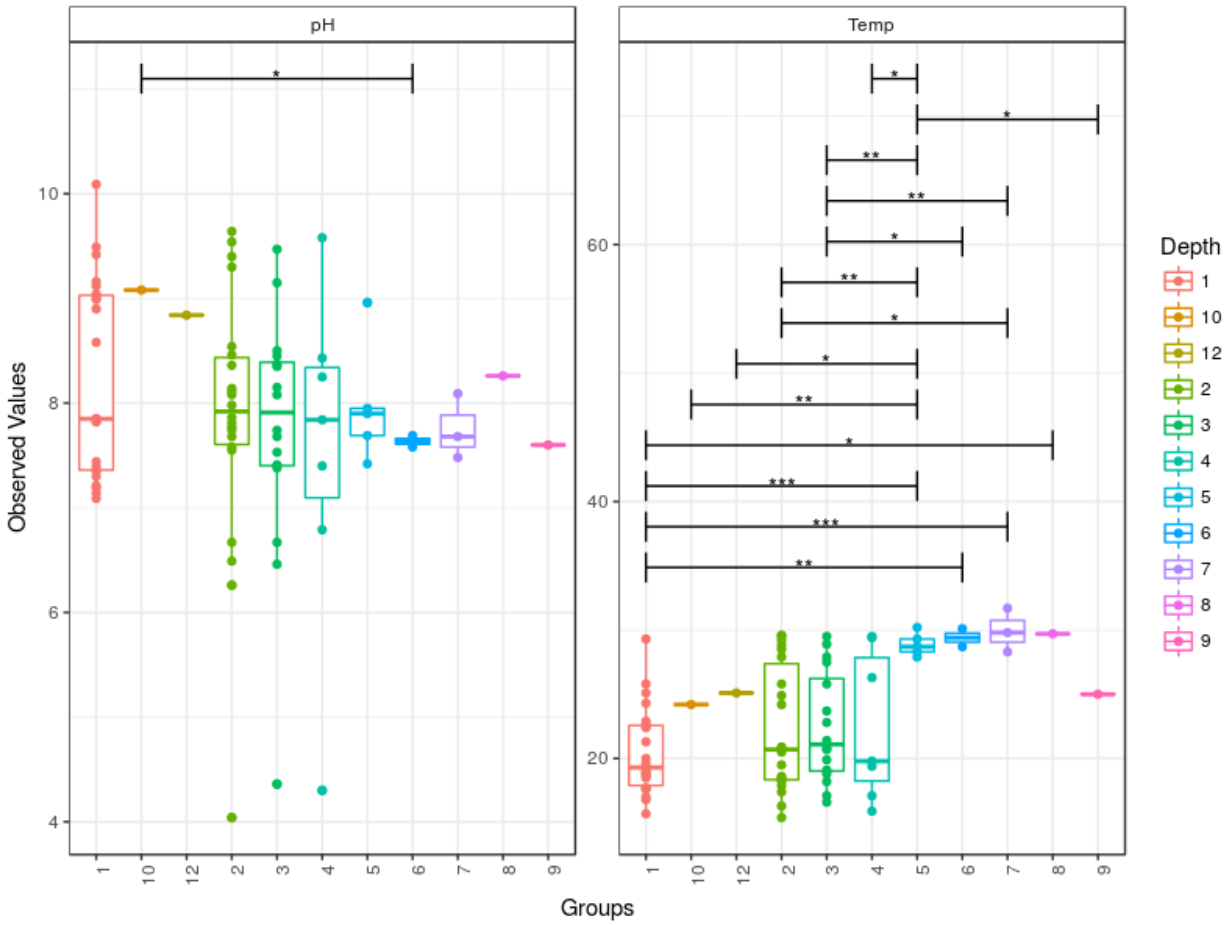
This function performs analysis of variance on selected environmental variables plots the distribution of variables annotated with significance of variation in specified groups. `physeq` is a required phyloseq object containing taxa abundance and meta data. `grouping_column` is character string specifying the variable in the meta data with respect to which the data should be grouped, `pvalueCutoff` the threshold p-value in `anova` of environment variables, default set to `0.05`. `selec.variables` is a list of character strings for the variables to be analysed. In the first example, two variables "Temp" and "pH" are selected and grouped with respect to "Country".

```
p <- plot_anova_env(physeq, grouping_column = "Country", pValueCutoff = 0.05,
    select.variables = c("Temp", "pH"))
print(p)
```



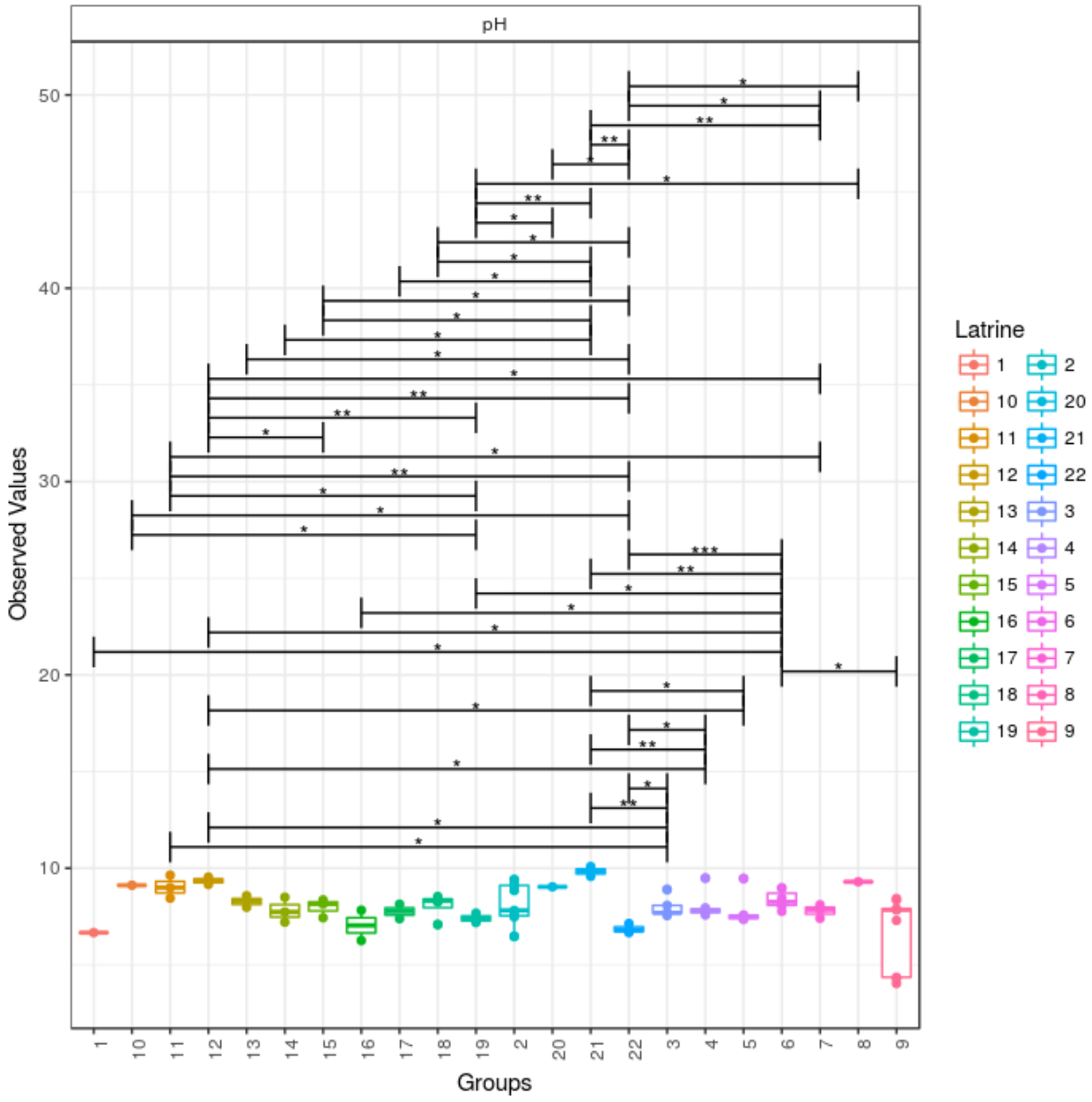
Selecting “Temp” and “pH” and grouping by “Depth”, we obtain the following plot.

```
p2 <- plot_anova_env(physeq, grouping_column = "Depth", select.variables = c("Temp",
    "pH"))
print(p2)
```



Selecting only “pH” and grouping by “Latrine”gives the following plot.

```
p <- plot_anova_env(physeq, grouping_column = "Latrine", select.variables = "pH")
print(p)
```



Dependencies

This packages depends on a number of other packages which include: phyloseq (McMurdie and Holmes 2013), vegan (J. Oksanen et al. 2007), DESeq2 (Love, Huber, and Anders 2014), ggplot2 (Wickham 2016),randomForest (Liaw and Wiener 2002), grid Extra (Auguie 2016), data.table (Dowle and Srinivasan 2017), fso (Roberts 2013), sna (Butts 2016), network (Butts 2008), WGCNA (Langfelder and Horvath 2012), igraph (Csardi and Nepusz 2006).

* References

Auguie, Baptiste. 2016. "GridExtra: Miscellaneous Functions for Grid Graphics." https://CRAN.R-project.org/package=gridExtra.

Butts, Carter T. 2008. "Network: A Package for Managing Relational Data in R." *Journal of Statistical*

Software 24 (2). http://www.jstatsoft.org/v24/i02/paper.

———. 2016. "Sna: Tools for Social Network Analysis." https://CRAN.R-project.org/package=sna.

Csardi, Gabor, and Tamas Nepusz. 2006. "The Igraph Software Package for Complex Network Research." *InterJournal* Complex Systems: 1695. http://igraph.org.

Deng, Ye, Yi-Huei Jiang, Yunfeng Yang, Zhili He, Feng Luo, and Jizhong Zhou. 2012. "Molecular Ecological Network Analyses." *BMC Bioinformatics* 13 (1). BioMed Central: 113.

Dowle, Matt, and Arun Srinivasan. 2017. "Data.table: Extension of Data Frame." https://CRAN.R-project.org/package=data.table.

Guimera, Roger, and Luis A Nunes Amaral. 2005. "Functional Cartography of Complex Metabolic Networks." *Nature* 433 (7028). NIH Public Access: 895.

Langfelder, Peter, and Steve Horvath. 2012. "Fast R Functions for Robust Correlations and Hierarchical Clustering." *Journal of Statistical Software* 46 (11): 1–17. http://www.jstatsoft.org/v46/i11/.

Legendre, Pierre, and Miquel Cáceres. 2013. "Beta Diversity as the Variance of Community Data: Dissimilarity Coefficients and Partitioning." *Ecology Letters* 16 (8). Wiley Online Library: 951–63.

Liaw, Andy, and Matthew Wiener. 2002. "Classification and Regression by RandomForest." *R News* 2 (3): 18–22. http://CRAN.R-project.org/doc/Rnews/.

Love, Michael I, Wolfgang Huber, and Simon Anders. 2014. "Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2." *Genome Biology* 15 (12). BioMed Central: 550.

McMurdie, Paul J, and Susan Holmes. 2013. "Phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data." *PloS One* 8 (4). Public Library of Science: e61217.

Oksanen, Jari, Roeland Kindt, Pierre Legendre, Bob O'Hara, M Henry H Stevens, Maintainer Jari Oksanen, and MASS Suggests. 2007. "The Vegan Package." *Community Ecology Package* 10: 631–37.

Roberts, David W. 2013. "Fso: Fuzzy Set Ordination." https://CRAN.R-project.org/package=fso.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer.

Williams, Ryan J, Adina Howe, and Kirsten S Hofmockel. 2014. "Demonstrating Microbial Co-Occurrence Pattern Analyses Within and Between Ecosystems." *Frontiers in Microbiology* 5. Frontiers Media SA.