DOI: http://dx.doi.org/10.22201/fmvz.24486760e.2023.1030.900
Vol. 10 | 2023

Veterinaria México OA
Publicación Digital de la Facultad de Medicina Veterinaria y Zootecnia https://veterinariamexico.fmvz.unam.mx/

Claudia C Márquez-Mota[1]
0000-0002-4311-381X
Laura Piña-González[1]
0000-0002-0505-4126
Mónica Sánchez Tapia[2]
0000-0001-9748-5734
Nimbe Torres[2]
0000-0002-1741-0984
Armando R. Tovar[2]
0000-0002-1292-1312
Juan J. Loor[3]
0000-0003-1586-4365
Abdulrahman Alharthi[3]
0000-0002-9719-9444
Luis Corona[1]*
0000-0002-6640-7626

[1] Universidad Nacional Autónoma de México,
Facultad de Medicina Veterinaria y Zootecnia
Departamento de Nutrición Animal y Bioquímica,
Ciudad de México, México

[2] Instituto Nacional de Ciencias Médicas
y Nutrición Salvador Zubirán
Departamento de Fisiología de la Nutrición,
Ciudad de México, México.

[3] University of Illinois
Division of Nutritional Sciences
Department of Animal Sciences,
Urbana, Illinois.

***Corresponding author:**
Email address:
gochi@unam.mx

Supplementary material

Changes in the bacterial diversity of the ruminal liquid fraction of dairy cows on a corn stover based diet

Cite this as:

Márquez Mota CC, Piña-González L, Sánchez Tapia M, Torres N, Tovar AR, Loor JJ, et al. Changes in the bacterial diversity of the ruminal liquid fraction of dairy cows on a corn stover based diet. Veterinaria México OA. 2023;10. Supp.mat. doi: 10.22201/fmvz.24486760e.2023.1030.900.

Submitted: 2022-01-04
Accepted: 2023-02-08
Published: 2023-04-24

Additional information and declarations
can be found on page 13

UPSTREAM ANALYSIS

Decompress files from Illumina sequencing

```
tar –xvf Name_File_PrimerSortedDemultiplexed.tgz
```

Join two reds (R1 and R2)

```
join_paired_ends.py –f $PWD/16S_V3_Sample1_GTATCGTCGT_R1.fastq –r
$PWD/16S_V3_Sample1_GTATCGTCGTA_R2.fastq –o $PWD/fastq-join_joinedSample1
```

- Output: you get a folder for each sample, containing 3 files: join reads, unjoin 1 and unjoin 2 (all in .fastq format)

It can be done for each samples or use the multiple join script

```
multiple_join_paired_ends.py –i folder_samples –o folder_joinreads
--read1_indicator '_R1' --read2_indicator '_R2'
```

- Output: folder with a log.txt file and a folder for each sample. Inside the folder of each sample there are 3 files (same as above)

For further steps you will need the join reads file (.fastq format). For convenience I took each file and changed the name of it according to the sample (i.e. S01_join.fastq, S02_join.fastq, S03_join.fastq…) and store them on a new folder.

Split libraries

After you have a folder with join reads you continue to Split libraries (it's a step for quality filtering)

```
multiple_split_libraries_fastq.py –i join_reads –p parameter_file.txt –o
split_librariesSamples --demultiplexing_method sampleid_by_file
```

- Input: folder with the join reads of each sample as for –p you have to make a parameter file in .txt format where you specify the setting for the original script (ie. If you want to increase de quality score you will have to write on the parameter file: split_libraries_fastq.py:phred_quality_threshold 19)
- Output: you get a folder with 4 files (histograms.txt, log_.txt, seqs.fna, split_library_log.txt). The seqs.fna is the file you will need for further steps (in this file now you only have the sequences and not the quality score anymore)

Pick OTUs and chimera checking

On the same script you can pick the OTUs and do the chimera checking if you do it with the method of “usearch” which uses UCHIME (it has been shown that is faster and better than other methods for chimera checking)

```
pick_otus.py –i split_librariesSamples/seqs.fna –m usearch --db gold.fa –o
usearch_qf_resultsSamples/ --word_length 64
```

- Input: the seqs.fna file you got from the split libraries script
- Output: you get a folder with 27 files. The file you will need for further steps is seqs_otus.txt. On the website: http://qiime.org/tutorials/usearch_quality_filter.html you can see the 11 steps this scripts performs.

Pick representative set of sequences

After you have the OTUs you need to pick the representative sequence of each OTU

```
pick_rep_set.py –i usearch_qf_results3samples/seqs_otus.txt –f
split_librariesSamples/seqs.fna –m most abundant –o rep_setSamples.fna
```

- Input: you will need the file from the last step (pick_otus.py) and also as –f you need the seqs.fna file you obtained from the split libraries script
- Output: a file that contains the representative sequence of each OUT in .fna format

You can change the method for picking the representative sequences: random, longest or first

Assign taxonomy to OTUs

With this script you’re able to assign taxonomy to each OUT

```
assign_taxonomy.py –I rep_setSamples.fna –m rdp –c 0.80
```

- Input: the file you obtained from the last step of choosing the representative sequence of each OTU (rep_setSamples.fna)
- Output: a folder with 2 files (rep_setSamples_tax_assignments.log and rep_setSamples_tax_assignments.txt) you will need the file .txt for further steps.

There are different methods to assign taxonomy: BLAST, RDP, RTAX, mothur or uclust. I used RDP (Wang et al, 2007) since is the recommended, only changing the confidence value from 0.5 (default) to 0.8.

Make an OTU table

Now you tabulate the number of times an OTU is found in each sample

```
make_otu_table.py –i usearch_qf_resultsSamples/seqs_otus.txt –t
rdp_assigned_taxonomy/rep_setSamples_tax_assignments.txt –o
otu_tableSamples.biom
```

- Input: the file you obtained from the pick otus script (seq_otus.txt) and as –t you need the file from the assign taxonomy script (rep_setSamples_tax_assignments.txt)
- Output: a file in .biom format (Biological Observation Matrix).

Align sequences

The sequences must be aligned to infer a phylogenetic tree

```
align_seqs.py –i rep_setSamples.fna –o alignmentSamples/
```

- Input: the representative set of sequences file (.fna) from the pick_rep_set script
- Output: a folder with 3 files (sequences aligned.fasta, failures.fasta and log.txt) you will need the aligned.fasta por further steps.

The default and recommended method in PyNAST, but you can choose from MUSCLE, INFERNAL and PyNAST. The –t file (file of pre-aligned sequences) is a default file (from GreenGenes) and QIIME has it already installed

Filter alignment

This script will remove positions, which are gaps in every sequence (common for PyNAST)

```
filter_alignment.py –i alignmentSamples/rep_setSamples_aligned.fasta –o
filtered_alignmentSamples/ --remove_outliers
```

- Input: the aligned.fasta file from the last step
- Output: a folder with one file (rep_setSamples_aligned_pfiltered.fasta) you will need this file to create the phylogenetic tree

I chose the option to remove sequences whose distance from the majority consensus sequence is more that 3 SD above the mean

Make a phylogenetic tree

The phylogenetic tree represents the relationships among sequences in terms of the amount of sequence evolution from common ancestor

```
make_phylogeny.py –i
filtered_alignmentSamples/rep_setSamples_aligned_pfiltered.fasta –o
rep_set_treeSample.tre
```

- Input: the sequences aligned and filtered from the last step (rep_set_Samples_aligned_pfiltered.fasta)
- Output: a file in .tre format that you can view with FastTree

DOWNSTREAM ANALYSIS

Filter OTU Table

To reduce the number of OTUs (the ones with little number of sequences)

```
filter_otus_from_otu_table.py –i otu_table.biom –o filter_otu_table.biom
--min_count_fraction 0.00005
```

- Input: OTU table; with a min count fraction of 0.00005 you are removing the OTUs that have <0.005 of sequences from the total number of sequences
- Output: new OTU table filtered

OTU Table Summary

With this script you can review your OTU table: number of OTUs, number of samples, total counts, min, max, mean, SD, sequences of each sample

```
print_biom_table_summary.py  -i filter_otu_table.biom
```

- Input: any OTU table
- Output: you don’t get a file, on the terminal window you see the summary

Alpha Rarefaction

This script runs a workflow of other scripts:

1. **multiple_rarefaction.py**
2. **alpha_diversity.py**
3. **collate_alpha.py**
4. **make_rarefaction_plots.py**

```
alpha_rarefaction.py –i filter_otu_table.biom –o alpharare/ -t tree_repset.tre
–m map_file.txt –e 3778 –p parameter_file.txt
```

- Input: filtered OTU table, phylogenic tree, map file and parameter file (with the parameter file you get to choose the index you want to use it needs to be a .txt file alpha_diversity:metrics shannon,PD:whole_tree,chao1,observed_tree), -e is the depth of the sequences you want to run the script,

- Output: a folder with two folders and a .txt file. You get the rarefaction plots.

Beta Diversity Through Plots

Calculates beta diversity, PCoA and 3D plots

```
beta_diversity_through_plots.py –i filter_otu_table.biom –o betadiv_plots –t
tree_repset.tre –m map_file.txt –e 3778
```

- Input: filtered OTU table, phylogenic tree, map file, and the depth of the sequences.
- Output: a folder with two folders and six files. With this you get the distance matrix, principal component and unweighted/weighted 3D plots

Summarize Taxa

It creates plots and .txt files for each taxonomy level (2,3,4,5,6)

```
summarize_taxa_through_plots.py –o taxa_summary –i filter_otu_table.biom –m
map_file.txt
```

- Input: filtered OTU table and map file
- Output: plots and otu tables of each level

Make Distance Boxplots

It compares distances between and within groups

```
make_distance_boxplots.py  -d betadiv_plots/unweighted_unifrac_dm.txt –m
map_file.txt –f Treatment –n 200 –o distance_boxplots
```

- Input: you need the distance matrix file from the beta_diversity_through_plots.py script (unweighted_unifrac_dm.txt), the map file. The script needs the fields to compare –f (ie. By Treatment) and the number of permutations –n
- Output: two files, a .pdf file with the boxplots and a .txt file with the statistics

Compare Alpha Diversity

Compares alpha diversity using t test (parametric or non-parametric)

```
compare_alpha_diversity.py –i alpharare/alpha_div_collated/PD_whole_tree.txt
–m map_file.txt –c Treatment –n 200 –o alphadiversity_compare
```

- Input: you need the .txt file from the alpha_rarefaction.py script, the category to compare –c (ie. Treatment) and the number of permutations –n
- Output: a folder with two files, a .pdf file with the boxplots and a .txt file with the statistics

Jackknifed Beta Diversity

Uses jackknife replicates to estimate the uncertainty in PCoA and fierachical clustering of microbial communitites. Performs several steps:

1. **Computes beta diversity distance matrix from full OTU table and tree**
2. **Builds UPGMA tree from full distance matrix**
3. **Builds rarefies OTU tables**
4. **Computes distances matrices from rarefied OUT tables**
5. **Builds UPGMA trees from rarefied distance matrices**
6. **Compares rarefied UPGMA trees and determinates jackknife support for tree nodes**
7. **Computes PCoA on each rarefied distance matrix**
8. **Compares rarefied PCoA plots from each rarefied distance matrix**

```
jackknifed_beta_diversity.py –i filter_otu_table.biom –o
jackknifedbetadiversity –e 3778 –m map_file.txt –t tree_repset.tre
```

- Input: filtered OTU table, the map file and depth of sequences –e
- Output: a folder with one file and four other folders

Compare Categories

To analyze statistical significance of sample groupings using distance matrices; several statistics can be performed:

- **adonis**
- **ANOSIM**
- **BIO-ENV**
- **Moran's I**
- **MRPP**
- **PERMANOVA**
- **db-RDA**

```
compare_categories.py --method Adonis –i
betadiv_unweightedunifrac/unweighted_unifrac_filter_otu_table.txt –m
map_file.txt –c Treatmet –o Adonis/
```

- Input: this scripts needs the .txt file from the beta diversity script (ie. Unweighted_unifrac_otu_table.txt), the method for the statistical analysis –method, mapping file and the category to compare –c
- Output: a folder, which contains a .txt file with the statistical results (p value, R, r^2, F, etc.)

Make OTU Network

This script generates the OTU network files to be used in Cytoscape and statistics for those networks.

```
make_otu_network.py –i filter_otu_table.biom –m map_file.txt –b Treatment –o
otu_network
```

- Input: the filtered OTU table, and the category between samples –b
- Output: a folder containing the files necessary for creating the OTU network on Cytoscape

Make OTU Heatmap

This script makes a heatmap where each row is an OTU and each column is a sample

```
make_otu_heatmap.py –i filter_otu_table.biom –m map_file.txt –c Treatment –t
tree_repset.tre –o heatmap_by_treatment
```

- Input: filtered OTU table, tree, mapping file and category to order the samples –c
- Output: a .pdf file with an image of the heatmap

** Additional Scripts**

Bootstrapped Tree

Creates a tree colored by bootstrap support:

- **Red: 75-100%**
- **Yellow: 50-75%**
- **Green: 25-50%**
- **Blue: <25%**

```
make_bootstrapped_tree.py –m
jackknifedbetadiversity/unweighted_unifrac/upgma_cmp/master_tree.tre –o
bootstrapped_tree.pdf
```

- Input: In this case I used the master tree you get from the jackknifed script
- Output: A .pdf file with the colored tree

2D Plots

Generates 2D PCoA plots using the principal coordinates file generated from beta diversity of and OTU table

```
make_2d_plots.py    –i    jackknifedbetadiversity/unweighted_unifrac/pcoa/    -m
map_file.txt  –b  Treatment  --ellipsoid_opacity=0.2  --ellipsoid_method=  IQR  –o
2dplots_jackknifed/
```

- Input: a file with the principal coordinate analysis or a folder (in case you use the output from jackknifed), map file, category between samples (ie. Treatment) –b

- Output: a folder with the 2D plots.