Chapter 15

Diversity Analysis in Viral Metagenomes

Jorge Francisco Vázquez-Castellanos

Abstract

Viruses are the most abundant and diverse biological entity in the earth. Nowadays, there are several viral metagenomes from different ecological niches which have been used to characterize new viral particles and to determine their diversity. However, viral metagenomic data have the disadvantage to be high-dimensional compositional and sparse. This type of data renders many of the conventional multivariate statistical analyses inoperative. Fortunately, different libraries and statistical packages have been developed to deal with this problem and perform the different ecological and statistical analyses. In the present chapter, it is analyzed simulated viral metagenomes, based on real human gut-associated viral metagenomes, using different R and python packages. The example presented here includes the estimation and comparison of different indexes of diversity, evenness, and richness; perform different ordination and statistical analysis using different dissimilarity metrics; determine the optimal cluster configuration and perform biomarker discovery. The scripts and the simulated datasets are in https://github.com/jorgevazcast/Viromic-diversity

Key words Ordination analysis, Viral metagenomics, Alpha diversity, Beta diversity, Clustering and biomarker discovery

1 Introduction

Biodiversity refers to the wide variety of living beings on Earth and the natural patterns that make it up. The biodiversity describes the diversity from genes to species and studies the number of different populations, species and, the durable interactions between species and their environment [1]. Species diversity refers to the number of species that are found within a landscape. The study of diversity can be divided into three distinct types: alpha, beta. and gamma diversity. The alpha diversity measures how many different species are and how evenly distributed are in a certain habitat, study-area, or sample, it is also known as the within-sample diversity [2]. The alpha diversity can be measured using indexes such as the Shannon index [2, 3] or the effective number of species [4]. Moreover, different approximations such as the species richness, which quantifies the number of different species within a habitat (ex: the Chao1 [5, 6] and the ACE index [6]) and the species evenness, which is the

Andrés Moya and Vicente Pérez Brocal (eds.), *The Human Virome: Methods and Protocols*, Methods in Molecular Biology, vol. 1838, https://doi.org/10.1007/978-1-4939-8682-8_15,

similarity in the relative abundance of the species (ex: Pielou's J′ evenness [7]), are complementary to describe alpha diversity. The beta diversity was defined as "the extent of change in community composition, or degree of community differentiation, in relation to a complex-gradient of the environment, or a pattern of environments" [8]. It is also known as the between-sample diversity and measures the differences between two different habitats or samples. Commonly different dissimilarity distance and index are used to study beta diversity. In metagenomic analysis the most common are the Bray-Curtis dissimilarity (quantifies dissimilarities based on the species counts), the Jaccard distance (quantifies the differences between samples based on the presence/absence species), the UniFrac distance (quantifies dissimilarities incorporating phylogenetic information), the Euclidean distance after a proper data normalization among several others [8, 9]. Finally, the gamma diversity is the total species diversity in a landscape, this encompasses alpha and beta diversity [8].

Diversity analysis is fundamental for the study of the microbiome so that there are different libraries and specialized packages mainly focused on the study of bacterial diversity [10, 11]. This is because bacterial microbiomes have been the most abundant in the last decade. Recent massive metagenomic studies [12, 13] have increased the knowledge of viral diversity and distribution, have discovered a large number of possible new viral species, and have elucidated their putative hosts. In this scenario, the study of the viral metagenomics is a useful tool for studying viral diversity since it allows characterizing phages that infect non-culturable bacterial species. However, there are technical limitations which complicate the study of viral diversity such as the poor taxonomic annotation [14], the lack of phylogenetic gene markers, and the high diversity. The sum of these generates a large number of sequences without correct taxonomic annotations and the identification of viral particles which are specific for only a few samples. These methodological biases generate sparse data [15], which refer to data matrices that contain a large proportion of zeros. Sparse data is problematic for several ordination and regression analyses that assume normality. Moreover, viral metagenomes also suffer from the high-dimensional compositional data problem which is common for all the metagenomic datasets. Compositional data renders many standard multivariate statistical methods inappropriate or inapplicable [15, 16]. Fortunately, there are different statistical packages which can correctly deal with this type of problems and make estimates of the diversity in viral metagenomics as well as the detection of biomarkers related to different conditions. The purpose of this chapter is to show these techniques and provide a practical example.

2 Materials

1. PC with Linux 3 or higher (tested on Ubuntu 16.04 and 14.04 distributions), alternatively, macOS or Windows 10 (with Windows subsystem for Linux enabled).
2. Although all these analyses can be performed on a computer with at least two processors and two gigabytes of RAM, I recommend using a computer that has at least 8 processors (64 bits) and 8 gigabytes of RAM. This will help to greatly reduce the computation times in several of the analyses.
3. R version 3.2.3 or above.
4. python 2.7 or above and below python 3.
5. Inkscape https://inkscape.org/en/.
6. Bash 4 or equivalent shell.
7. Stable internet connection (>1 Mbps is preferable).

3 Methods

All scripts and datasets are accessible at https://github.com/jorgevazcast/Viromic-diversity.

3.1 Install R and the Packages Necessary for the Data Analysis

3.1.1 Open a Bash Terminal and Install R

Important: the "$" symbol indicates the prompt and is not part of the command and you should not copy-paste these instructions into the prompt line.

```
$ sudo add-apt-repository 'deb [arch=amd64,i386]
https://cran.rstudio.com/bin/linux/ubuntu xenial/'
(see Note 1)
$sudo apt-get update
$ sudo apt-get install r-base r-base-dev
```

3.1.2 Open R and Install the Necessary Packages for the Current Analysis

R is open by typing R in the Bash prompt.

```
$ R
```

If the installation was successful, this should appear on the screen.

R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing.
Platform: ×86_64-pc-linux-gnu (64-bit).
R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.
Natural language support but running in an English locale.
R is a collaborative project with many contributors.
Type 'contributors()' for more information and

```
'citation()' on how to cite R or R packages in publications.
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
>
```

The ">" symbol indicates that now you are into the R environment (*see* **Note 2**)

3.1.3 Install the Necessary Libraries that Are Going to Be Used in the Analysis

Important: the ">" symbol indicates the R prompt and is not part of the command and you should not copy-paste these instructions into the promt line.

```
>install.packages(c("fossil","vegan","FactoClass","scatterplot3d","MASS","cluster","gplots","RcolorBrewer","pvclust","splines", "stats4", "survival", "mvtnorm", "modeltools", "coin","fields") ) (see Note 3)
>source("https://bioconductor.org/biocLite.R")
>biocLite("metagenomeSeq") (see Note 4)
```

3.1.4 Quit R

```
>q()
```

3.2 Install Python and the Packages Necessary for the Data Analysis

```
$ python -V
```

3.2.1 Check the Python Version

```
Python 3.6.0 :: Continuum Analytics, Inc.
```

3.2.2 If Python 2.7 Is Not Installed, Install Using This Command

```
$sudo apt-get install python2.7
```

3.2.3 Install the Packages

Install the following packages: rpy2, argparse, matplotlib, numpy, pandas, scikit-learn, scipy

The packages can be installed via pip install

```
$ python 2.7 pip install scipy # (see Note 5)
```

or via Anaconda

```
$conda install numpy # (see Note 6)
```

Repeat one of these actions for all the before-mentioned packages.

3.3 Download and Install the Biomarker Discovery Software

3.2.4 Download the LEfSe Software (See Note 7)

Download the LEfSe (Download repository) from this website: https://bitbucket.org/nsegata/lefse/downloads/

Move the software to the desired directory (etc., software LEfSe). For this example, the directory will be named software.

```
$ mkdir ~/software/
$ unzip ~/Downloads/nsegata-lefse-54694b4b0d9e.zip -d
~/software/ # (see Note 8)
```

3.4 Create the Working Directory and Download the Data

```
$ mkdir example
$ cd example
$ wget https://raw.githubusercontent.com/jorgevaz-cast/Viromic-diversity/master/simulated_viral_data.tsv
$ wget https://raw.githubusercontent.com/jorgevaz-cast/Viromic-diversity/master/viral_diversity.R
$ wget https://raw.githubusercontent.com/jorgevaz-cast/Viromic-diversity/master/supplementary_diversity_functions.R
```

3.5 Read Data and Normalization

3.5.1 Open the R Environment

```
$ R
```

3.5.2 Open All the Libraries That Are Going to Be Used

```
> library(fossil)
> library(vegan)
> library(cluster)
> library(gplots)
> library (RColorBrewer)
> library(pvclust)
> library(metagenomeSeq)
>library(RCurl)
> source("./supplementary_diversity_functions.R")
```

3.5.3 Read the Data Table of the Relative Abundances of the Viral "Species" (see Note 9)

Function to read the file that is in the table format

```
> abundance.table<-read.table(file="simulated_viral_data.tsv", header=T, row.names = 1 ,dec=".",sep="\t")
```

Function to visualize the data

```
> head(abundance.table)
```

3.5.4 Create the Metadata Table (see Note 10)

The metadata simulates two conditions, the "C" and the "H" condition

```
>metadata<-cbind(colnames(abundance.table),c(rep("H",100) , rep("C",100) ))
>rownames(metadata)<-colnames(abundance.table)
>metadata<-as.data.frame(metadata)
>colnames(metadata)<-c("samle_name","HC")
> head( metadata )
```

3.5.5 Exclude Rare Species That Could Be Taxonomic Binning Artifacts

The "zero.vec" contains the percentage of "0" for each sample

```
>zero.vec<-apply(abundance.table,1,function(x){ length(x[x==0])/length(x)} )
```

Exclude all those virus like particles whose viral like particles are above the 10% of the samples

```
> abundance.table.nr.sp<-abundance.table[zero.
vec<0.1,] # (see Note 11)
```

Compare the dimension of both matrices

```
>dim(abundance.table)

>dim(abundance.table.nr.sp)
```

3.5.6 Plot the Rarefaction Curves Using the Function "Rarecurve" from the Vegan Library

Estimates the minimum number of sequences (*see* Fig. 1)

```
>min.number.secs<-min(apply(abundance.table.
nr.sp,2,sum))
>#png(filename = " Fig1_rarefaction_curves.png",
width = 600, height = 600) # (see Note 12)
>pdf(file = "Fig1_rarefaction_curves.pdf")
> rarecurve(t(abundance.table.nr.sp), step = 200,
sample = min.number.secs, col = "blue", cex = 0.6)
# (see Note 13)
>dev.off() # (see Note 12)
```

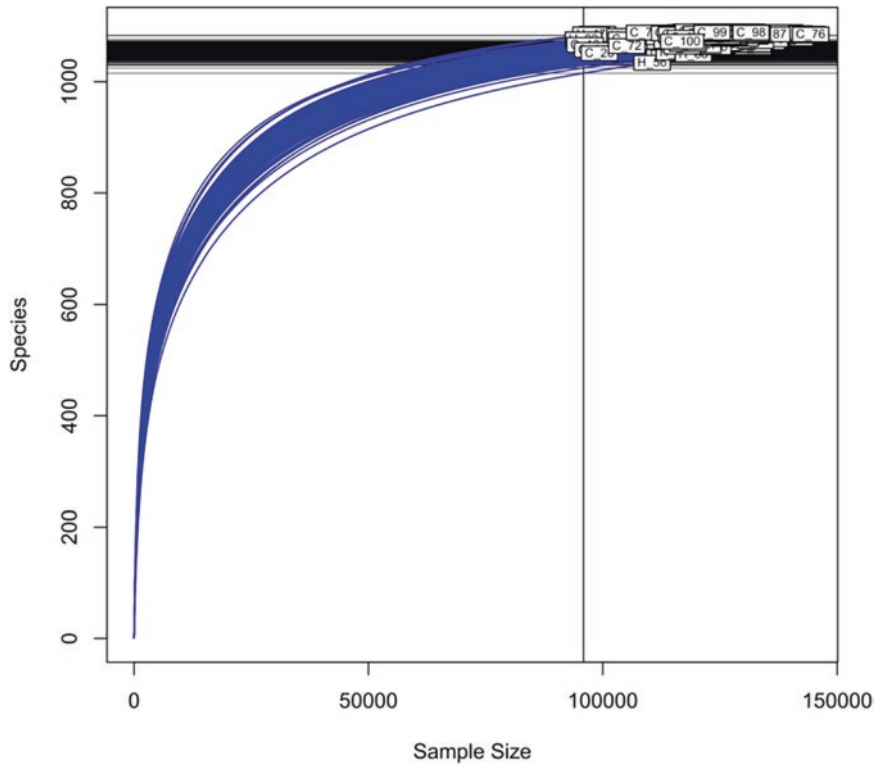

Fig. 1 Rarefaction curves. Rarefactions curves for the 200 samples. All samples reach the diversity plateau which indicates that the sequencing effort has saturated the diversity curves. The vertical line points to the sample with the lesser number of reads. Horizontal lines represent the rarefied species richness

3.5.7 Rarefaction: Performs a Rarefaction to the Sample with the Minimum Number of Reads

Perform the rarefaction

```
>viromic.rar<-rrarefy(t(abundance.table.nr.sp),
min(apply(abundance.table.nr.sp,2,sum))) # (see
Note 14)
```

Verify the output

```
>apply(viromic.rar,1,sum)
```

3.5.8 Standardization: Performs the CSS Standardization

It standardizes each one of the samples by its respective CSS value. The "cumNormMat" function is allocated into the metagenomeSeq library.

```
>viromic.CSS<-t(cumNormMat(as.matrix(abundance.table.
nr.sp))) #(see Note 15)
```

3.6 Alpha Diversity Analyses (See Note 16)

Alpha diversity indexes

```
>Shannon <- diversity(viromic.rar) # Shannon in-
dex, the function is coded in the "vegan" library
>Specnumber <- specnumber(viromic.rar) # Expected
number of species, the function is coded in the
"vegan" library
>Pielou <- pielou(viromic.rar) # Pielou index, the
function is coded in the supplementary functions
file "supplementary_diversity_functions.R"
>Chao1<-apply(viromic.rar,1, chao1) # Chao1 index,
the function is coded in the "fossil" library
>ACE<-apply(viromic.rar,1, ACE) # ACE index, the
function is coded in the "fossil" library
```

3.6.1 Plot the Boxplot of the Different Diversity Index. The Significance Between Condition Was Given by the Wilcoxon Test (See Fig. 2)

Divide the plotting window in three rows and two columns

```
>pdf("Fig2_Alpha_diversity_index_comparison.pdf")
>par(mfrow=c(3,2))
```

Calculate the p-value of the Wilcoxon test

```
>p.val<-round(wilcox.test(Shannon~metadata[,2])$p.
value, digits=3)
> boxplot(Shannon ~ metadata[,2],col=c("red","dark
green"), main="Shannon index",sub=
paste("WT p-val =",p.val), ylab=
"Shannon index") # Represents the data in a box-
plot # (see Note 17)
>legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=0.7,box.lty=0,
title="Legend") # Add the plot legend
>p.val<-round(wilcox.test(Specnum
ber~metadata[,2])$p.value, digits=3)
> boxplot(Specnumber ~ metadata[,2],col=c("
red","darkgreen"), main="Expected number of
species",sub=paste("WT p-val =",p.val), ylab="Num
species")
```

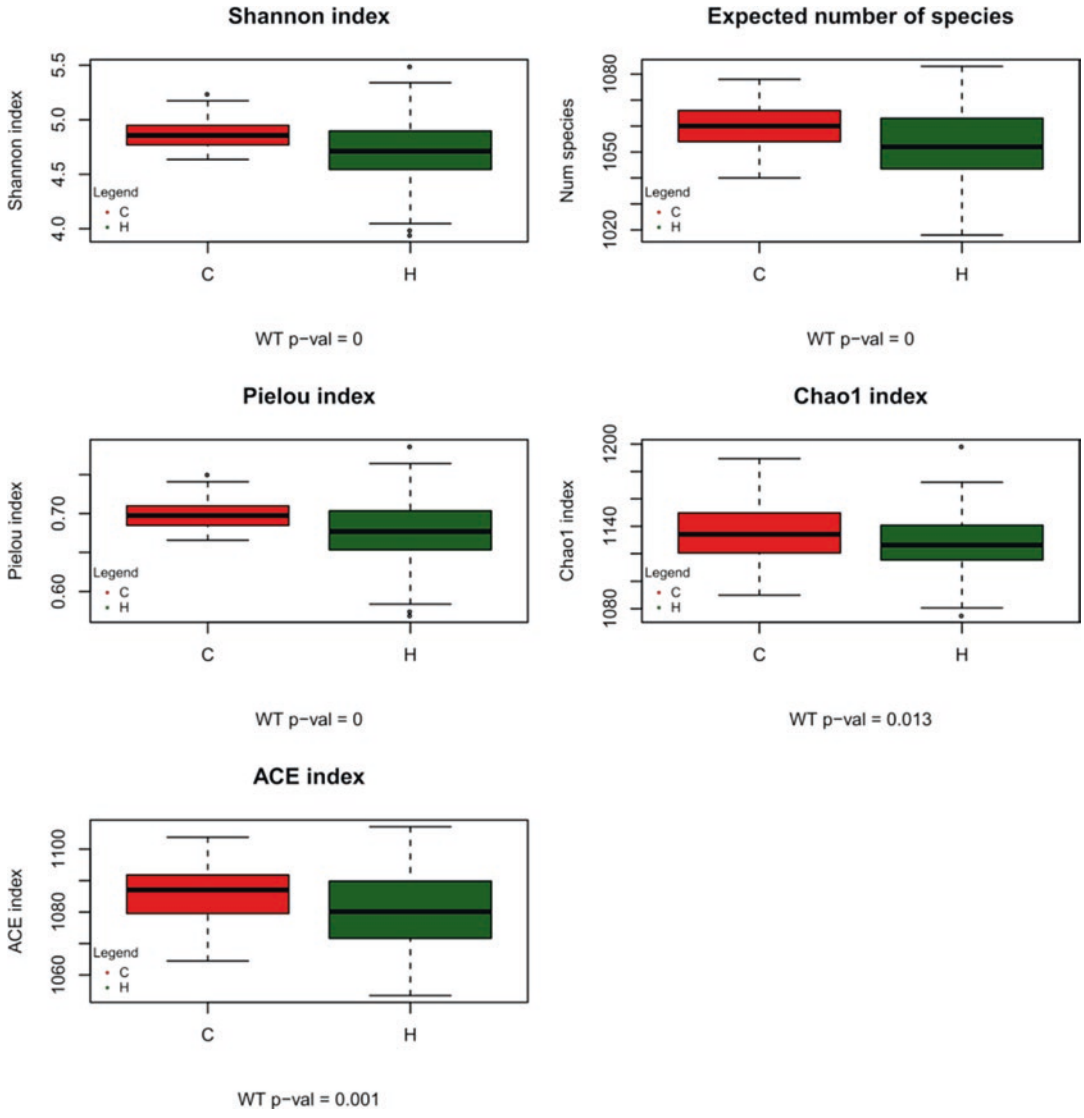

Fig. 2 Alpha diversity index comparison. Boxplot of the five different diversity indexes used in the analysis for both conditions H (green) and C (red). The statistical significance between both conditions is determined by the Wilcoxon test

```
>legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=0.7,box.lty=0,
title="Legend") # Add the plot legend

>p.val<-round(wilcox.test(Pielou~metadata[,2])$p.
value, digits=3)

> boxplot(Pielou ~ metadata[,2],col=c("red","dark
green"), main="Pielou index",sub=paste("WT p-val
=",p.val), ylab="Pielou index")

>legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=0.7,box.lty=0,
title="Legend") # Add the plot legend
```

```
>p.val<-round(wilcox.test(Chao1~metadata[,2])$p.
value, digits=3)
> boxplot(Chao1 ~ metadata[,2],col=c("red","dark
green"), main="Chao1 index",sub=paste("WT p-val
=",p.val), ylab="Chao1 index")
>legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=0.7,box.lty=0,
title="Legend") # Add the plot legend
>p.val<-round(wilcox.test(ACE~metadata[,2])$p.
value, digits=3)
> boxplot(ACE ~ metadata[,2],col=c("red","darkgre
en"), main="ACE index",sub=paste("WT p-val =",p.
val), ylab="ACE index")
>legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=0.7,box.lty=0,
title="Legend") # Add the plot legend
>dev.off()
```

3.7 Beta Diversity Analyses (See Note 18)

3.7.1 Calculate a Principal Coordinate Analysis (PCoA) Using the Bray-Curtis Index

```
>PCoA<-capscale(viromic.CSS ~ 1, metaMDS = TRUE,sqrt.
dist= TRUE) # (see Note 19)
>pdf("Fig3_Default_PCoA.pdf")
```

Plot the PCoA using default parameters (*see* Fig. 3)

```
> plot(PCoA)
>dev.off()
```

3.7.2 Other Dissimilarity Index to Calculate the Beta Diversity (see Note 20)

Dissimilarity index to calculate the beta diversity:

(a) Bray-Curtis

```
>rar.bray.dist<-vegdist(viromic.rar)
>CSS.bray.dist<-vegdist(viromic.CSS)
```

(b) Hellinger norm and Euclidean distance

```
>rar.hell.dist<-vegdist(decostand(viromic.rar,meth
od="hell"),method="euclidean")
>CSS.hell.dist<-vegdist(decostand(viromic.CSS,meth
od="hell"),method="euclidean")
```

(c) Jaccard distance

```
>viromic.rar.binary<-viromic.rar
>viromic.rar.binary[viromic.rar.binary>0]=1 # (see
Note 21)
>rar.jaccard.dist<-vegdist(viromic.rar.binary, method =
"jaccard", binary = TRUE)
```

3.7.3 Compare the Group Categories Using the Adonis Test (See Note 22)

```
>CH.cond<-metadata[rownames(as.matrix((rar.
bray.dist))),2]
```

Perform the ADONIS test

```
>adonis.rar.bray<-adonis( CSS.bray.dist ~ CH.cond )
```

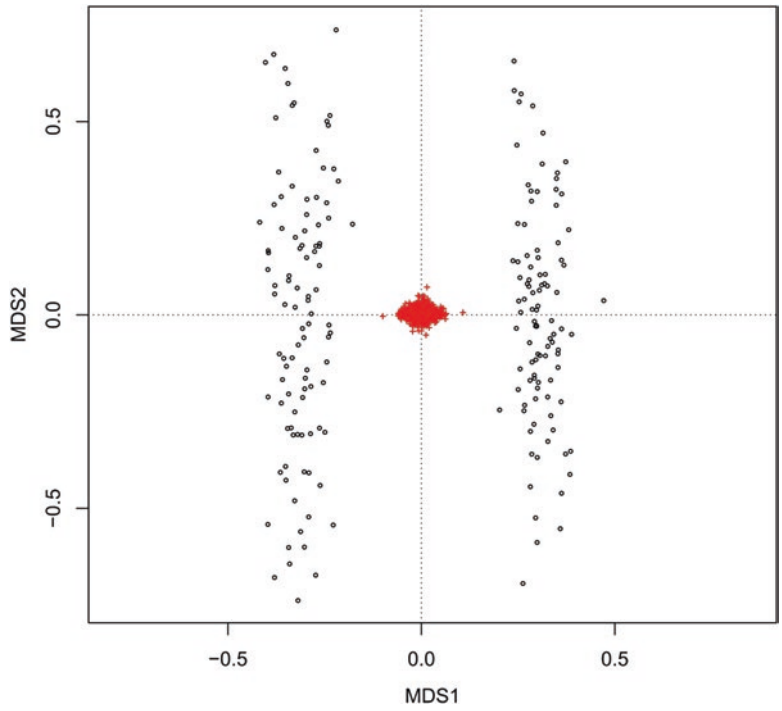

Fig. 3 Default PCoA. The circles represent the samples, the red addition symbols the species. The PCoA was estiamted using the Bray-Curtis index using the CSS normalized dataset. In this plot there is no axis labeling or the adition of color or labes to diferentiate the group samples

Visualize the adonis results

```
>adonis.rar.bray
```

Extract the ADONIS *p*-value

```
>adonis.rar.bray$aov.tab[ 6]$Pr[1]
```

3.7.4 Compare the Variance Homogeneity Between Groups (See Note 23)

Perform the betadisper test

```
>betadisper.res<-betadisper(CSS.bray.dist,
CH.cond, type = c("median"))
```

Look for the significance for the test

```
>permutest(betadisper.res, pairwise = FALSE, per-
mutations = 9999)
```

3.7.5 Represent the PCoA Using Different Function Approximations (See Note 24 and Fig. 4)

Function that decouples data from a vegan function

```
>PCoA.scores<-scores(PCoA)
```

Calculate the eigenvalues

```
>eig <- eigenvals(PCoA)
```

Estimate the proportion of variance explained

```
>eigVariance<- round(eig / sum(eig) * 100,digits=3)
```

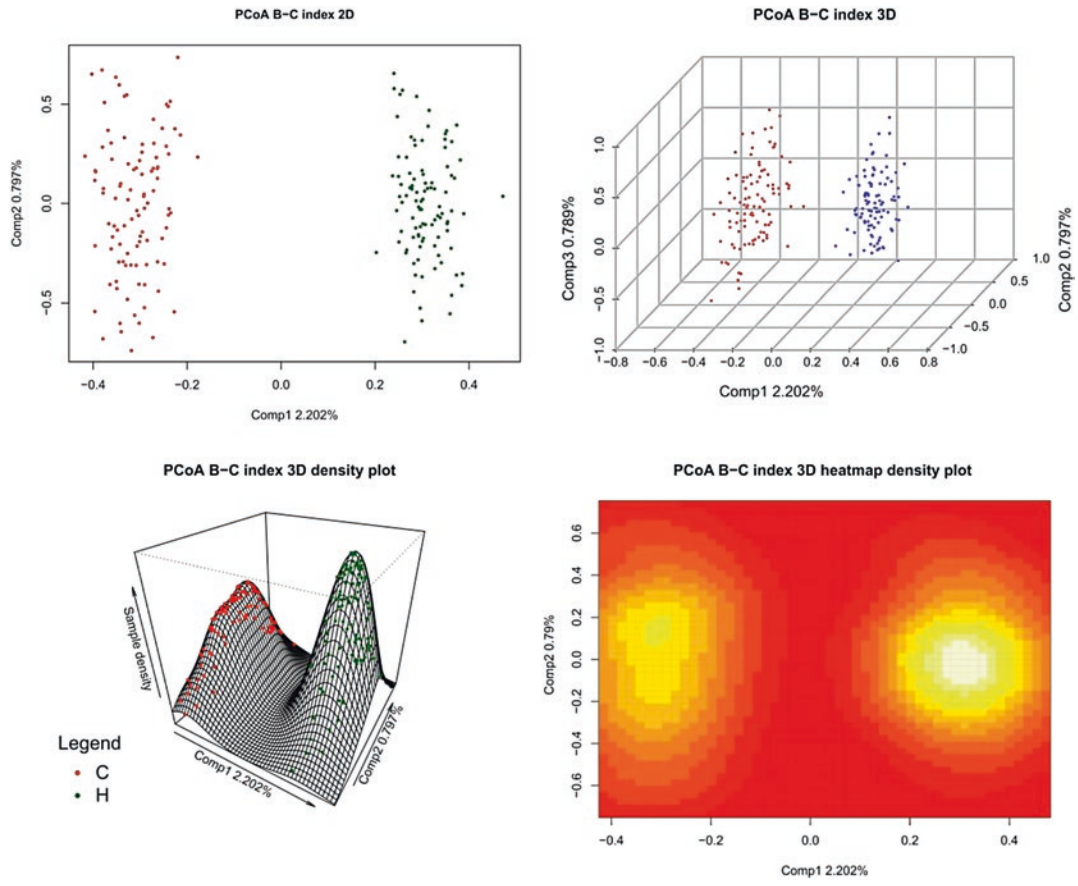

Fig. 4 Different PCoA representations. The top PCoA are based only on the representation of the first two and three components. The lower PCoA representations are based on the projection of the first two components and the sampling density surface. This surface is a continuous model of the distribution of the samples across the first two components and can be visualized as a density plot or as a heatmap. It is useful for the visualization group clustering and dispersion. The PCoA was estimated by the Bray-Curtis index using the CSS normalized dataset

Proportion of variance explained of the first component

```
>x.lab<-paste("Comp1",paste(eigVariance[1],"%",sep=""))
```

Proportion of variance explained of the second component

```
>y.lab<-paste("Comp2",paste(eigVariance[2],"%",sep=""))
```

Proportion of variance explained of the third component

```
>z.lab<-paste("Comp3",paste(eigVariance[3],"%",sep=""))
```

Divide the plot window into 2 rows and 2 columns

```
>pdf("Fig4_Different_PCoA_representations.pdf",width=12, height=10)
 >par(mfrow=c(2,2))
```

Plot the first two components of the PCoA

```
> plot(PCoA.scores$sites, cex = 1, col= c(rep("
red",100),rep("darkgreen",100)), pch = 16, cex.
main=1,cex.sub=0.7, xlab=x.lab, ylab=y.lab,
main="PCoA B-C index 2D")
```

Plot the first three components of the PCoA, using the libraries: FactoClass and scatterplot3d

```
>library("FactoClass")
```

```
>library("scatterplot3d")
```

Estimate the scores for the first three components

```
>PCoA.scores<-scores(PCoA,choices=c(1,2,3))
```

Plot the first three components

```
> scatterplot3d(PCoA.scores$sites[, 1:3], pch =
16, grid=FALSE, box=FALSE,color=c(rep("red",100),
rep("blue",100)), xlab=x.lab, ylab=y.lab,zlab=z.
lab,main="PCoA B-C index 3D")
```

Add the grid

```
>addgrids3d(PCoA.scores$sites[, 1:3], grid =
c("xy", "xz", "yz"))
```

Plot the first two components of the PCoA and the sample density using the MASS library

```
>library("MASS")
```

```
>x <- PCoA.scores$sites[, 1]
```

```
>y <- PCoA.scores$sites[, 2]
```

```
>z <- PCoA.scores$sites[, 3]
```

Estimate the kernel density surface (*see* **Note 25**).

```
>dens3d <- kde2d(x, y,n = 50)
```

```
> res<-persp(dens3d, box=TRUE, theta=30,phi=30,
xlab=x.lab, ylab=y.lab, zlab="Sample density",
main="PCoA B-C index 3D density plot")
```

Library to plot the sample points into the density plot

```
>library(fields)
```

Create a model to interpolate the values for the first two components into the density surface (*see* **Note 25**)

```
>density.z<- interp.surface( dens3d, cbind(x,y))
```

```
>points(trans3d(x, y, density.z, pmat = res), col
= c(rep("red",100),rep("darkgreen",100)), pch =
16, cex=1)
```

Add the plot legend

```
legend("bottomleft", legend=c("C", "H"), pch=16,
col=c("red", "darkgreen"), cex=1.5,box.lty=0,
title="Legend") #
```

Heat density plot

```
>image(dens3d, xlab=x.lab, ylab=y.lab, main="PCoA
B-C index 3D heatmap density plot")
```

Close the plot

```
>dev.off()
```

3.7.6 Comparison of the Efficiency of Different Dissimilarity Matrices (See Note 26 and Fig. 5)

```
>distance_matrix_comparison(viromic.CSS,viromic.rar,metadata) # (see Note 27)
```

Sum variance three comp ADONIS *p*-value betadisper *p*-value

CSS bray	9.501	0.001	1e-04
CSS hell	26.002	0.001	1e-04
CSS jac	19.515	1.000	1e-04
rar bray	8.998	0.001	1e-04
rar hell	25.821	0.001	1e-04
rar jac	11.811	0.001	1e-04

The results from Fig. 5 and the table show a clear separation between classes C and H. This is independent of the normalization method. It is also observed that the Hellinger distance is the one that greater variation explains for both methods of normalization, while the Jaccard tends to agglomerate all the samples in the same area for the case of the normalization by CSS.

3.7.7 Perform a Nonmetric Multidimensional Scaling Analysis (NMDS) (See Fig. 6)

The Bray-Curtis and the Hellinger distance were the ones that show the best performance. Now we will use these distances to make an NMDS. The NMDS is a nonparametric method to perform an ordination analysis using as input a distance matrix. The NMDS attempt to represent the whole variance of the distance matrix into a predefined number of dimensions (*see* **Note 28**)

```
>pdf("Fig6_NMDS_comparison.pdf",width=17)
>par(mfrow=c(1,3))
```

Create NMDS from using the CSS normalized dataset and the Bray-Curtis index

```
>nmds.CSS.bray<-metaMDS(CSS.bray.dist,
trymax=2000,pc=T) # (see Note 29)
>plot(nmds.CSS.bray$points, cex = 1.5,
col= c(rep("darkgreen",100),rep("r
ed",100)), sub=paste("Stress",round(nmds.CSS.
bray$stress,digits=3),sep="="), cex.main=1.7,cex.
sub=1,pch=16, main=" Non-metric multidimensional
scaling B-C index")
```

Create NMDS from using the CSS normalized dataset and the Hellinger distance

```
>CSS.hell.dist<-vegdist(decostand(viromic.CSS,meth
od="hell"),method="euclidean")
>nmds.CSS.hell<-metaMDS(CSS.hell.dist,
trymax=2000,pc=T)
>plot(nmds.CSS.hell$points, cex = 1.5,
col= c(rep("darkgreen",100),rep("r
ed",100)), sub=paste("Stress",round(nmds.CSS.
hell$stress,digits=3),sep="="), cex.main=1.7,cex.
```

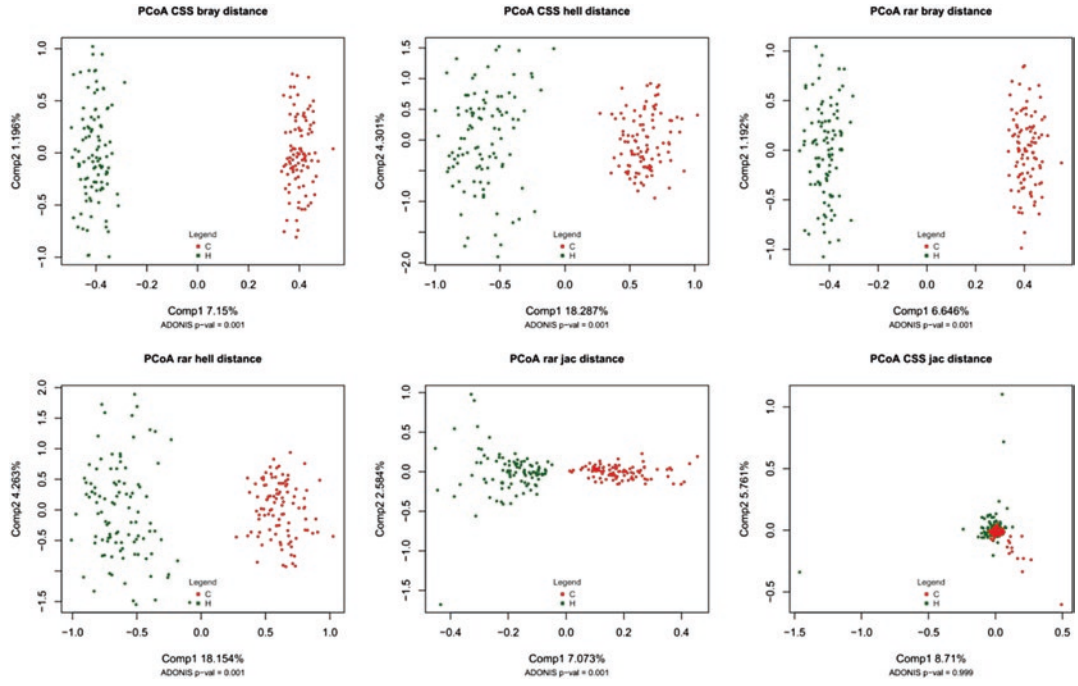

Fig. 5 PCoA using different dissimilarity matrices. Six different PCoA analyses using different distance matrix and normalization methods. Those included the two normalization methods, CSS and rarefaction, and three different dissimilarity indexes: The Bray-Curtis, the Hellinger and the Jaccard. For all the methods the ADONIS group *p*-value was calculated

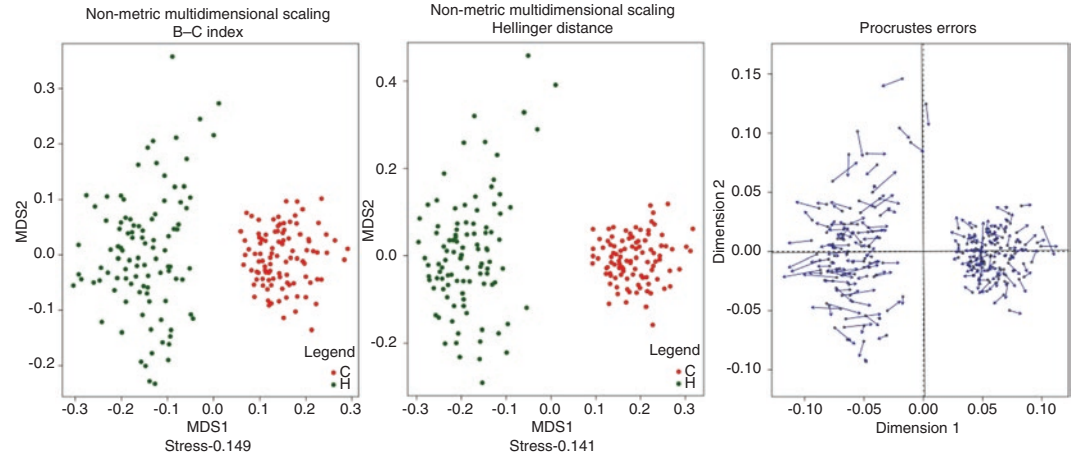

Fig. 6 NMDS comparison. Comparison of two NMDS scaling analyses using the Bray-Curtis and the Hellinger distance. Both NMDS show similar results and present a similar Stress. This indicates that both solutions are suitable for working and that the distribution of the data does not suffer from methodological artifacts. Moreover, the Procrustes analysis shows a statistical congruence between the two-ordination analysis

```
sub=1,pch=16, main=" Non-metric multidimensional
scaling Hellinger distance")
```

3.7.8 Perform a Procrustes Test (See Fig. 6)

To verify if the two ordination analyses are significantly similar we can employ the protest function, this test verifies if the cluster configurations are congruent

```
>prtest<-protest(nmds.CSS.bray,nmds.CSS.hell)
> prtest$signif
>plot(prtest)
```

[1] 0.001

In this case, the *p*-value is below 0.01, this indicates that they are significantly similar.

```
>dev.off()
```

3.8 Clustering Analysis

In Clustering analysis, it is the task of grouping a set of objects in such a way that members of the same group (called a cluster) are more similar, in some sense or another. It is widely used in metagenomic analysis to reveal the structure of subpopulations within environment/condition/groups.

3.8.1 Partitioning Around Medoids (PAM) Algorithm (See Fig. 7)

The PAM algorithm finds a cluster configuration in which the dataset could be divided optimally into K groups, where K is the desired number of groups (for a deeper compression of the algorithm check (https://doi.org/10.1007/s10852-005-9022-1 and https://doi.org/10.1016/j.eswa.2008.01.039)). This clustering method has been used to find the enterotypes [17].

Here, the cluster configuration of the groups, divided from 2 to 20, will be calculated using the PAM algorithm and will be weighted using the silhouette index (*see* **Note 30**)

```
>obs.silhouette.temp=NULL
>nclusters.silhouette=NULL
>pdf("Fig7_NMDS_PAM_clustering.pdf")
```

Estimate the value of the silhouette index for clusters configurations from 2 to 20

```
>num.clusters<-(length(colnames(as.matrix(CSS.
bray.dist)))-180)
>for (k in 1:num.clusters) {
> if (k==1) {
> obs.silhouette.temp[k]=NA
> } else {
> data.cluster.temp<-as.vector(pam(CSS.bray.dist,
k, diss=TRUE)$clustering)
> obs.silhouette.temp<-mean(silhouette(data.clus-
ter.temp, CSS.bray.dist)[,3])
> nclusters.silhouette[k]=obs.silhouette.temp
> }
>}
```

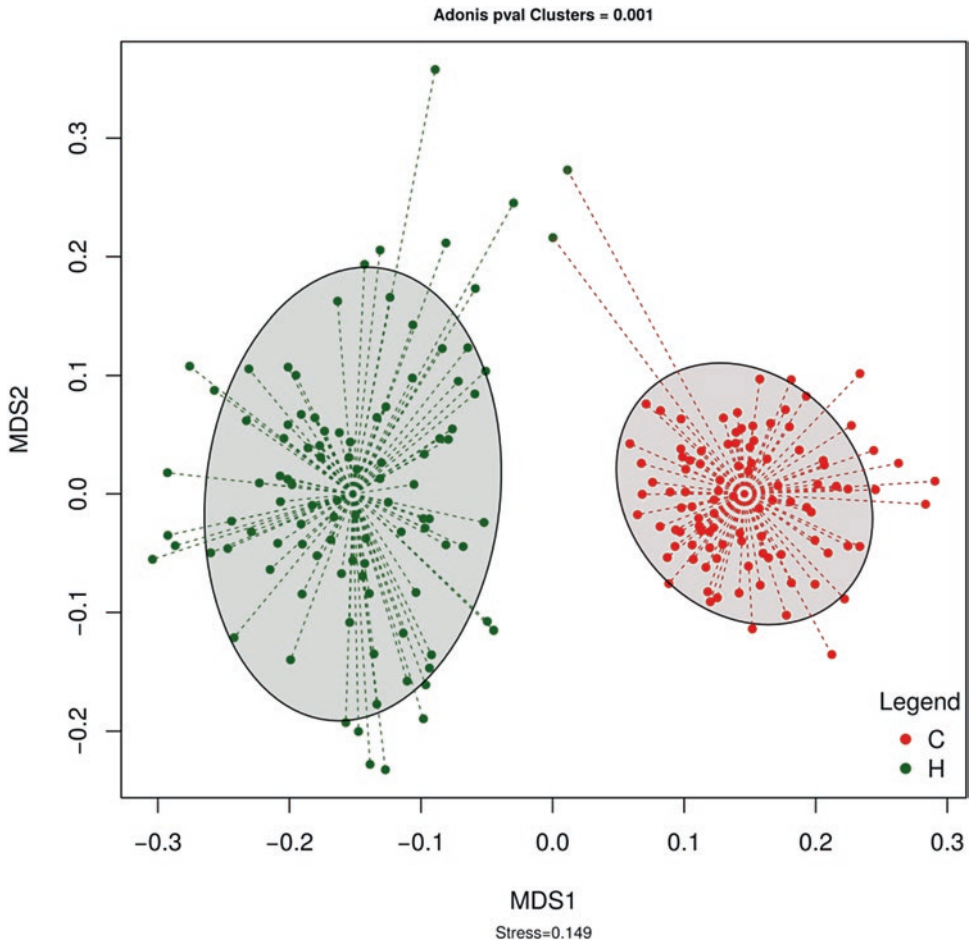

Fig. 7 NMDS PAM clustering. NMDS scaling analysis estimated using the Bray-Curtis index. The ellipses represent the cluster configuration estimated by the PAM algorithm. The ellipses represent the 80% of the group variance. The cluster configuration was the one that maximizes the Silhouette index and was validated by the ADONIS test (*p*-value = 0.001)

Obtain the cluster configuration that maxims the Silhoutte index

```
>silhouette.num.clusters<-
grep("TRUE",max(nclusters.silhouette, na.rm =
T)==nclusters.silhouette)
```

Perform the cluster configuration

```
data.cluster<-as.vector(pam(CSS.bray.dist, silhou-
ette.num.clusters, diss=TRUE)$clustering)
```

Validate the cluster configuration using the ADONIS test

```
>adonis.clusters<- adonis( CSS.bray.dist ~ data.
cluster )$aov.tab[ 6]$Pr[1]
```

```
>clust.title<-paste("Adonis pval
Clusters",round(adonis.clusters,digits=3),sep=" =
")
```

Plot the cluster configuration into the NMDS analysis representing the clusters using the functions "ordiellipse" and "ordispider" available in the vegan library

```
>plot(nmds.CSS.bray$points, cex = 1, pch=20,
col= c(rep("darkgreen",100),rep("red",100)),
main=clust.title, sub=paste("Stress",round(nmds.
CSS.bray$stress,digits=3),sep="="), cex.
main=0.7,cex.sub=0.7)
>treat<-data.cluster
>for(i in unique(treat)) {
> print(i)
> if(i==1){
> color="darkgreen"
> }
> if(i==2){
> color="red"
> }
```

Ellipses confidence is set to the 80%.

```
>ordiellipse(nmds.CSS.bray$point[grep(i,treat),],
groups=treat[treat==i],col=color,draw="polygon",al
pha=40, conf=0.80,show.groups=i)
> ordispider(nmds.CSS.bray$point[grep(i,treat),],
groups=treat[treat==i],col=color,lty=2,lwd=0.7)
>}
>dev.off()
```

3.8.2 Hierarchical Clustering and Heatmaps

The hierarchical clustering seeks to find a hierarchy of clusters. This method is useful in combination with the hetmaps to visualize a detailed cluster configuration and observe the relative abundance of each species in each cluster.

Estimate the hierarchical clustering

```
>hcluster.col <- hclust(CSS.bray.dist, method
="ward.D")
```

Only plot the most abundant species (*see* **Note 31**)

```
>prop.viromic.CSS<-prop.table(viromic.CSS,1) * 100
>apply(prop.viromic.CSS,1,sum)
>vec.above<-apply(prop.viromic.CSS,2,function(x)
{mean(x)>0.5})
>virus.aboce<-names(vec.above[grep(TRUE,vec.
above)])
```

Estimate the row (species) clustering

```
distance.row<-vegdist(t(viromic.CSS[,virus.ab-
oce]))
>hcluster.row <- hclust(distance.row, method
="ward.D")
```

Add the sample condition colors

```
>ColSideColors<-c(rep("red",100),rep("blue",100))
```

Define the colors for the heatmap, function "colorRampPalette" is in the RcolorBrewer library

```
>heatcolors <- colorRampPalette(brewer.pal(9,
"Spectral"))
```

Abbreviate the taxonomic annotation

```
>plot.heatmap.matrix<-t(prop.viromic.CSS)
>rownames(plot.heatmap.matrix)<-
sapply(rownames(plot.heatmap.matrix),function(x)
{unlist(strsplit(x, ";"))[5]})
```

Plot the results

```
>pdf("Fig8_Heatmap_hclust.pdf ") # (see Note 32)
```

Calculate and represent the heatmap, the heatmap.2 function is in the gplot library (Fig. 8)

```
>heatmap.2( plot.heatmap.matrix,
     >#main = paste( "test"),
     >col=heatcolors,
     >trace="none",
     >margins =c(5,7),
     >ColSideColors=ColSideColors,
     >colCol = ColSideColors,
     >dendrogram="both",
     >Rowv = as.dendrogram(hcluster.row),
     >Colv = as.dendrogram(hcluster.col),
     >key.xlab = "% Relative abundance",
     >cexRow =1,
     >cexCol = 0.8,
     >na.rm = TRUE )
>dev.off()
```

3.8.3 Add a Statistical Significance to the Hierarchical Clustering Partitions

The function "pvclust," library pvclust performs a bootstrapping to add statistical significance to the clustering partitions. For more information visit the http://stat.sys.i.kyoto-u.ac.jp/prog/pvclust/ web site (Fig. 9).

```
>pv.hclust <- pvclust(t(viromic.CSS), method.
dist="cor", method.hclust="ward.D",parallel=T,
nboot=1000) # (see Note 33)
>pdf("Fig9_Bootstrapping_HC.pdf",width=24)
>plot(pv.hclust)
```

Highlight with a rectangle those clusters with a significant p-value

```
>pvrect(pv.hclust, alpha=0.95)
>dev.off()
```

3.9 Biomarker Discovery

The determination of different biomarkers (species), which are related to each condition, is not an easy task, the use of wrong

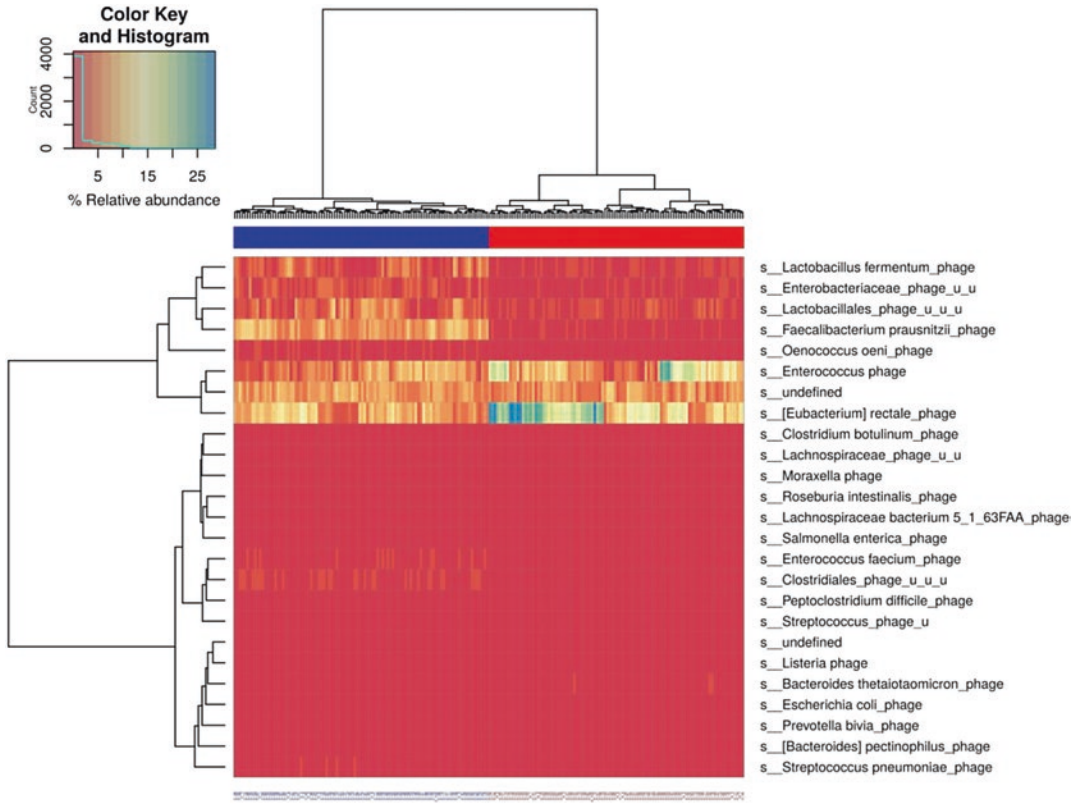

Fig. 8 Heatmap. Hierarchical clustering analysis of the complete dataset represented as the top dendrogram. The heatmap only shows those species whose relative abundance is above 0.5%. The color gradient represents the relative abundance of the spices. The column colors represent the group of the samples

statistical test can generate false positives results or disparage species that may actually be biomarkers. In this section, two statistical packages that have been commonly used in metagenomic studies will be presented. All these use multivariate models and make different corrections to avoid the FDR.

3.9.1 Compute Differential Abundance Analysis Using a Zero-Inflated Log-Normal Model

This approximation is integrated into the metagenoSeq library, function "fitFeatureModel." It is strongly recommended to check the tutorial https://www.bioconductor.org/packages/devel/bioc/vignettes/metagenomeSeq/inst/doc/metagenomeSeq.pdf and the published paper of Paulson et al. [18]. The method fit a zero-inflated log-normal model for each viral specie. Then, perform a log Fold change test between the two conditions and return the *p*-value and the adjusted *p*-value. This method can deal with sparsity data which is common in metagenomic viral data.

Create an MRexperiment object (*see* **Note 34**)

```
>Metadata = AnnotatedDataFrame(metadata)
>taxVir<-sapply(rownames(t(viromic.
CSS)),function(x){unlist(strsplit(x, ";"))})
```

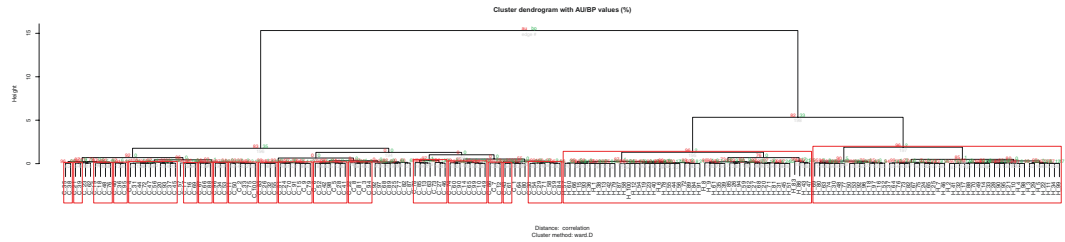

Fig. 9 Bootstrapping hierarchical clustering. The rectangles represent those partitions whose *p*-value is <0.05. The plot shows two bootstrap values, the approximate unbiased (AU) represented in red and the BP (Bootstrap Probability) in green. The authors recommend the use of the AU values

```
>taxVir<-t(taxVir)
>taxVir<-cbind(rownames(taxVir),taxVir)
>colnames(taxVir)=c("Taxonomy","superkingdom","ord
er","family","genus","species")
>taxVir=data.frame(taxVir)
>OTUdata = AnnotatedDataFrame(taxVir)
>obj= newMRexperiment(as.matrix(abundance.table.nr
.sp),phenoData=Metadata,featureData=OTUdata)
```

Normalize data

```
>obj <-cumNorm(obj, p = 0.5)
>pd <-pData(obj)
# Create the model matrix
>mod <-model.matrix(~1 + HC, data = pd)
```

Compute de model using the function fitFeatureModel

```
>vir.mod =fitFeatureModel(obj, mod)
# Extract the results
>df.mod.res<-MRcoefs(vir.mod, number = dim(taxVir)
[1])
 # Select those coefficients which are statisti-
cally significant
>df.mod.sigVirtax<-subset(df.mod.
res,adjPvalues<0.05)
```

Show those specie whose relative abundance is statistically different

```
> df.mod.sigVirtax
```

3.9.2 Linear Discriminant Analysis (LDA) Effect Size (LEfSe) Method

LEfSe is a very popular tool used in metagenomics, it is a three-step algorithm that first perfoms a Kruskal Wallis test between classes, then a Wilcoxon test between all the parwise subclasses and finally a linear discrimiannt analysis. Those species/genes that possess a LDA score above a certain cutoff are selected as biomarkers [19]. LEfSe has an online version in which it details in a very clear and precise way the necessary steps to carry out this analysis https://huttenhower.sph.harvard.edu/galaxy/. It is strongly

recommend to use this version unless the server is down, the dataset is very large, or you have to make multiple comparisons.

Create the LEfSe infile in R

```
>LEfSe.infile<-t(viromic.CSS)
>rownames(LEfSe.infile)<-
gsub(";","|",rownames(LEfSe.infile)) # (see Note 36)
>Classes<-as.character(metadata[colnames(LEfSe.
infile),2])
>samples<-colnames(LEfSe.infile)
>LEfSe.infile<-rbind(Classes,samples,LEfSe.infile)
>write.table(LEfSe.infile, "LEfSe.infile" ,col.
names=F,row.names = TRUE,quote=FALSE,sep = "\t")
```

Exit from R

```
q()
```

Perform an analysis with LEfSe (*see* **Note 7**)

```
$LEfSe_dir=~/software/nsegata-lefse-e3cabe93a0d1
```

Define the infiles

```
$ mkdir LEfSe
$ mv LEfSe.infile ./LEfSe
$cd LEfSe
# This is the inflie created in R
$infile=LEfSe.infile
# The current directory
$indir=`pwd`
$outir=$indir/out_lefse_LDA
$mkdir $outir
```

First row contains the class name for each sample "C" or "H"

```
$class=1
# No subclasses
$subclass=0
# Samples
$samples=2
```

Print the help and program options

```
$ python2.7 $LEfSe_dir/format_input.py -h
# format_input.py, create the LEfSe infile
# -o set the normalization value (default -1.0 meaning
no normalization)
$ python2.7 $LEfSe_dir/format_input.py
$indir/$infile $outir/$infile.format_input -c $class
-s $subclass -u $samples -o 1000000
```

Print the help and program options

```
$ python2.7 $LEfSe_dir/run_lefse.py -h
```

Perform the LEfSe analysis

```
# -w set the alpha value for the Wilcoxon test
(default 0.05). In this case is going to be
0.00001

# -l set the threshold on the absolute value of
the logarithmic LDA score (default 2.0). Change
value to 3

# -s 1 set the multiple testing correction options

$ python2.7 $LEfSe_dir/run_lefse.py $outir/$infile.
format_input -w 0.00001 -s 1 -l 3 $outir/$infile.res
```

Plot the barplot in a pdf format with a resolution of 600 dpi (*see* **Notes 35** and **36**)

```
$python2.7 $LEfSe_dir/plot_res.py $outir/$infile.
res $outir/$infile.pdf --dpi 600 --format pdf
```

Plot the hierarchical cladogram in a pdf format with a resolution of 600 dpi (*see* **Notes 35** and **36**)

```
$ python2.7 $LEfSe_dir/plot_cladogram.py
$outir/$infile.res $outir/$infile.cladogram.pdf
--dpi 600 --format pdf
```

4 Notes

1. This repository is for Ubuntu 16.04, you should specify your proper Ubuntu R repository. This link https://cran.r-project.org/bin/linux/ubuntu/README.html explains more in detail how to install R.
2. To obtain a better intuition of the R language you can visit the https://www.statmethods.net/r-tutorial/index.html web site.
3. The function "install.packages" will prompt a screen for choosing the country-server from which you would like to download the packages. The installation of the packages can take several minutes depending on the speed of the processor with which you are working. In addition, it may request that programs or libraries be installed in R.
4. The package "metagenomeSeq" belongs to the bioconductor repositories, this is the reason why this package has a different installation.
5. In case that the default python is not python 2.7, you should type the python2.7 before any python script or command, this is to run the scripts under this version. To better understand the pip installation, see this website https://packaging.python.org/tutorials/installing-packages/.
6. To better understand the Anaconda (https://docs.anaconda.com/anaconda/install/) or minconda (https://conda.io/docs/user-guide/install/index.html) setup.

7. There is an easy-to-use web server of LEfSe discovery tool if the number of analyses to be done is small and if the dataset is not very large, this option should be used https://huttenhower.sph.harvard.edu/galaxy/.
8. The directory will contain a sh script (nsegata-lefse-54694b4b0d9e/example/run.sh) which is a clear example of how to use LEfSe.
9. The data come from the simulation of a dataset based on the work of Pérez-Brocal et al. [20]. Two conditions were simulated, the "H" condition which comes from a direct simulation of the real samples and the "C" condition, which comes from a mixture of the real samples and a simulated community which its species abundances follow a power law distribution. About 20 percent of the species were increased in one of two conditions either by adding one random number to each species-sample or by multiplying each species-sample by a random number. The final dataset contains 200 samples and 2299 virus-like species. This can be visualized in R using the command dim() ie:

   >dim(abundance.table)

   [1] 2299 200.
10. The metadata table is created by simply associating each "H" or "C" sample with its corresponding group label.
11. Ordination analyses tend to present problems when the data has a large number of zeros. This type of data is called sparse data and there are statistical methods to deal with it. In the case of viral metagenomes, the high number of zeros can render inoperable even the methods considered to deal with wide data. For this reason, it is recommended to eliminate those viral species that are not present in a certain number of samples. In this case, it will be the 10%.
12. R contains different ways to save the plots in different formats, here it is presented the plot saved in the pdf format, function "pdf" and into the png format using the function "png" this later is commented. This is denoted by adding the "#" symbol before the function. The function dev.off() will close the plot. For this particular example, the rarefaction curve will be saved in the pdf file named Fig1_rarefaction_curves.pdf.
13. Rarefactions curves allow us to quantify the species richness for a given number of samples. These curves are used to determine if the sample sequencing effort is enough to saturate the species diversity. The saturation in the number of species is observed if the rarefaction curve reaches the plateau, this indicates that although the number of reads increases the number of new species remains almost equal. The rarefaction curves

estimation can take several minutes depending on the number of samples and the sequencing effort.

14. The rarefaction to the minimum number of samples is a common technique for data standardization and has obtained good results in simulated datasets [21]. The sample-rarefaction helps to avoid to find statistical biases given the sequencing effort. It is important to that if the sample with the minimum number of reads is very low it is better to remove it from the analysis and take as the reference a sample that reaches the plateau in the rarefactions curves.
15. The Cumulative sum scaling (CSS) normalization is an effective method for sample normalization [18, 20]. The method is based on a quantile normalization for each of the samples of the dataset. The method is specified in the work of Paulson et al. [18] and in the metagenoSeq documentation https://www.bioconductor.org/packages/devel/bioc/vignettes/metagenomeSeq/inst/doc/metagenomeSeq.pdf.
16. All the analyses are based on the rarefied table of species relative-abundances. However, there are tools that are able to estimate the Shannon index based on the spectrum of contigs (see PHACCS [22]). These types of tools have been widely used in the study of viromes and its use is recommended. Since for the present analysis the read data of metagenomes are lacking, this type of approximations cannot be made. The diversity indices used in this analysis were the following:

    Specnumber: is the expected number of species.

    Shannon: quantify how evenly are the species distributed in the sample (The relation between the number of species and the number of individuals in a random sample of animal population).

    Pielou: Quantify the evenness into the species abundances of the community [7].

    Chao1: Richness estimator that adds a correction factor to quantify low abundance species singletons (species observed only once).

    ACE: Similar to the Chao1 index but the correction is for species with fewer than ten individuals.
17. There are several parameters to improve or modify the boxplot, here we are modifying the color of the box ("col" parameter), the axis names ("ylab" and "xlab"), the name of the plot ("main"), and the sub name ("sub"). To see all the modifiable parameters of each function, you can use the command:

    ```
    >?Function_name
    ```

this will return all the parameters and the description of the function without the need to search the Internet. To exit this help, type q and enter.

18. The beta diversity measures the differences in species composition between habitats. Normally, a dissimilarity index or distance is used to study such differences.

19. The PCoA (also known as Multidimensional scaling) is a method to visualize similarities into a dissimilarity matrix. The method transforms the distance matrix into a set of new coordinates that maximize the variance of the dataset. Then, the set of axes is returned, ordered according to the amount of variance explained. So, the first components are the ones that explain more variance. The vegan function "capscale" is used to calculate a PCoA by first transforming the normalized abundance species-table in a dissimilarity matrix, by default using the Bray-Curtis index. The function needs a formula to perform the analysis, in the case in which informative metadata is available, then it can be incorporated into the formula (see https://www.rdocumentation.org/packages/vegan/versions/1.11-0/topics/capscale). In case you only want to calculate the PCoA of the species-table use this command:

    capscale(data_matrix ~ 1, metaMDS = TRUE,sqrt.dist = TRUE)

20. In ecological analysis, different distances and indexes can be used to perform a PCoA others [8, 9]. The vegan library includes several of these useful distances. In the present example the Bray-Curtis index, the Jaccard distance, and the Hellinger distance are going to be used; this is because they have been used in metagenomic analysis and in viromics [23, 24].

    Bray-Curtis: based to quantify compositional dissimilarity between two samples. It is the ratio of the sum of the lesser values between two species, multiplied by two, divided by the sum of both species.

    Hellinger: the Hellinger distance is the normalization of the data matrix by the Hellinger normalization and then calculate the Euclidean distance between the two samples.

    vegdist(decostand(viromic.CSS,method="hell"),method="euclidean")

    Jaccard: distance estimated based on the presence/absence of the species within the dataset. This distance is thought to be used with binary data (0 and 1).

21. The Jaccard distance is for binary data, then the abundance matrix must be transformed into a presence/absence matrix.

22. The analysis of variance using distance matrices (ADONIS) is a nonparametric Manova suited for data retrieved from dissimilarity index or distances. The statistical power of the test relays into the number of permutations (at least 999). In the present example, the function is used to test if the samples of group H are significantly more distant from those of group C than of themselves. For more information about the ADONIS test, check the [25, 26]. The anosim function also can be used for the same propose http://cc.oulu.fi/~jarioksa/softhelp/vegan/html/anosim.html.
23. The betadisper test if the variance between the groups is similar. If it is not true, this can generate false positives in the ADONIS test, to remedy this, complementary clustering tests can be performed. The significance of the test is given by a permutation test.
24. The PCoA can be improved using different approximations, here the PCoA in 2D, 3D, and the density plot will be represented.
25. A density plot was used to determine the distribution of the samples along the first two components. This method models a 2D density surface based on the sample distribution of the first two components. It can be visualized as a surface ("persp" function) or as a heatmap ("image" function).
26. This function compares different indexes and distances to see which of them explains a greater variation in the data. The comparison includes the distances of Jaccard, Bray-Curtis, and Hellinger. These distances were estimated from the data of the normalized species abundance table and the rarefaction table. The function returns a plot with six different PCoA and a table with the variance explained by the first three components, the ADONIS *p*-value and, the betadisper *p*-value.
27. This function is found in the supplementary library of data "supplementary_diversity_functions". This performs different predetermined comparisons.
28. The NMDS is a powerful statistical method to represent the distance matrix in a low-dimensional space. Before the analysis you have to define the number of dimensions (axis) in which you would like to represent the NMDS (normally two or three dimensions are fine). Then the algorithm tries to find an object configuration that best fits the N-dimensional ordinations space [27] performing multiple iterations that minimize the Stress value. This value represents the differences between the reduced dimensions and the complete dataset. Stress values <0.2 are considered good.
29. The NMDS could be very computing depending on the dataset and sometimes could not reach the optimal solution; for this reason, a fair number of iterations must be done in the case

that the algorithm does not find the optimal solution. In this example, 2000 iterations were done.

30. The Silhouette is a method to validate the consistency of the clusters within the data. The method measures how well each object lies within each cluster.
31. Although the clustering is performed with the complete dataset only the most abundant at species are represented, this is because the output could be very large.
32. Sometimes the output of the function "heatmap.2" library gplots could be truncated; this could be fixed using programs as inkscape (https://inkscape.org/en/). In this program, it is very easy to adjust the size of the drawing canvas.
33. Depending on the number of samples and species the pvclust function can take several hours; it is recommended to use it with the parallel option to parallelize the bootstraps.
34. The MRexperiment object is the data structure that the package metagenoSeq uses for its analyses (check https://www.bioconductor.org/packages/devel/bioc/vignettes/metagenomeSeq/inst/doc/metagenomeSeq.pdf).
35. LEfSe has very nice plotting options, you can specify the full taxonomy of the species separated by "|" ie:
    "k__Viruses|o__Herpesvirales|f__Herpesviridae|g__Cytomegalovirus|s__Cytomegalovirus_u"
    to allow the program to detect which taxonomic level has more influence for sampling segregation.
36. The figures of the LEfSe are in the https://github.com/jorgevazcast/Viromic-diversity/tree/master/LEfSe/out_lefse_LDA.

References

1. Council NR (1999) Perspectives on biodiversity: valuing its role in an everchanging world. The National Academies Press, Washington
2. Whittaker RH (1972) Evolution and measurement of species diversity. Taxon 21:213–251
3. Shannon C (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423
4. Tuomisto H (2010) A consistent terminology for quantifying species diversity? Yes, it does exist. Oecologia 164:853–860
5. Chao A (1984) Non-parametric estimation of the number of classes in a population. Scand J Stat 1:265–270
6. Chao A, Lee SM (1992) Estimating the number of classes via sample coverage. J Am Stat Assoc:210–217
7. Mulder CPH, Bazeley-White E, Dimitrakopoulos PG et al (2004) Species evenness and productivity in experimental plant communities. Oikos 107:50–63
8. Whittaker RH (1960) Vegetation of the siskiyou mountains, Oregon and California. Ecol Monogr 30:279–338
9. Faith DP, Minchin PR, Belbin L (1987) Compositional dissimilarity as a robust measure of ecological distance. Vegetatio 69:57–68
10. Caporaso J, Kuczynski J, Stombaugh J et al (2010) QIIME allows analysis of high-throughput community sequencing data. Nat Methods 7:335–336
11. Schloss PD, Westcott SL, Ryabin T et al (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. Appl Environ Microbiol 75:7537–7541

12. Philosof A, Yutin N, Flores-Uribe J et al (2017) Novel abundant oceanic viruses of uncultured marine group II Euryarchaeota. Curr Biol 27:1362–1368
13. Paez-Espino D, Eloe-Fadrosh EA, Pavlopoulos GA et al (2016) Uncovering earth's virome. Nature 536:425–430
14. Vázquez-Castellanos JF, García-López R, Pérez-Brocal V et al (2014) Comparison of different assembly and annotation tools on analysis of simulated viral metagenomic communities in the gut. BMC Genomics 15:37
15. Aitchison J (1981) A new approach to null correlations of proportions. Math Geol 12:175–189
16. Li H (2015) Microbiome, metagenomics, and high-dimensional compositional data analysis. Annu Rev Stat Its Appl 2:73–94
17. Arumugam M, Raes J, Pelletier E et al (2011) Enterotypes of the human gut microbiome. Nature 473:174–180
18. Paulson JN, Stine OC, Bravo HC et al (2013) Differential abundance analysis for microbial marker-gene surveys. Nat Methods 10:1200–1202
19. Segata N, Izard J, Waldron L et al (2011) Metagenomic biomarker discovery and explanation. Genome Biol 12:R60
20. Pérez-Brocal V, García-López R, Nos P et al (2015) Metagenomic analysis of crohn's disease patients identifies changes in the virome and microbiome related to disease status and therapy, and detects potential interactions and biomarkers. Inflamm Bowel Dis 21(11):2515–2532
21. Weiss S, Xu ZZ, Peddada S et al (2017) Normalization and microbial differential abundance strategies depend upon data characteristics. Microbiome 5:27
22. Angly F, Rodriguez-Brito B, Bangor D et al (2005) PHACCS, an online tool for estimating the structure and diversity of uncultured viral communities using metagenomic information. BMC Bioinfo 6:41
23. Reyes A, Haynes M, Hanson N et al (2010) Viruses in the faecal microbiota of monozygotic twins and their mothers. Nature. Nat Publ Group 466:334–338
24. Yatsunenko T, Rey FE, Manary MJ et al (2012) Human gut microbiome viewed across age and geography. Nature 486:222–227
25. Oksanen J, Kindt R, Legendre P et al (2008) Vegan: community ecology package
26. Anderson MJ (2001) A new method for non-parametric multivariate analysis of variance. Austral Ecol 26:32–46
27. Kruskal JB (1964) Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika 29:1–27