IOSR Journal of Mathematics (IOSR-JM)
e-ISSN: 2278-5728, p-ISSN: 2319-765X. Volume 13, Issue 3 Ver. III (May - June 2017), PP 24-32
www.iosrjournals.org

A priori groups based on Bhattacharyya distance and partitioning around medoids algorithm (PAM) with applications to metagenomics

Rodríguez-Casado, Clara I.[1]; Monleón-Getino, Toni[1,2,+]; Cubedo, Marta[1]; Ríos-Alcolea, Martín[1,2]

[1](Department of Genetics, Microbiology and Statistics, University of Barcelona, Barcelona, Spain)
[2](Research Group in Biostatistics and Bioinformatics (GRBIO), Barcelona, Spain)
[+](Correspondance author)

Abstract:*Plants, animals and humans live in close association with microbial organisms. Increasingly, biologists have come to appreciate that microbes make up an important part of an organism's phenotype. This microbial community contains a unique complexity that makes it difficult to study their diversity. However, for many questions on the structure of the microbial community one only needs to know the relative order of diversity among samples rather than the total diversity. Unfortunately the culture of microorganisms can be complex but this has prompted the development of new scientific methodologies for their study. One of these methodologies is metagenomics. An important problem in metagenomics is measuring the dissimilarity between distributions of features, such as taxons or groups. The focus of this note is the proposal of a new method based on using Bhattacharyya distance and establishing a priori groups using the partitioning around medoids algorithm (PAM). The results reveal a good reduction in the size of the dataset and an interesting way of revealing possible subgroups "a priori" or communities among the microorganisms that make up the analyzed sample.*

Keywords:*Multivariate methods, applied statistical methods, data analysis, multidimensional scaling, metagenomics, cluster, biology, microbiology, metrical distances*

I. Introduction

The gut microbiota is home to more than 99% of the genetic information in humans and although there is an important connection between the gut microbiome and metabolism, immune health, disease, autism, allergies, and obesity, it remains a largely unexplored area of science [1]. Microbial communities contain a unique complexity that makes it difficult to study their diversity. However, for many questions on the structure of the microbial community one only needs to know the relative order of diversity among samples rather than total diversity. Unfortunately the culture of microorganisms can be complex, prompting the development of new scientific methodologies for their study. One of these methodologies is metagenomics.

Metagenomics (also referred to as environmental and community genomics) is the study of genetic (genomic analysis of microorganisms) material recovered directly from environmental samples by direct extraction and cloning of DNA from an assemblage of microorganisms [2]. In any biological system information is ultimately linked to the DNA sequences present, and microbial communities are no exception. In microbial communities we used 'word' frequency profiles of operational taxonomic units (OTUs) as a proxy for the composition of the bacterial community at the genomic level, thus avoiding the need to define bacterial species or taxonomic groups [3].

The broad field of metagenomics may also be referred to as environmental genomics, ecogenomics or community genomics. While traditional microbiology and microbial genome sequencing and genomics rely upon cultivated clonal cultures, early environmental gene sequencing cloned specific genes (often the 16S rRNA gene) to produce a profile of diversity in a natural sample [4].

The development of metagenomics stemmed from the ineluctable evidence that as-yet-uncultured microorganisms represent the vast majority of organisms in most environments on earth. This evidence was derived from analyses of 16S rRNA gene sequences amplified directly from the environment, an approach that avoided the bias imposed by culturing and led to the discovery of vast new lineages of microbial life [2].

In a very recent study [3], we addressed the question of how to explore diversity (species richness) and complexity (frequency distribution) in microbial communities directly from a limited amount of metagenomic data and how to characterize communities efficiently. For this purpose we built the library MetagenOutLDA.

Next generation sequencing and other recent techniques applied to microbial metagenomics have transformed the study of microbial diversity. Microbial metagenomics, or sequencing of DNA extracted from microbial communities, provides a means to determine what organisms are present without the need for

isolation and culturing, which can only be used for less than 1% of the species in a typical environment [5]. In many cases the number of groups or communities living together in the samples is unknown, although we would like to have an exploratory method that would allow us to establish a series of communities by types of taxa or samples for later study. The development of statistics to extract ecologically meaningful information from these datasets has not developed as quickly as the metagenomic techniques. In particular, tools that can account for the discrete nature, sparsity, and variable size of these datasets are lacking [5].

The main objective of this paper is to present a new exploratory multivariate method to establish a priori a classification of the microbial communities directly from the metagenomics matrices. For this purpose we will use an example to illustrate how the method works in practice.

II. Material and Methods

II.I. *R package*

All the statistical methods proposed in this work were developed in the computer GNU project: R. R is a widely used free software environment and programming language for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS, and is supported by the R Foundation for Statistical Computing [6]. The R language is widely used among statisticians and data scientists for developing statistical software and data analysis. The popularity of R has increased substantially in recent years.

The source code for the R software environment is written primarily in C, Fortran and R. R is freely available under the GNU General Public License on https://www.r-project.org/[6].

II.II. *Reported cases*

The data analyzed corresponds to a metagenomics matrix from matrix yy4 (see TABLE 1) of the R package matR (metagenomics analysis tools for R) which is an analysis client for the MG-RAST metagenome annotation engine, part of the US Department of Energy (DOE) Systems Biology Knowledge Base (KBase). See: https://github.com/MG-RAST/matR BIOM annotation data for certain metagenomes and projects. The matrix yy4 is an object of class biom for demonstration purposes, containing annotation data for certain sets of metagenomes.

Table 1: *Metagenomics matrix yy4 used as example from library matR*

num	Taxon (OTU)	sample mgm4447102.3	sample mgm4447103.3	sample mgm4447192.3	sample mgm4447943.3
1	Acetobacteraceae	63	140	27	11
2	Acholeplasmataceae	11	37	0	75
3	Acidaminococcaceae	160	1025	151	2172
4	Acidimicrobiaceae	0	0	4	0
5	Acidithiobacillaceae	0	67	6	0
6	Acidobacteriaceae	0	38	13	0
7	Acidothermaceae	158	26	55	113
8	Actinomycetaceae	15561	3511	4139	14627
9	Aerococcaceae	260	631	247	694
10	Aeromonadaceae	109	401	124	95
11	Alcaligenaceae	411	880	174	144
12	Alcanivoracaceae	37	255	28	5
…	…	…	…	…	…
266	Xanthomonadaceae	311	543	115	44

To use the matrix yy4, first load package matR on R package and use it by means of: >library(matR) >yy4

Table 2: Theoretical structure of M (metagenomics matrix input)

num	**Taxon (OTU)**	**Sample 1**	**Sample 2**	**Sample jth**	**Sample n**	
1	OTU.1	m_{11}	m_{12}	⋯	m_{1n}	$N_{1.}$
2	OTU.2	m_{21}	m_{22}	⋯	m_{2n}	$N_{2.}$
⋮	OTU ith	⋮	⋮	m_{ij}	⋮	⋮
k	OTU.k	m_{k1}	m_{k2}	⋯	m_{kn}	$N_{k.}$
		$N_{.1}$	$N_{.2}$	⋯	$N_{.n}$	N

TABLE 2 shows the theoretical composition of a metagenomic matrix (***M***) (*k* rows: taxon or OTU (operational taxonomic unit) and *n* columns: samples). This matrix format is used for different statistical analyses in this paper. ***M*** shows the samples in the columns (in our example, see TABLE 1, *n*=4 samples) and the taxa identified by the molecular method (16S metagenomics) or the organism identified (OTU: operational taxonomic unit, in our example, see TABLE 1, k=266) in the rows. The dimension of ***M*** is:

$$\text{Dim}(M) = k \cdot n \quad (1)$$

As a result of metagenomic analysis, ***M*** can be very large and usually has few samples and thousands of OTUs, most with small frequencies or 0 (sparse matrix).

II.III.. *Data codification*

Each sample is represented by one k-dimensional random vector$X_{.j}; X_{.j} = (m_{1j}, m_{2j}, \dots, m_{kj})$, where m_{kj} represents the number (frequency) of the jth sample at the kth OTU. In the same manner the n-dimensional random vector $X_{i.}$ can be defined.

An important issue is the probability distribution underlying the matrix ***M*** and this was studied in the article "Bacterial Metagenomics: Associated Probability Distributions and Profile Analysis" [7] in which different distribution probability models such as multinomial, Diritchlet-multinomial and their combinations were tested.

The distribution of each random vector $X_{i.}$ and $X_{.j}$ can be fitted to a multinomial distribution,

$$X_{.j} \sim MN\left(N_{.j}, \theta_{1j}, \dots, \theta_{kj}\right);\ \forall j = 1,..,n \qquad \text{with } \sum_{i=1}^{k} \theta_{ij} = 1 \ \ \forall j \quad (2)$$

$$X_{i.} \sim MN\left(N_{i.}, \tilde{\theta}_{i1}, \dots, \tilde{\theta}_{in}\right);\ \forall i = 1,..,k \quad \text{with } \sum_{j=1}^{n} \tilde{\theta}_{ij} = 1 \ \ \forall i \quad (3)$$

The multinomial distribution is a multivariate generalization of the binomial distribution, where $X_{ij} \sim Bin\left(m_{ij}, \theta_{ij}\right); 1 \le \theta_{ij} \le 1;\ \forall j = 1,..,n;\ \forall i = 1,..,k$

e.g. if we consider the partition of all sample space Ω^j the j-sample space in *k* parts:

$$A_{1j} \quad A_{2j} \quad \dots \quad A_{kj}$$

One individual selected randomly has the probability θ_{kj} of belonging to the taxon A_{kj} in the above partition:

$$\left.\begin{array}{l} P\left(A_{1j}\right) = \theta_{1j} \\ P\left(A_{2j}\right) = \theta_{2j} \\ \vdots \\ P\left(A_{kj}\right) = \theta_{kj} \end{array}\right\} \sum_{i=1}^{k} \theta_{ij} = 1\,;\ \forall j = 1,..,n$$

If we wish to calculate for a sample j the probability of having $N_{j.}$ individuals, m_{1j} belongs to class A_{1j}, m_{2j} belongs to class A_{2j},...,m_{kj} belongs to class A_{kj}, with the restriction

$$\sum_{i=1}^{k} m_{ij} = N_{j.} \quad , \forall j = 1,..,n \quad (4)$$

Using the multinomial function of density (mass function) we can calculate this probability, $MN(N_{j.}; \theta_j = (\theta_{1j}, \theta_{2j}, \dots, \theta_{kj}))$:

$$P\left[\left(A_{1j} = m_{1j}\right) \cap \left(A_{2j} = m_{2j}\right) \cap \dots \cap \left(A_{kj} = m_{kj}\right)\right] = \frac{N_{.j}!}{m_{1j}!m_{2j}!\dots m_{kj}!} \theta_{1j}^{m_{1j}} \cdot \theta_{2j}^{m_{2j}} \cdot \dots \cdot \theta_{kj}^{m_{kj}} \quad (5)$$

where $0 \le \theta_{ij} \le 1$ for all i in 1 to k, and $\theta_{1j} + \dots + \theta_{kj} = 1$, and if k = 1 the mass function reduces to the binomial, $\forall j = 1,..,n$.

II.IV.*Statistical analysis*

A powerful multivariate methodology was used in this study based on the work of Rios et al. [8] and this is the first time that it has been combined with metagenomics data. Fig. 1 shows, schematically, how the different statistical procedures are used to obtain a classification according to different OTUs.These processes are discussed further below.

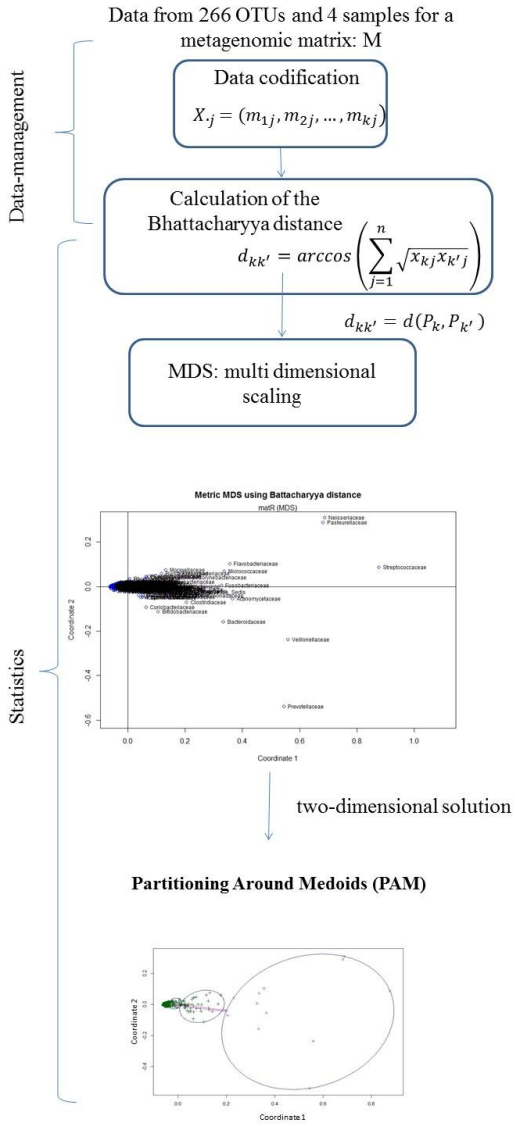

Figure 1: Algorithm of the data management and statistical processes used during this study to revealing possible subgroups "a priori" or communities among the microorganisms.

II.IV.I. Calculation of the Bhattacharyya distance

Let *p(i)* and *p'(i)* represent two multinomial populations, each consisting of *N* classes with respective probabilities $p(i = 1), \ldots, p(i = N)$ and $p'(i = 1), \ldots, p'(i = N)$. Since $p(i)$ and $p'(i)$ represent probability distributions, $\sum_{i=1}^{N} p(i) = \sum_{i=1}^{N} p'(i) = 1$. The Bhattacharyya distance [9] (or measure or coefficient of Bhattacharyya, BC) is a divergence-type measure between distributions, defined as:

$$BC(p, p') = \sum_{i=1}^{N} \sqrt{p(i)p'(i)} \quad (6)$$

The Bhattacharyya distance has a simple geometric interpretation [10] as the cosine of the angle between the N-dimensional vectors $\left(\sqrt{p(1)}, \ldots, \sqrt{p(N)}\right)^T$ and $\left(\sqrt{p'(1)}, \ldots, \sqrt{p'(N)}\right)^T$. Thus, if the two distributions are identical, we have:

$$\cos(\theta) = \sum_{i=1}^{N} \sqrt{p(i)p'(i)} = \sum_{i=1}^{N} \sqrt{p(i)p(i)} = \sum_{i=1}^{N} p(i) = 1 \quad (7)$$

Consequently, $\theta = 0$. Furthermore, based on Jensen's inequality [11] we have:

$$0 \leq BC(\mathrm{p},\mathrm{p}') = \sum_{i=1}^{N} \sqrt{p(i)p'(i)} = \sum_{i=1}^{N} p(i) \sqrt{\frac{p'(i)}{p(i)}} \leq \sqrt{\sum_{i=1}^{N} p'(i)} = 1 \qquad (8)$$

A potentially undesirable property of the distance is that it does not impose a metric structure since it violates at least one of the distance metric axioms [12].

In the problem that concerns us, each sample of the matrix ***M*** (see TABLE 1 and 2) was assigned to the vector $x_i = (x_{i1}, \ldots, x_{in})$ a realization of the random vector x_i, defined as:

$$x_i = \left(x_{i1} = \frac{m_{i1}}{N_{1.}}, x_{i2} = \frac{m_{i2}}{N_{2.}}, \ldots, x_{in} = \frac{m_{in}}{N_{n.}}\right);\ \forall i = 1, .., k \qquad (9)$$

where x_{ij} is the fraction per unit of the jth sample for the ith OTU, meaning that in our example the following restriction is imposed: $x_{1j} + x_{2j} + \cdots + x_{266j} = 1\ ;\ \forall j = 1, .., n.$
If we consider $x_i$as a random variable distributed as a multinomial with parameters $(x_{i1}, x_{i2}, \ldots, x_{i4})$, the distance between the OTU k represented by $P_k = (x_{k1}, x_{k2}, \ldots, x_{k4})$and the OTU k' represented by$P_{k'} = (x_{k'1}, x_{k'2}, \ldots, x_{k'4})$is given by$d$, where *d* is the Bhattacharyya distance [9]:

$$d_{kk'} = arccos\left(\sum_{j=1}^{n=4} \sqrt{x_{kj}\, x_{k'j}}\right) \qquad (10)$$

To calculate the Bhattacharyya distance we used the following R script:

```
> Q <- sqrt(M) #M is the normalized metagenomics matrix yy4 from library matR.
> BC <- Q%*%t(Q)
> BC <- ifelse(BC>1., 1., BC)
> D2B <- acos(BC)
> D2B # Bhattacharyya matrix of distances
```

II.IV.II. Multidimensional scaling

Multidimensional scaling (MDS) is a means of visualizing the level of similarity of individual cases in a dataset. It refers to a set of related ordination techniques used in information visualization, in particular to display the information contained in a distance matrix. An MDS algorithm aims to place each object in N-dimensional space such that the between-object distances are preserved as well as possible. Each object is then assigned coordinates in each of the N dimensions [13].
Based on the Bhattacharyya distances obtained, a MDS was performed. The distance matrix between OTUs, $D = \left(d_{kk'}\right)_{266\times266}$, $d_{kk'} = d\left(P_k, P_{k'}\right)$is a distance between $P_{k'}$, $P_{k'}$ OTUs *k,k'* respectively. $A = \left(a_{kk'}\right)_{266\times266}$, $a_{kk'} = -\frac{1}{2}d_{kk'}^2$is a measure of the similarity $H = \left(h_{kk'}\right)_{266\times266}$,$H = I - \frac{1}{266}EE'$where I is an identity matrix of size *266x266. E=(1,…,1)'* is a column vector of size *266.B=HAH* is the sample matrix of covariance of the values of *A.* Their eigenvalues are: $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r \geq 0 \geq \lambda_{r+1} \geq \cdots \geq \lambda_{266}$ (real numbers since B is symmetrical) and eigenvectors of the matrix *B* were calculated.

$$B = TD_\lambda T = \left(TD_\lambda^{1/2}\right)\left(D_\lambda^{1/2}T\right) = \left(TD_\lambda^{1/2}\right)\left(TD_\lambda^{1/2}\right) \qquad (10)$$

$$D_\lambda = diag(\lambda_1, \ldots, \lambda_{266}) \qquad (11)$$

where the *ith* column of *T* is the *ith* eigenvector associated with λ_i . If

$$Y = TD_\lambda^{1/2} \qquad (12)$$

the rows of *Y* are the coordinates of *266* points *($Q_1, \ldots, Q_{266}$)* on R^{266}. If *B* is a non-negative definite matrix:

$$d\left(Q_k, Q_{k'}\right) = d\left(P_k, P_{k'}\right) \qquad (13)$$

If *B* is not a non-negative definite matrix, we obtain pure imaginary values when calculating $D_\lambda^{1/2}$ and the rows of *Y* are in the form:

$$\left(y_{k1}, \ldots, y_{kr}, \sqrt{-1}y_{kr+1}, \ldots, \sqrt{-1}y_{k266}\right) \qquad (14)$$

If $\psi = (\lambda_1^2 + \lambda_2^2 + ... + \lambda_r^2)/(\lambda_1^2 + ... + \lambda_{266}^2)$ is a sufficiently large value (greater than 0.9) we can consider the rows of *Y* in the form:

$$(y_{l1}, ..., y_{lr}, 0, ..., 0) \quad (15)$$

Then we performed a MDS analysis with the 266 points $(Q_1, ..., Q_{266})$. This method allowed us to construct a diagram showing the relationships between a number of objects. The diagram is a small *p*-dimensional space, generally *p=2* or *p=3*. The main purpose is to reduce data from a large number of variables to fewer components, so making it possible to view the similarities and differences between the OTUs studied ([13],[14],[15]).

All analyses were performed using the function cmdscale() from the R package HSAUR as reported by Venables & Ripley [15] for metric MDS.

II.IV.III.Partitioning Around Medoids (PAM) algorithm

Cluster analysis aims to group a set of objects (e.g. OTUs (see TABLE 1 and 2) from a matrix ***M***) in such a way that objects in the same group (called a cluster) have a high degree of similarity in the same cluster. There are different methods of clustering and one of the most popular is the partitioning method. This requires the analyst to specify the number of clusters to extract. Nowadays many disciplines are using these kinds of algorithms to separate datasets into groups in an automated way, whilst still achieving good quality results.

The clustering process is not a universal process because there are many groups of datasets, for some of which the kind of metric used is relevant, whereas for others the entities that represent each cluster are more interesting. Like dataset groups there are many clustering algorithms and each one tries to take advantage of the data type, with each one being more suited for a specific kind of data.

This section will explain a little more about the Partitioning Around Medoids (PAM) algorithm, showing how the algorithm works, its parameters and what they mean, an example of a dataset, how to execute the algorithm, and the result of that execution with the dataset as input.

The PAM algorithm was developed by Leonard Kaufman and Peter J. Rousseeuw [16],[17] in 1987, and this algorithm is very similar to K-means, mostly because both are partitional algorithms. This algorithm is described in the entry on Partitioning Around Medoids (PAM) Algorithm [18] and the mathematical description of the algorithm adapted to the example at hand is reproduced in part.

PAM breaks the dataset into groups (clusters), by trying to minimize the error. However, PAM works with medoids, which are an entity of the dataset that represents the group in which it is inserted, and K-means works with centroids, which are an artificially created entity that represent its cluster. A nice property is that PAM allows clustering with respect to any specified distance metric like the Bhattacharyya distance, $d_{kk'}$(see (10). In addition, the medoids are robust representations of the cluster centers in the reduced space, which is particularly important in the common context that many elements do not belong well to any cluster, so we consider that it is a good method to use in the genetic field. PAM computes medoids for each cluster. PAM is computationally more costly than K-means since it requires pairwise distance calculations in each cluster.

The PAM algorithm partitions the dataset of n objects into p clusters, where both the dataset and the number p is an input of the algorithm. This algorithm works with a matrix of dissimilarity, whose goal is to minimize the overall dissimilarity between the representatives of each cluster and its members. The algorithm uses the following model to solve the problem:

$$F(x) = \min \sum_{k=1}^{n} \sum_{k'=1}^{n} d(k, k') z_{kk'} \quad (16)$$

Subject to:

1. $\sum_{i=1}^{n} z_{kk'} = 1, j = 1,2, \dots, n$
2. $z_{ij} \leq y_i, i, j = 1,2, \dots, n$
3. $\sum_{i=1}^{n} y_i = p, p = num\ clusters$
4. $y_i, z_{ij} \in \{0,1\}, i, j = 1,2, \dots, n$

where *F(x)* is the main function to minimize, *d(k,k')* is the dissimilarity measurement between the entities *k* and *k' (see (10)*, and z_{ij} is a variable that ensures that only the dissimilarity between entities from the same cluster will be computed in the main function. The other expressions are constraints that have the following functions: (1.) ensures that every single entity is assigned to one cluster and only one cluster, (2.) ensures that the entity is assigned to its medoid that represents the cluster, (3.) ensures that there are exactly *p* clusters and (4.) lets the decision variables assume just the values of 0 or 1.

The PAM algorithm can work over two kinds of input, the first is the matrix representing every entity and the values of its variables, and the second is the dissimilarity matrix; in the latter the user can provide the

dissimilarity directly as an input to the algorithm, instead of the data matrix representing the entities. Either way the algorithm reaches a solution to the problem, and in a general analysis the algorithm proceeds this way:

Build phase:
1. Choose *p* entities to become the medoids, or if these entities were provided use them as the medoids;
2. Calculate the dissimilarity matrix if it was not supplied; here it is the Bhattacharyya distance, $d_{kk'}$(see (10)
3. Assign every entity to its closest medoid.

Swap phase:
4. For each cluster determine whether any of the entities of the cluster lower the average dissimilarity coefficient; if that is the case select the entity that lowers this coefficient the most as the medoid for this cluster;
5. If at least one medoid has changed go to (3), otherwise end the algorithm.

III. Results

The Battacharyya distance was calculated over matrix *M* (see Table 1; Table 2 describes its structure) using the mathematical process developed in section 2.3.1 (see (9), (10) and (11)). MDS was performed using the steps described in section II.IV.II.

The MDS produced a two-dimensional solution and the main end points found are summarized in Table 3. The Mardia percentages accounting for the first two axes were 77.7% and 13.8% and these components explained more than 91.5% of the total variation. The two-dimensional coordinates and the display obtained are shown in Table 4 and Fig. 2. The first component reflects the relative abundance between samples, indicating the general level of abundance of each OTU and sample. On the left-hand side of Fig. 2 are the OTUs with a low level of relative abundance in all samples. The first component was clearly positive (indicating highest relative abundance and importance of OTUs) in 15 OTUs (*Streptococcaceae, Prevotellaceae, Veillonellaceae,* …). The second component reflects only 13.8% of the variance, and we think this reflects changes in the composition of the samples (sample variability): on the positive axes we find the less variables samples and on the negative axes the more variable samples.

Table 3: Eigenvalues, criterion of Mardia and inertia percentages

Eigenvalues (λ_i) *for all axes*	$\lambda_1 = 3.752463e+00$ $\lambda_2 = 6.658428e-01$ $\lambda_3 = 2.712456e-01$ … $\lambda_{266} = -5.270244e-02$
Criterion of Mardia for the first two axes	$\frac{\lambda_1^2 + \lambda_2^2}{\sum_{i=1}^{266} \lambda_i^2} = 0.9944883$
Inertia percentages for the first two axes	$\frac{\lvert\lambda_{1\rvert} + \lvert\lambda_2\rvert}{\sum_{i=1}^{266} \lvert\lambda_i\rvert} = 0.9145867$
Inertia percentages for the first axe	$\frac{\lvert\lambda_1\rvert}{\sum_{i=1}^{266} \lvert\lambda_i\rvert} = 0.7767577$
Inertia percentages for by the second axe	$\frac{\lvert\lambda_2\rvert}{\sum_{i=1}^{266} \lvert\lambda_i\rvert} = 0.1378291$

Table 4: *Coordinates of the OTUs used in the MDS analysis*

	Coordinate 1	**Coordinate 2**
OTU_1	-0.029165151	0.010939870
OTU_2	-0.045212593	-0.009269291
OTU_3	0.044075543	-0.048979870
OTU_4	-0.060191831	0.002857603
OTU_5	-0.05180509	0.007561300
…	…	…
OTU_{266}	0.007561399	0.02163315

All of these results can be complemented by the heat-map in Fig. 3, which is a graphical representation of data (OTUs and samples) where the variation of the abundance (frequency) contained in the matrix *M* is represented as the intensity of color. The heat-map shows the most frequent taxons in the upper part and the less frequent taxons in the bottom part, but only those that have an abundance higher than 0.5%.

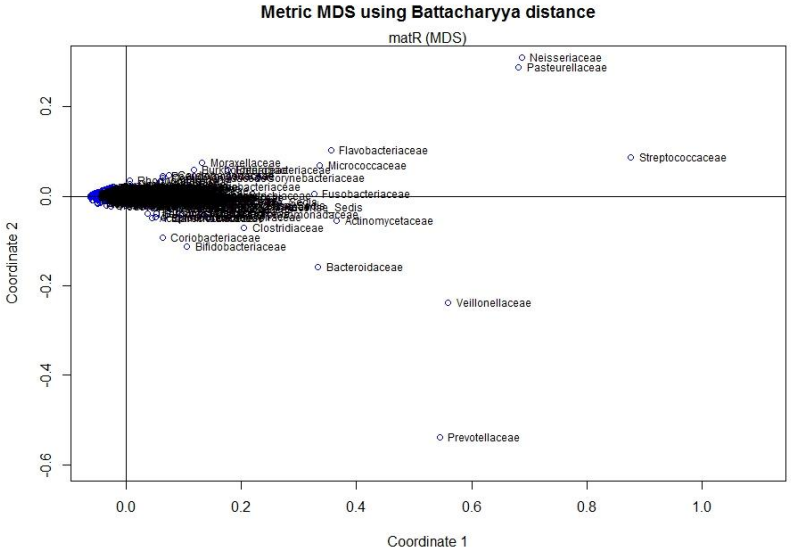

Figure 2. Two-dimensional graphic display of matrix matr based on metric Multidimensional Scaling and the use of Battacharyya distance.

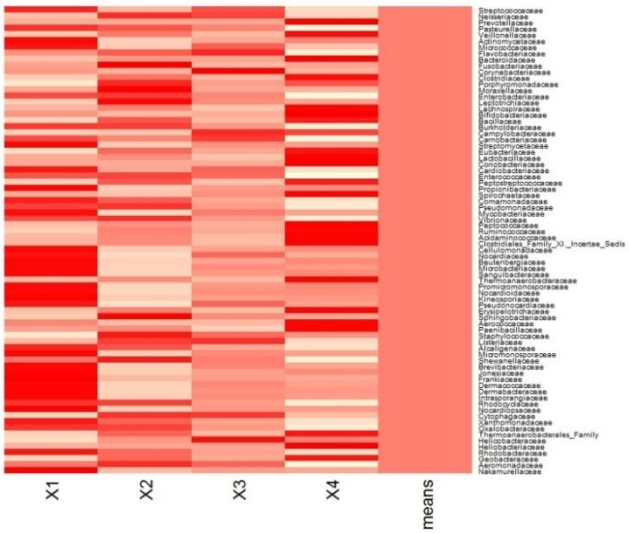

Figure 3: Heat map of the percentage of OTUs and samples (X1, X2, X3, X4 and their mean). The intensity of the color indicates a greater abundance. Only those OTUs> 0.5% (right list)

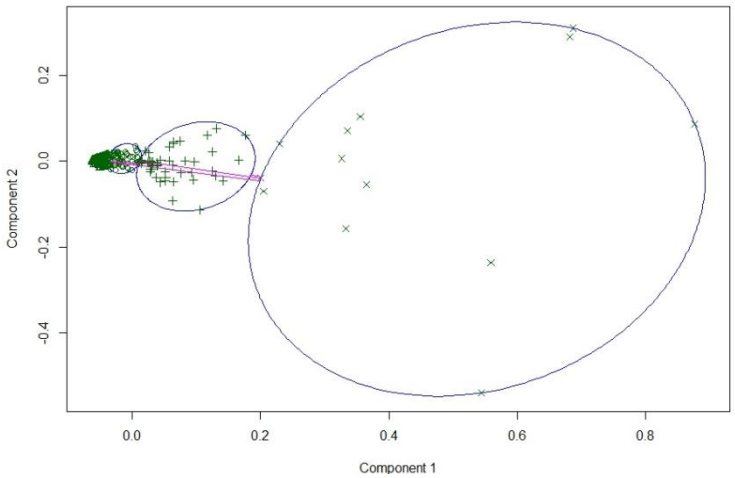

Figure 4: On the representation made in Figure 4, 3 a priori groups were obtained using the PAM algorithm

Fig. 4 is the PAM analysis performed using the cluster algorithm described in section 2.3.3. PAM automatically divided the 206 OTUs into three groups which overall account for 61.4% good classification: 78.9% for the first group, 26.2% for the second group and 2.3% for the third group. Three is the number of "a priori" groups that we can find in this sample, because this number gives the best classification in accordance with the criteria. This number of groups was established using the silhouette method, a method of interpretation and validation of consistency within clusters of data; the silhouette provides a succinct graphical representation of how well each object lies within its cluster (Rousseeuw, 1986). In the other cases the percentage of good classification was 57.5% for two groups, and 47.6% for four groups.

IV. Conclusion

Measurement of diversity is important for understanding community structure and dynamics, but has been particularly challenging for microbes. Microbiologists have recently discovered that ecologists and evolutionary biologists studying the diversity of macroorganisms have developed a range of approaches to analyze the environmental diversity patterns, many of which can be applied to microorganisms. Basically the analysis of biodiversity that is carried out in metagenomics is fundamentally based on the analysis used in classical ecology, involving richness, abundance, alpha diversity and beta diversity and often incorporating concepts such as phylogenetic relationships and taking into account how the samples were obtained and the technical noise.

The focus of this note was the proposal of a new method (used with this type of data for the first time) based on using Bhattacharyya distance, MDS and establishing a priori groups using the partitioning around medoids algorithm (PAM). The results revealed a good reduction in the size of the dataset and an interesting way of revealing possible subgroups "a priori" or communities among the microorganisms that make up the analyzed sample.

These groups are found "automatically" once studied and validation of the method by experts in metagenomics would help advance the development of new statistical multivariate methods to extract ecologically meaningful information from these datasets, including characterizing the biodiversity of the microbiome, finding the number of communities (subpopulations) that interact in the sample and observing the behavior and complexity of the microorganisms in the sample analyzed.

Acknowledgements

This work is partially supported by grant 2014 SGR 464 (GRBIO) from the Departament d'Economiai Coneixement de la Generalitat de Catalunya (Spain).

References

[1] Pollan, M. (2013). Some of My Best Friends Are Germs. New York Time magazine. (Retrieved May 03, 2016, from http://www.nytimes.com/2013/05/19/magazine/say-hello-to-the-100-trillion-bacteria-that-make-up-your-microbiome.html?_r=1)
[2] Handelsman J. 2004. Metagenomics: Application of Genomics to Uncultured Microorganisms Microbiology and Molecular Biology Review 68(4): 669–685.
[3] Rodríguez CI, Monleón-Getino T. 2016. A new R library for discriminating groups based on abundance profile and biodiversity in microbiome metagenomic matrices. Article in International Journal of Scientific and Engineering Research 7(10):243-253
[4] Wikipedia (a), 2016. Metagenomics. (From https://en.wikipedia.org/wiki/Metagenomics)
[5] Holmes I, Harris K, Quince C (2012) Dirichlet Multinomial Mixtures: Generative Models for Microbial Metagenomics. PLoS ONE 7(2):
[6] R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.
[7] Pizarro P. 2016. Bacterial Metagenomics: Associated Probability Distributions and Profile Analysis. Master thesis of the master in Biostatistics and Bioinformatics (UOC-OPC). Adviced by Toni Monleón Getino.
[8] Ríos D, Monleón-Getino T, Cubedo M, Ríos M. 2017. A Graphical Classification of European Countries According to Physical Activity Level of Its Citizens. Open Access Library Journal 03(12):1-11
[9] Bhattacharyya A. (1946). On a measure of divergence between two multinomial populations. Sankhyä. 7:401-6.
[10] Derpanis KG. 2008. The Bhattacharyya Measure. Semantic Scoolar. (From https://pdfs.semanticscholar.org/768f/743d6697332e909587234b8643839fdd326e.pdf)
[11] Cover, T., & Thomas, J. (1991). Elements of information theory. Wiley
[12] Fukunaga, K. (1990). Introduction to Statistical Pattern Recognition (Second Edition). Academic
[13] Borg I. & Groenen P. J. F. (2005). Modern Multidimensional Scaling: Theory and Applications. New York: Springer. 2nd. Ed.
[14] Mardia K. V., Kent J. T. & Bibby J. M. (1979). Multivariate Analysis. Academic Press. London
[15] Venables W. N. & Ripley B. D. (2002). Modern Applied Statistics with S. 4th. Ed. Springer. N. Y.
[16] Rousseeuw PJ. 1987. Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. Computational and Applied Mathematics. 20: 53–65.
[17] Kaufman, L. and Rousseeuw, P.J. (1987), Clustering by means of Medoids, in Statistical Data Analysis Based on the –Norm and Related Methods, edited by Y. Dodge, North-Holland, 405–416
[18] Wikipedia(b). 2016. The Partitioning Around Medoids (PAM) Algorithm (https://en.wikibooks.org/wiki/Data_Mining_Algorithms_In_R/Clustering/Partitioning_Around_Medoids_(PAM))