Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

_____ _____

Jiahui Xu Date

Comparison of Thresholding in QIIME and DADA2 for Analysis of Microbiome Data

By

Jiahui Xu
Degree to be awarded: MSPH
Master of Science in Public Health

Department of Biostatistics and Bioinformatics

Yi-Juan Hu, PhD (Thesis Advisor) Date

Glen Satten, PhD (Reader) Date

Comparison of Thresholding in QIIME and DADA2 for Analysis of Microbiome Data

By

Jiahui Xu
MSPH, Emory University, 2017
B.Sc., Nanjing University, 2013

Thesis Committee Chair: Yi-Juan Hu, PhD
Reader: Glen Satten, PhD

An abstract of
A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science in Public Health
in Department of Biostatistics and Bioinformatics
2017

Abstract

Comparison of Thresholding in QIIME and DADA2 for Analysis of Microbiome Data

BY
Jiahui Xu

For analyzing 16S rRNA gene sequences of the human microbiome, several bioinformatics software tools, such as QIIME, Mothur and DADA2, have been developed. In previous studies, QIIME and Mothur output significantly more spurious sequences than DADA2, which contained chimeric and nonchimeric errors, but QIIME ran relatively faster than DADA2. In this thesis, we intended to compare QIIME and DADA2 in preprocessing raw sequencing data and generating diversity indices and ordination. We also compared the diversity indices using different thresholds. Regarding computation time, QIIME tended to take less time than DADA2, partly due to skipping quality filtering and chimera removal, in addition to the substantive difference between the two. The numbers of taxa generated by QIIME were half the numbers of taxa by DADA2, due to QIIME's tendency to pool OTUs with less than 3% difference. Since DADA2 lost 40% reads after filtering and trimming, the resultant library sizes and taxa total counts were much smaller than those from QIIME, the community data had less richness, and the MDS-bray ordination plot showed a clean separation of three body sites without overlapping. Obviously, DADA2 lost much information including that vagina and rectum shared common strains. As the thresholds became more stringent, the data became less rich but more even. The ordination plots based on Bray-Curtis dissimilarity, with or without threshold, are very similar.

Comparison of Thresholding in QIIME and DADA2 for Analysis of Microbiome Data

By

Jiahui Xu
MSPH, Emory University, 2017
B.Sc., Nanjing University, 2013

Thesis Committee Chair: Yi-Juan Hu, PhD
Reader: Glen Satten, PhD

A thesis submitted to the Faculty of the
Rollins School of Public Health of Emory University
in partial fulfillment of the requirements for the degree of
Master of Science in Public Health
in Department of Biostatistics and Bioinformatics
2017

Acknowledgements

This thesis included data collected through funding provided by a grant from the National Institutes of Health, National Institute of Nursing Research (R01NR014800), awarded to Drs. Elizabeth J. Corwin and Anne L. Dunlop. I would like to thank these investigators as well as others on the award team including Drs. Carol J. Hogue, Bradley Pearce, Cherie C. Hill, Timothy D. Read, and Jennifer Mulle.

I would like to thank Professor Yi-Juan Hu for her expert guidance throughout this thesis, for providing advice for solving the problems I met, and for encouragement.

I would also like to thank Dr. Glen Satten, PhD, for asking questions about the results from DADA2 and suggesting further steps after obtaining preprocessed data from QIIME and DADA2, which gave me a comprehensive understanding of this study and insightful inspirations.

I am grateful to Kelly Ann Shaw who shared the QIIME scripts as a pipeline. Without her, I would be struggling to push QIIME through from scratch.

Finally, I would like to thank my family and friends, for any support I needed during this project.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

BV	Bacterial Vaginosis
IBD	Inflammatory Bowel Disease
MDS	Multidimensional Scaling
NGS	Next Generation Sequencing
OTU	Operational Taxonomic Unit
PCoA	Principal Coordinates Analysis
QIIME	Quantitative Insights Into Microbial Ecology
rRNA	Ribosomal Ribonucleic Acid
WMS	Whole-Metagenome Shotgun

1. INTRODUCTION

The human microbiome consists of the microbes (bacteria, archaea, viruses and fungi) that live in and on our bodies[1]. The microbial diversity within a given body habitat can be defined as the number and abundance distribution of different types of organisms, which has been associated with a number of human diseases[2]. For example, reduced diversity and/or imbalances of microbiota in the gut is linked to inflammatory bowel disease (IBD)[3][4][5][6] and obesity[7], and a more taxon-rich and diverse vaginal microbiota relates to bacterial vaginosis (BV)[8].

The development of next generation sequencing (NGS) technology has enabled investigations of the human microbiome, with remarkable resolution and throughput[9]. There are two main methods used for quantifying the composition of the human microbiome[10]: 16S ribosomal RNA (rRNA) gene amplicons and shotgun metagenomics. Inferences can be made by sequencing PCR amplicons from the 16S rRNA gene, whose domain is confined to bacteria and archaea[11], as the specific marker gene[12]. Since rRNA comprises 80% of total bacterial RNA, this approach allows for detecting rare species of the community with high sensitivity[10]. However, 16S rRNA-based sequencing may be biased due to unequal amplification of species' 16 rRNA genes[13] and it does not provide information about bacterial gene inventory and functionality[10]. Alternatively, shotgun metagenomics, also known as whole-metagenome shotgun (WMS) sequencing empowers researchers to thoroughly sample all genes in all organisms present in a given complex sample[14]. Metagenomic shotgun sequencing provides functional and biological process-level characterization of microbial communities, allows the reconstruction of draft genome sequences for a single community member, and makes

possible the detection of new species and new genes[10]. However, much deeper sequencing is required to achieve the same level of sensitivity in identifying rare taxa as 16S rRNA sequencing[10] and assembling metagenomics shotgun sequence data is very difficult.

To analyze 16S rRNA gene sequences, several bioinformatics methods have been introduced. Two remarkable ones are QIIME and Mothur. QIIME (pronounced *chime*), short for Quantitative Insights Into Microbial Ecology, is an open source software pipeline built by PyCogent toolkit[15] to deal with taking sequencing data from raw sequences to interpretation and database deposition[16] . Raw sequencing data can come from one or more sequencing technologies, such as Illumina, Roche/454, Sanger and others[17] . QIIME provides vast microbial community analyses and visualizations that have been essential to several recent high-profile studies, including network analysis, within- or between sample diversity and analysis of consistent representation of core sets of organisms in certain habitats[16]. Mothur is a single software platform written in C++, integrating the algorithms implemented in previous tools such as DOTUR, SONS, TreeClimber, LIBSHUFF, -LIBSHUFF and UniFrac and also overcoming the limitations of these online tools[18]. Mothur is intended to address the problems of transferring gigantic datasets across the Internet for analysis, expansion of the number of sequences, the relatively slow executing speed of code written in Python or Perl compared with code written in C and C++, and finally the integration and further development[18].

Besides QIIME and Mothur, there is a novel open-source software package DADA2 for modeling and correcting Illumina-sequenced amplicon errors. Sample composition is inferred by segregating amplicon reads into partitions consistent with the

error model[19]. DADA2 is reference free and can be applied to any genetic locus[19]. Comparing to DADA2, QIIME, which utilizes uclust OTU method, and Mothur, which implements average linkage OTU method, output significantly more spurious sequences, although this deficiency is reduced when merging reads[19]. The spurious output of QIIME and Mothur contains chimeric and nonchimeric errors[19]. Nevertheless, for the filtered Balanced forward reads (33,516 unique sequences), DADA2 (21 s) runs a little slower than QIIME (17 s) on a 2013 MacBook Pro[19].

The goal of this thesis is to compare QIIME and DADA2 in preprocessing raw sequencing data and generating diversity indices, and ordination, and to compare the diversity using different thresholds.

2. METHODOLOGY

2.1 Raw Sequencing Data

We obtained the raw sequencing data from a study of birth outcomes. The data included 16S sequence data from three body sites: vaginal, oral and rectal. The sample sizes in each body site were 366, 363 and 396, respectively. These samples were run in two batches by the core. The first batch had IDs like EIGCxxx and the second batch had IDs like MZWxxx.

2.2 Bioinformatics Analysis

We analyzed the paired-end 16S rRNA raw sequencing read files (i.e., fastq.gz files) using two bioinformatics pipelines: QIIME (Version 1.9.1) and DADA2 (Version 1.2.0). The analyses were run on a Linux cluster x86_64-redhat. The executing software for QIIME was Python (version 2.7) and for DADA2 was R (version 3.3.2).

2.2.1 QIIME

First, we used PANDAseq to join the paired-end reads, using arguments -f to read in FASTQ files containing forward reads, -r to read in FASTQ files containing reverse reads, -B to allow for unbarcoded sequences, -T to indicate the number of parallel threads, -N to eliminate all sequences with unknown nucleotides in the output, and -g to output log to a text file. Then we concatenated the preprocessed sequences into one file. We did not carry out demultiplexing and quality filtering because the data had already been split.

Second, we performed closed-reference OTU picking based on the combined file by pick_closed_reference_otus.py with the reference 97_otus.fasta and the taxonomy file 97_otu_taxonomy.txt. Specifically, pick_closed_reference_otus.py called two

subroutines, pick_otus.py and make_otu_table.py to pick the OTUs and make the OTU table, respectively.

The script pick_otus.py assigned similar sequences to operational taxonomic units (OTUs), by clustering sequences based on a user-defined similarity threshold (default was 0.97, roughly corresponding to species-level OTUs)[20]. The clustering method implemented in pick_otus.py was uclust_ref, which used UCLUST algorithm and a reference database as seeds of sequences which generated clusters based on percent identity[20]. The UCLUST algorithm employed the USEARCH algorithm as the subroutine[21]. The USEARCH algorithm searched a query sequence against target sequences and recorded the k-mers shared between the two sequences[22]. Then UCLUST worked for the clustering aspect. In uclust_ref, a reference database of 16s reads was used to generate the centroids. Each cluster centroid (target sequence) had a level of similarity below a pre-specified threshold level with each other centroid[22]. The query sequences were assigned to a centroid based on identity threshold[22]. We suppressed the creation of new clusters so reads not aligning to the reference centroids were discarded.

The script make_otu_table.py took the result of pick_otus.py as the input. It tabulated the number of times an OTU was found in each sample, adding the taxonomic predictions for each OTU in the last column if a taxonomy file was supplied[23].

2.2.2 DADA2

The DADA2 workflow for paired-end sequencing data required R packages ‘dada2’ (version 1.2.0) and ‘phyloseq’ (version 1.19.1)[24]. The workflow included filtering and trimming, sample inference, merging paired-end reads, constructing sequence table, removing chimeras and assigning taxonomy.

First, we plotted and examined the quality profiles, i.e., the distribution of quality scores as a function of sequence position, of forward and reverse reads using plotQualityProfile function. We performed raw read filtering based on several user-definable criteria using fastqPairedFilter function. The quality-score threshold for read bases were set to 20, which roughly corresponds to removing trimLeft = c(0,0) bases from the start of each read and truncating reads after truncLen = c(290,220) bases for forward and reverse reads. After truncation, reads that had more errors than the maximum number of expected errors allowed (maxEE) would be discarded. Expected errors (EE) were calculated from the nominal definition of the quality score: $EE = sum(10^{-Q/10})$[25], the sum of the error probabilities. The error probability can be calculated from the Quality or Q score by $10^{-Q/10}$ [26]. We used the default value, Inf, for maxEE, which meant no EE filtering. Reads were truncated at the first instance of a quality score less than or equal to a value specified by truncQ[25]. We used the default truncQ=0. Meanwhile, sequences with more than maxN Ns were discarded. Since the following dada function did not allow N to be the value of a base, we set the maximum N's allowed (maxN) to 0.

In the stage of sample inference, we first included all target samples to iteratively estimate the error rates and infer the sequence variants. But due to the computational limitations, the job got killed at this step. Therefore, we drew a subset of samples from the filtered data to estimate forward and reverse error rates, 25 samples for each. We applied derepFastq function to dereplicate the sequences, which substantially reduced computation time by eliminating redundant comparisons[27]. dada function took as input dereplicated amplicon sequencing reads and returned the inferred composition of the samples[28]. Since selfConsist = TRUE, the algorithm would alternate between sample

inference and error rate estimation until convergence[28]. err can be set to NULL and an initial error rate matrix would be estimated from the data by assuming that all reads were errors away from one true sequence[28]. Then we assigned the error rates to every single filtered file by using derepFastq function and dada function. We attempted to merge each denoised pair of forward and reverse reads using mergePairs function, rejecting any pairs which did not sufficiently overlap or which contained too many (>0 by default) mismatches in the overlap region[29].

We constructed a sequence table that resembled the "OTU table" produced by classical methods[27] using makeSequenceTable function. For chimeric sequences removal, we used removeBimeraDenovo function. assignTaxonomy function utilized Ribosomal Database Project (RDP Training Set 14) to assign taxonomy. We assigned species against the RDP species-level training set using assignSpecies function. We combined the OTU table (converted from the sequence table), sample names and taxonomy using phyloseq function.

2.3 Analysis of Diversity Based on Thresholds

This analysis was implemented on R (version 3.3.2) and required package 'phyloseq' (version 1.19.1), 'ggplot2' (version 2.2.1), 'data.table' (version 1.10.0) and 'gridExtra' (version 2.2.1).

For the results from DADA2, we loaded *.RData files, which contained the Phyloseq objects, as ps.vaginal, ps.oral and ps.rectal. Typing ps.vaginal, for example, in the RStudio Console gave the summary statistics of the dimensions of the object, including the OTU table, the Sample Data and the Taxonomy Table. In order to distinguish samples in the merged Phyloseq object, we added "SampleType" to each

Sample Data. Then we merged three Phyloseq objects into one by using merge_phyloseq function. To summarize the library size, i.e., the total number of reads per sample, we utilized functions quantile and mean on results from sample_sums function. To summarize taxa (OTUs) total counts, we utilized functions quantile and mean on results from taxa_sums function. Before we moved on, we calculated the number of 0 counts, the number of singletons and the number of doubletons among results from taxa_sums function. The definition of a singleton was a read with a sequence that was present exactly once, i.e., was unique among the reads[30]. Although singletons could be rare variants detected by the pipelines and removing them may reduce sensitivity, many suggested singletons should be discarded, because if sequencer errors were independent and randomly distributed, then the sequence in a bad read was improbable to be reproduced by chance and most singletons would include at least one error[30]. Here we adopted four approaches to set the thresholds, one maintaining the original OTU table, one removing singletons, one removing singletons and doubletons and one setting 1% of average OTU frequencies as threshold. Thus, we chose whether to perform taxa filtering or not before proceeding to diversity analyses. We visualized the alpha diversity using plot_richness function. To compare the dissimilarity among three body sites, we perform the approach of MDS (PCoA) ordination with Bray-Curtis distance using ordinate and plot_ordination functions.

For the results from QIIME, the analysis required additional packages ‘biom’ (version 0.3.12) and ‘qiimer’ (version 0.9.4) to preprocess the data. After reading the data from QIIME, we transposed the OTU table from OTU by Sample ID to Sample ID by OTU. We created a mapping file matching Sample IDs and sample names, extracted the

taxonomy for each OTU and then integrated the above three objects into a phyloseq object to conduct analysis. The steps of creating the mapping file in Excel were as follows:

1. Copy forward file names to the second column and create SampleID in the first column
2. Save as link_rectal.csv in the format CSV UTF-8 (Comma delimited) (.csv)
3. Open the CSV file with vim; type : %s/Ctrl-V Ctrl-M/\r/g then :%s/tab/,/g
4. In terminal, run sh link.sh and get link1_vaginal.csv
5. Add header to link1_vaginal.csv and save as mapping_vaginal.txt in the format of Tab Delimited Text (.txt)

3. RESULTS

Table 1 presented the computation time in seconds for processing different body sites in QIIME and DADA2, along with sample sizes and numbers of taxa in the OTU table.

Table 1 Computation time, number of taxa in the OTU table and sample size for each body site in either QIIME or DADA2 pipeline

Body Site	Computation Time (in seconds)		Number of Taxa		Sample size
	QIIME	DADA2	QIIME	DADA2	
Vaginal	113319.01	111025.61	10215	20068	366
Oral	93448.92	114388.13	9934	16004	363
Rectal	96985.30	173649.57	13103	22350	396

3.1 Results from QIIME

Paired-end reads were joined in alphabetically sorted order by PANDAseq. PANDAseq also created log files for each sample in the directory OUTPUT/pandafiles/pandalogs/. Meanwhile, a mapping file map.txt was generated in the directory OUTPUT/useful_files/ based on file names. The mapping file contained information about the samples, with header #SampleID, BarcodeSequence, LinkerPrimerSequence, SampleType and Description. #SampleID was the sequence number of a sample. BarcodeSequence and LinkerPrimerSequence were left empty. SampleType was the name of the forward file and Description was the name of the reverse file. Afterwards, a log file paired_end_joining_stats.txt summarizing PANDAseq logging statistics were produced in the directory OUTPUT/useful_files/, with header File, READS, NOALGN, LOWQ, HASN and ENDTOTAL.

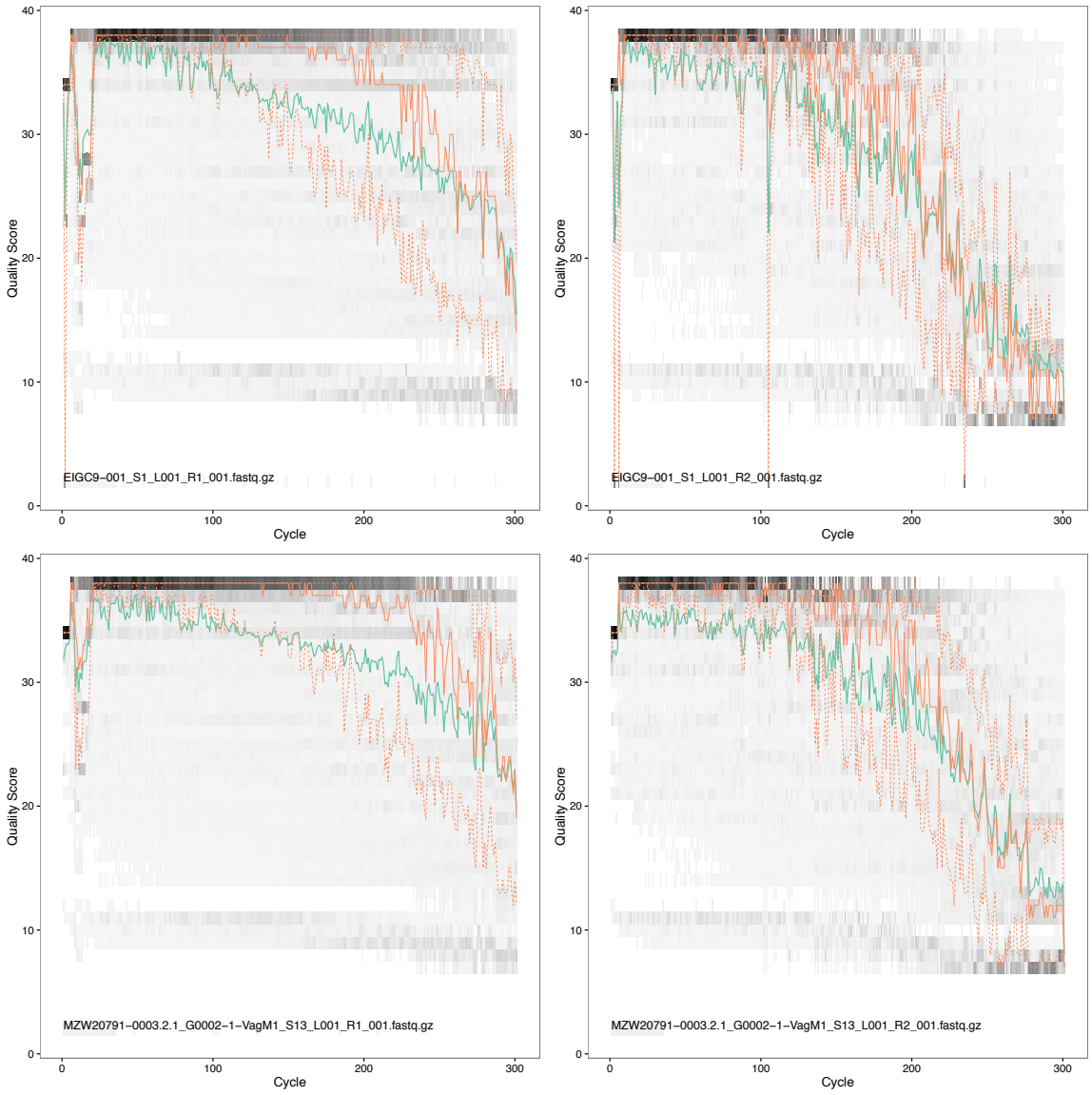
Joined sequences were then processed into the form of .fna files written to the directory OUTPUT/pandafiles/QIIME_input/ with header of each sequence changed and

the number of sequences in each sample were output to the QIIME log file. All .fna files were concatenated into a single file called combined_sequences.txt, which was ready for QIIME to use.

Performing pick_closed_reference_otus.py output a directory OUTPUT/closed_ref_OTUs. This directory consisted of a log file of the whole OTU picking process, a uclust_ref_picked_otus folder and representative sequences OTU_rep_sequences.fna created by pick_otus.py and the final outcome otu_table.biom generated by make_otu_table.py. The uclust_ref_picked_otus folder encompassed four files: combined_sequences_clusters.uc, combined_sequences_otus.log, combined_sequences_failures.txt and combined_sequences_otus.txt. combined_sequences_clusters.uc included the clustering information. combined_sequences_otus.log was the log file for pick_otus.py. combined_sequences_failures.txt included failures. combined_sequences_otus.txt was the picked OTUs. otu_table.biom was the OTU table in the form of OTU by Sample ID.

3.2 Results from DADA2

Figure 1 Quality profiles for forward and reverse reads from a sample from the EIGC batch and a sample from the MZW batch



The quality profiles obtained, shown in Figure 1, indicated that the forward reads had better quality than the reverse reads, especially at the end. Filtered forward reads and reverse reads were separately stored into the folders FWD_filtered and REV_filtered. After filtering and trimming, the percent of paired sequences output from the read-ins was approximately 60%. Since initial error matrix unspecified, error rate of 25 samples was initialized to the maximum possible estimate from the data and it converged within 10 rounds. After sample inferring, merging, constructing the sequence table and

removing chimeras, the reads were made into a sequence table and saved as seqtab_*.rds. The taxonomy table was in the form of sequence by taxonomic rank, with the column names “Kingdom”, “Phylum”, “Class”, “Order”, “Family”, “Genus” and “Species”. A final phyloseq object containing the OTU table, sample data and the taxonomy table was saved as phyloseq_*.RData.

3.3 Results from Analysis of Diversity Based on Thresholds

There were 20068 OTUs in the vaginal sample, 16004 OTUs in the oral sample and 22350 OTUs in the rectal sample from DADA2 results. And there were 10215 OTUs in the vaginal sample, 9934 OTUs in the oral sample and 13103 OTUs in the rectal sample from QIIME results.

Table 2 Distribution of library size

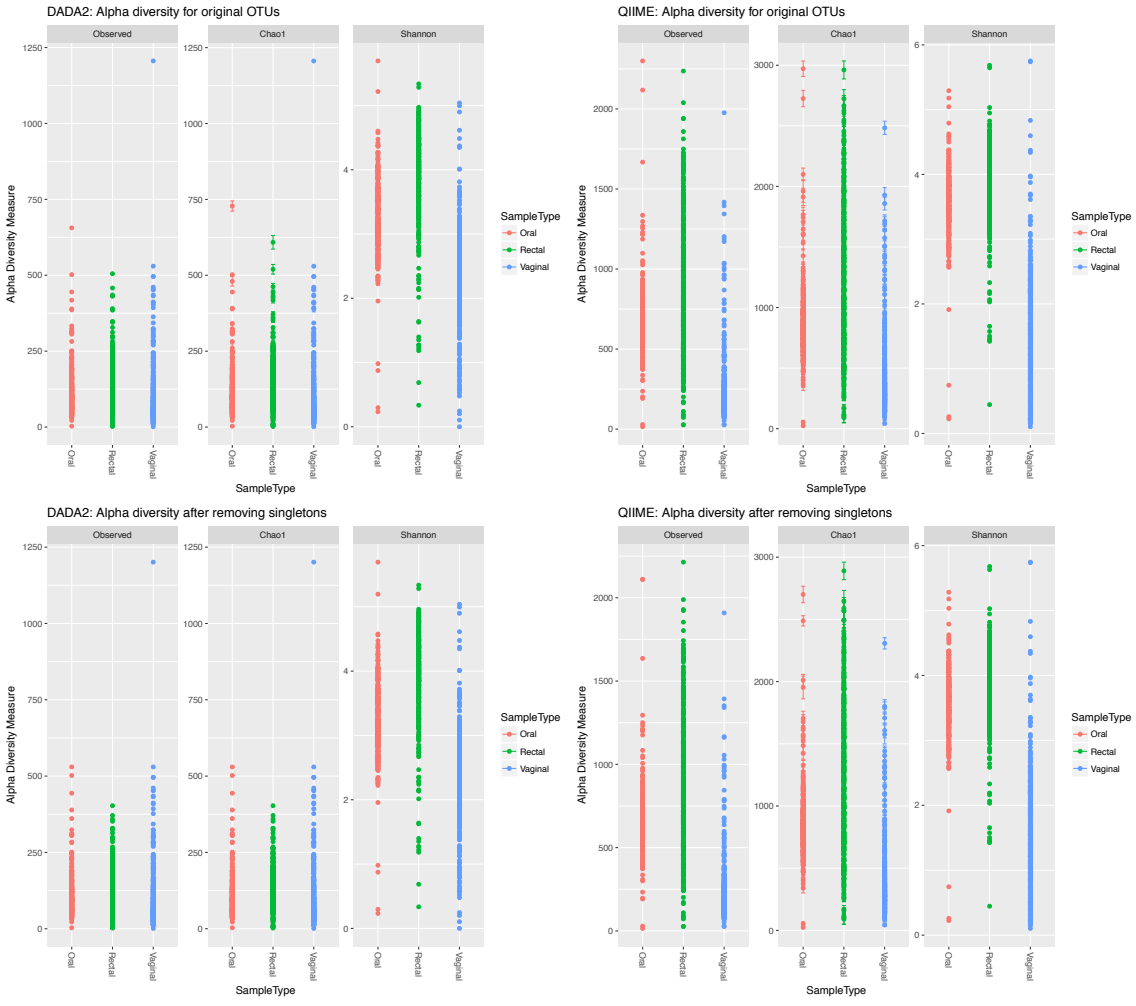
Pipeline	Body Site	Minimum	1st Quartile	Median	3rd Quartile	Maximum	Mean
DADA2	Total	0	11377	19466	31972	244084	23650.56
	Vaginal	0	12863.75	22304	34647.75	244084	26637.84
	Oral	0	12957.5	21081	32241.5	93593	23863.44
	Rectal	0	9404	16215	29565.75	114818	20694.46
QIIME	Total	25	70720	110765	153287	769526	117054.9
	Vaginal	53	55791	93806.5	139735.2	769526	103788.9
	Oral	25	74927	114050	152016.5	408131	118786.6
	Rectal	37	83064.25	125202.5	165466.75	432707	127728.5

Table 3 Distribution of taxa total counts

Pipeline	Body Site	Minimum	1st Quartile	Median	3rd Quartile	Maximum	Mean
DADA2	Total	1	2	4	13	1797594	474.3859
	Vaginal	1	2	4	10	1719689	485.8206
	Oral	1	2	4	16	1336848	541.2664
	Rectal	1	2	3	18	293111	366.6669
QIIME	Total	1	2	9	70	13679891	7372.868
	Vaginal	1	1	4	17	12528654	3718.723
	Oral	1	1	4	24	6373605	4340.603
	Rectal	1	2	9	66	2936182	3860.222

In the phyloseq object from DADA2, samples EIGC9-025_S25, EIGC9-026_S26, EIGC9-027_S27, EIGC9-028_S28, EIGC9-029_S29, EIGC9-030_S30 had 0 total number of reads, while no samples in the phyloseq object from QIIME had 0 total number of reads. The distribution of library size was given in Table 2. Table 3 recorded the distribution of taxa total counts. In DADA2 results, there were 5724 singletons and 12695 doubletons, and in QIIME results, there were 3182 singletons and 1692 doubletons.

Figure 2 Alpha diversity by measures of 'observed', 'Chao1' and 'Shannon' for different body sites, using three thresholding approaches



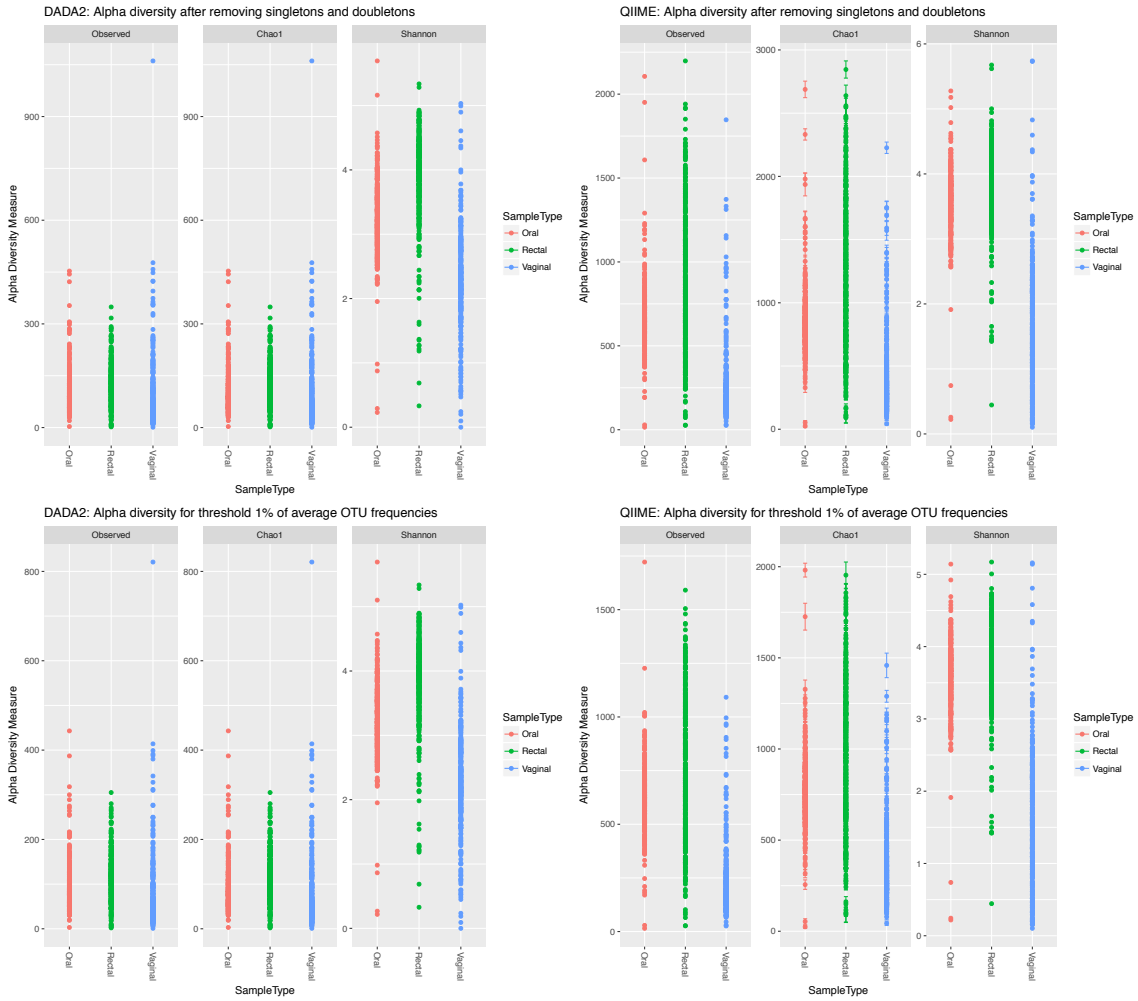
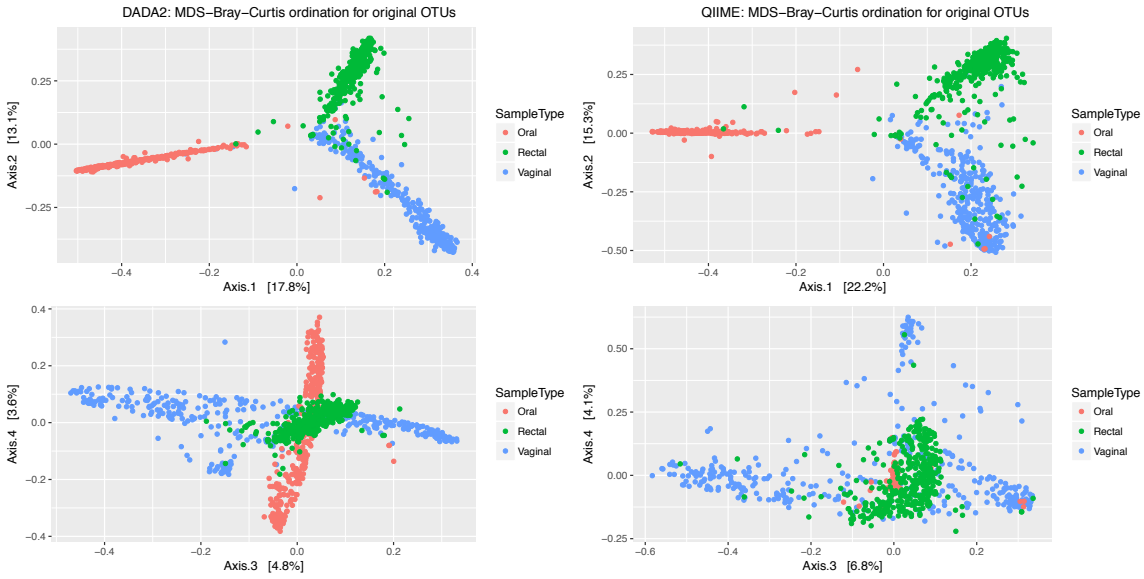


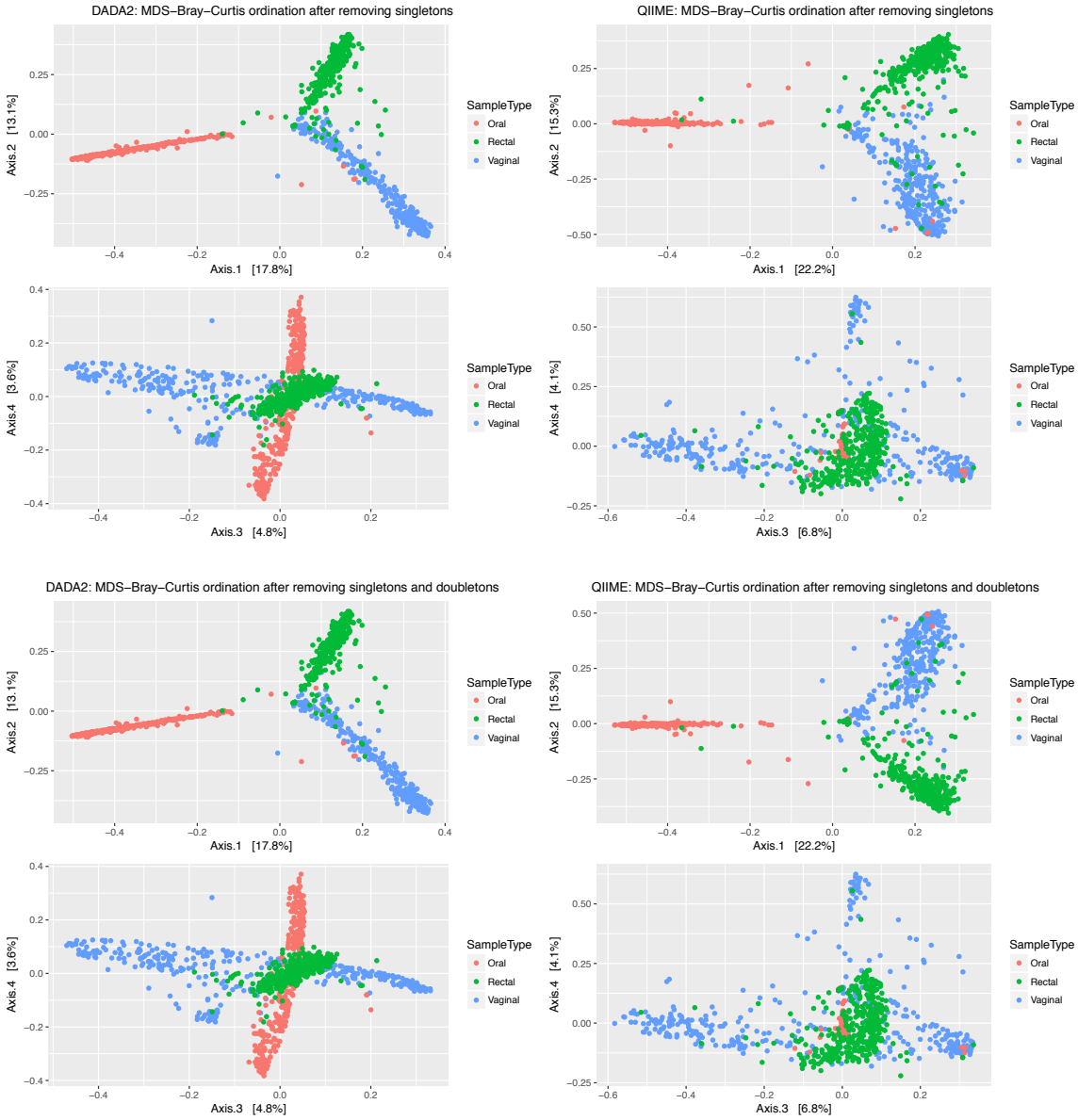
Figure 2 presented three kinds of measurements of alpha diversity, stratified by SampleType, using four thresholding approaches. According to this figure, oral and rectal samples showed more richness and evenness than vaginal samples. Moreover, data preprocessed by QIIME had more richness and unevenness than data from DADA2. As the thresholds became more restricted, the data became less rich but evener.

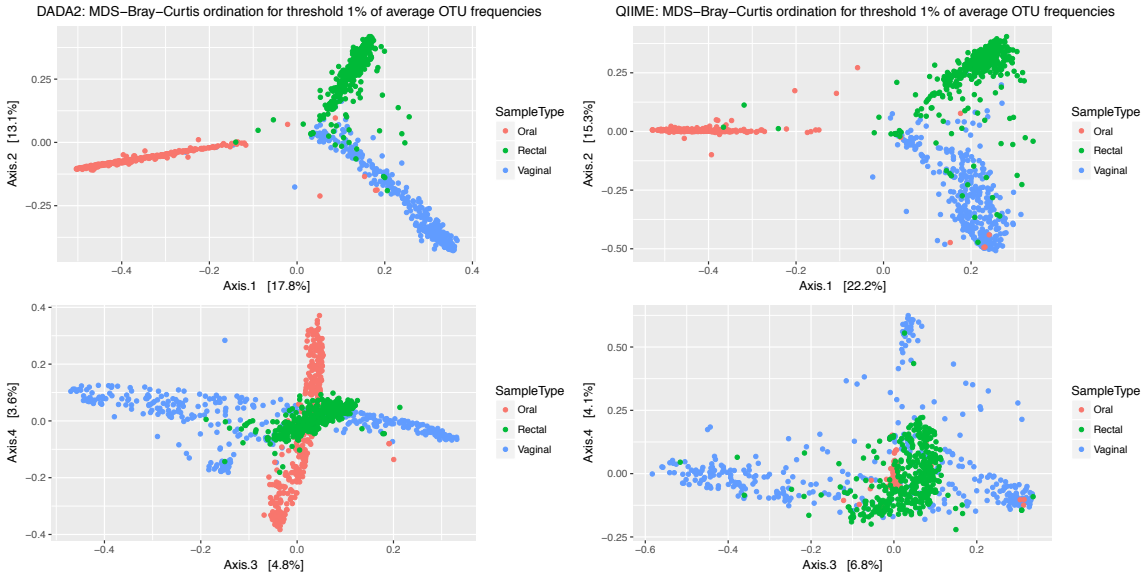
Figure 3 were MDS Bray-Curtis plots which indicated that three kinds of samples fell into three distinct clusters. To summarize the variability in the dataset, MDS produced a set of uncorrelated axes, each of which had an eigenvalue[31]. The magnitude of the eigenvalue indicated the amount of variation captured in that axis[31]. The relative importance of each axis was determined by the percent of its eigenvalue to the sum of all

eigenvalues[31]. Capturing 30.9% of total eigenvalue for DADA2 results and 37.5% of total eigenvalue for QIIME results, Axis 1 and Axis 2 revealed more importance than other axes. The Axis 1-Axis 2 plot for DADA2 results showed a clean separation of three body sites, while the plot for QIIME results showed a small overlap between rectal and vaginal samples. As we can see, there was no megascopic difference among plots using the first three thresholding approaches, but the fourth threshold induced changes in the relative position of samples in the ordination plot.

Figure 3 MDS-Bray-Curtis ordination for different body sites, using three thresholding approaches




DADA2: MDS−Bray−Curtis ordination after removing singletons
QIIME: MDS−Bray−Curtis ordination after removing singletons
DADA2: MDS−Bray−Curtis ordination after removing singletons and doubletons
QIIME: MDS−Bray−Curtis ordination after removing singletons and doubletons
Axis.1 [17.8%]
Axis.2 [13.1%]
Axis.1 [22.2%]
Axis.2 [15.3%]
Axis.3 [4.8%]
Axis.4 [3.6%]
Axis.3 [6.8%]
Axis.4 [4.1%]
SampleType
Oral
Rectal
Vaginal


DADA2: MDS−Bray−Curtis ordination for threshold 1% of average OTU frequencies
QIIME: MDS−Bray−Curtis ordination for threshold 1% of average OTU frequencies
Axis.1 [17.8%]
Axis.2 [13.1%]
Axis.3 [4.8%]
Axis.4 [3.6%]
Axis.1 [22.2%]
Axis.2 [15.3%]
Axis.3 [6.8%]
Axis.4 [4.1%]
SampleType
Oral
Rectal
Vaginal

4. DISCUSSION

In Table 1, oral samples and rectal samples had a tendency of increasing computation time for both pipelines as sample size increases, in accordance with common sense. However, with an intermediate sample size, vaginal samples had the longest computation time for QIIME and shortest computation time for DADA2. It looked weird but might trace back to the internal mechanism within two pipelines and the property of vaginal samples. As mentioned in 3.3, vaginal samples had less richness and evenness in biological diversity than samples from the other two body sites. QIIME tended to have less computation time than DADA2, partly due to skipping steps of quality filtering and removing chimeras when running QIIME, in addition to the substantial difference between the two. The numbers of taxa for both pipelines seemed to match the sample size in all three body sites. Noticeably, the numbers of taxa for QIIME were half the numbers of taxa for DADA2, resulting from the fact that two OTUs that were less than 3% different would be pooled together by QIIME, while DADA2 tended to keep them as separate, especially if they differed at a single locus.

In Table 2 and Table 3, library sizes and taxa total counts in DADA2 results were much smaller than those in QIIME results, which was caused by the loss of 40% reads after filtering and trimming in DADA2. For the same reason, data preprocessed by QIIME had more richness than data from DADA2 in Figure 2 and the plot from DADA2 showed a clean separation of three body sites without overlapping in Figure 3. It had been suggested that a part of the vaginal microbiota probably originated from the rectal microbiota at strain level in women[32][33], based on a study[32] showing that 44% of the isolated strains from the vaginal sample were shared in both the vagina and the rectum.

Therefore, data preprocessed by DADA2 lost this information. On the other hand, however, the overlapping in the plot from QIIME might result from skipping the steps of chimeras removal.

In Figure 3, there was no megascopic difference among plots using the first three thresholding approaches, because of the calculation method used by the distance of Bray-Curtis dissimilarity. As defined by J. R. Bray and J. T. Curtis[34], the index of dissimilarity was:

$$BC_{ij} = 1 - \frac{2C_{ij}}{S_i + S_j}$$ [35]

where C_{ij} was the sum of the lesser value for only those taxa shared between two samples. S_i and S_j were the total number of taxa counted in each sample. Therefore, singletons and doubletons contributed little to the dissimilarity indices for samples with large OTUs, for example, samples from the QIIME pipeline. With significant changes in the numerator and the denominator, the fourth threshold induced changes in the relative position of samples in the ordination plot.

In DADA2, we did not set the quality score to 20 initially. But instead, we used quality score 30, with parameters maxN=0, maxEE=2, truncQ=2, which yielded 43% paired sequences output from the read-ins, and with parameters maxN=0, maxEE=Inf, truncQ=2, which yielded 60%. Trying quality score 10 yielded 0.3~0.6% for maxEE=2 and 15% for maxEE=Inf. Balancing between the chance of correctness of each base call and the percent of output sequences for overlapping in the later merging stage, we chose quality score 20, with parameter settings maxEE=Inf, truncQ=0, maxN=0.

A problem of running DADA2 on a cluster was that although we set.seed(100), we still got different OTU tables at different running rounds, i.e., results not reproducible,

because of sampling different samples to estimate error rates. This problem may be caused by the cluster system generating seeds based on time.

4.1. Strengths and Limitations

We employed pick_closed_reference_otus.py for QIIME. There were pros and cons in this approach. The advantages were speed and better trees and taxonomy. Closed-reference OTU picking was fully parallelizable, so that it was useful for extremely large data sets[36]. However, creation of new clusters was suppressed in the algorithm, therefore reads not aligning to reference centroids were eliminated, which made this method too conservative and led to a dramatic reduction in taxa.

4.2 Challenges

When running the analyses, we faced several challenges. The major one was the lack of memory or disk space induced by large datasets. For DADA2, we abandoned the regular tutorial and turned to *A DADA2 workflow for Big Data: Paired-end* (http://benjjneb.github.io/dada2/bigdata_paired.html), which subset the samples to get the estimated error rate and performed the inference separately on each individual, instead of doing estimating and inference on the entire samples. For QIIME, to address the problem of 'no space left on device' in the pick_otus.py step, we googled for the answer. According to Google Groups (https://groups.google.com/forum/#!topic/qiime-forum/7IHqiUwmE0E), we needed to change the temp_dir parameter to our own folder for temporary files in the QIIME configuration file, but it did not work. Then we found it as a bug when searching the Github: https://github.com/biocore/qiime/issues/2049. The solution was quite simple, just adding echo "export TMPDIR=/Pathname/tmp" >>

script.out to qsub_py.sh that executed the jobs (https://github.com/biocore/burrito-fillings/issues/55#issuecomment-91041061).

When merging three Phyloseq objects generated from QIIME, samples with the same SampleID, for example, Sample0, were combined into one row, which was incorrect for distinct samples. We changed the row names of the OTU table, i.e., from SampleID to sample names to avoid this issue.

4.3 Recommendations

We can introduce open-reference otu picking method in the revision of QIIME scripts. In an open-reference OTU picking process, reads are clustered against a reference sequence collection and any reads not hitting the reference sequence collection will be subsequently clustered de novo[36].

For results from both pipelines, many names of species in the taxonomy table were NAs. In future study, we can try to download all the target genus samples available in the NIH databases and use the consensus sequence for these OTUs to make our own assignments.

When running QIIME pipeline, we skipped the steps of quality filtering and chimeras removal. In later studies, we would suggest add these parts to the scripts. The mapping file generated in the QIIME procedure seemed useless in the downstream analyses. We would suggest include SampleID and SampleType in the file to save time for making a mapping file by ourselves.

5. REFERENCE

[1] Shreiner A. B., Kao J. Y., Young V. B. (2015). The gut microbiome in health and in disease. Curr. Opin. Gastroenterol. 31, 69–75.

[2] The Human Microbiome Project Consortium (2012). Structure, function and diversity of the healthy human microbiome. Nature 486, 207–214.

[3] Knights D., Lassen K. G., Xavier R. J. (2013). Advances in inflammatory bowel disease pathogenesis: linking host genetics and the microbiome. Gut 62, 1505–1510.

[4] Huttenhower C., Kostic A. D., Xavier R. J. (2014). Inflammatory bowel disease as a model for translating the microbiome. Immunity 40, 843–854.

[5] Kostic A. D., Xavier R. J., Gevers D. (2014). The microbiome in inflammatory bowel disease: current status and the future ahead. Gastroenterology 146, 1489–1499.

[6] Norman J. M., Handley S. A., Baldridge M. T., Droit L., Liu C. Y., Keller B. C., et al. . (2015). Disease-specific alterations in the enteric virome in inflammatory bowel disease. Cell 160, 447–460.

[7] Turnbaugh P. J., Hamady M., Yatsunenko T., Cantarel B. L., Duncan A., Ley R. E., et al. . (2009). A core gut microbiome in obese and lean twins. Nature 457, 480–484.

[8] Oakley B.B., Fiedler T.L., Marrazzo J.M., Fredricks D.N. (2008). Diversity of human vaginal bacterial communities and associations with clinically defined bacterial vaginosis. Applied and Environmental Microbiology 74(15), 4898–4909.

[9] Jovel J., Patterson J., Wang W., Hotte N., O'Keefe S., Mitchel T., et al (2016). Characterization of the gut microbiome using 16S or shotgun metagenomics. Front Microbiol 7, 459.

[10] Li, H. (2015). Microbiome, Metagenomics, and High-Dimensional Compositional Data Analysis. Annual Review of Statistics and Its Application, 2(1), 73-94.

[11] Janda, J.M. and Abbott, S.L. (2007) 16S rRNA Gene Sequencing for Bacterial Identification in the Diagnostic Laboratory: Pluses, Perils, and Pitfalls. Journal of Clinical Microbiology 45, 2761-2764.

[12] Tringe, S. G., & Rubin, E. M. (2005). Metagenomics: DNA sequencing of environmental samples. Nature Reviews Genetics, 6(11), 805-814. doi:10.1038/nrg1709

[13] Shah, N., Tang, H., Doak, T. G., & Ye, Y. (2010). COMPARING BACTERIAL COMMUNITIES INFERRED FROM 16S rRNA GENE SEQUENCING AND SHOTGUN METAGENOMICS. Biocomputing 2011, 165-176. doi:10.1142/9789814335058_0018

[14] Shotgun Metagenomic Sequencing. Retrieved March 22, 2017, from https://www.illumina.com/areas-of-interest/microbiology/microbial-sequencing-methods/shotgun-metagenomic-sequencing.html

[15] Knight, R., Maxwell, P., Birmingham, A., Carnes, J., Caporaso, J. G., Easton, B. C., . . . Huttley, G. A. (2007). PyCogent: a toolkit for making sense from sequence. Genome Biology, 8(8). doi:10.1186/gb-2007-8-8-r171

[16] Caporaso, J. G. et al. (2010). QIIME allows analysis of high-throughput community sequencing data. Nature Methods 7, 335–336.

[17] Kuczynski, J., Lauber, C. L., Walters, W. A., Parfrey, L. W., Clemente, J. C., Gevers, D., & Knight, R. (2011). Experimental and analytical tools for studying the human microbiome. Nature Reviews Genetics, 13(1), 47-58. doi:10.1038/nrg3129

[18] Schloss P.D., Westcott S.L., Ryabin T, Hall J.R., Hartmann M, Hollister E.B., Lesniewski R.A., Oakley B.B., Parks D.H., Robinson C.J., et al. (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities, Appl. Environ. Microbiol., 75, 7537-7541.

[19] Callahan, B. J., Mcmurdie, P. J., Rosen, M. J., Han, A. W., Johnson, A. J., & Holmes, S. P. (2016). DADA2: High-resolution sample inference from Illumina amplicon data. Nature Methods, 13(7), 581-583. doi:10.1038/nmeth.3869

[20] Pick_otus.py – OTU picking. Retrieved March 22, 2017, from http://qiime.org/scripts/pick_otus.html

[21] Edgar, R.C. (2010) Search and clustering orders of magnitude faster than BLAST, Bioinformatics 26(19), 2460-2461. doi: 10.1093/bioinformatics/btq461

[22] Plummer E, Twin J, Bulach DM, Garland SM, Tabrizi SN (2015) A Comparison of Three Bioinformatics Pipelines for the Analysis of Preterm Gut Microbiota using 16S rRNA Gene Sequencing Data. J Proteomics Bioinform 8:283-291. doi:10.4172/jpb.1000381

[23] Make_otu_table.py – Make OTU table. Retrieved March 22, 2017, from http://qiime.org/scripts/make_otu_table.html

[24] McMurdie PJ, Holmes S (2013) phyloseq: An R Package for Reproducible Interactive Analysis and Graphics of Microbiome Census Data. PLoS ONE 8(4): e61217. doi:10.1371/journal.pone.0061217

[25] fastqPairedFilter. Retrieved March 22, 2017, from https://www.rdocumentation.org/packages/dada2/versions/1.0.3/topics/fastqPairedFilter

[26] Edgar, R. C., & Flyvbjerg, H. (2015). Error filtering, pair assembly and error correction for next-generation sequencing reads. Bioinformatics, 31(21), 3476-3482. doi: 10.1093/bioinformatics/btv401

[27] Callahan, B. DADA2 Pipeline Tutorial (1.2). Retrieved March 22, 2017, from http://benjjneb.github.io/dada2/tutorial.html

[28] dada. Retrieved March 22, 2017, from https://www.rdocumentation.org/packages/dada2/versions/1.0.3/topics/dada

[29] mergePairs. Retrieved March 22, 2017, from https://www.rdocumentation.org/packages/dada2/versions/1.0.3/topics/mergePairs

[30] Singletons. Retrieved March 22, 2017, from http://www.drive5.com/usearch/manual/singletons.html

[31] Principal coordinates analysis - GUSTA ME. (n.d.). Retrieved March 30, 2017, from https://sites.google.com/site/mb3gustame/dissimilarity-based-methods/principal-coordinates-analysis

[32] El Aila NA, Tency I, Claeys G, Verstraelen H, Saerens B, Santiago GLDS, De Backer E, Cools P, Temmerman M, Verhelst R, Vaneechoutte M. Identification and genotyping of bacteria from paired vaginal and rectal samples from pregnant women indicates similarity between vaginal and rectal microflora. BMC Infect Dis. 2009;9:167. doi: 10.1186/1471-2334-9-167.

[33] Marrazzo JM, Antonio M, Agnew K, Hillier SL. Distribution of genital Lactobacillus strains shared by female sex partners. J Infect Dis. 2009;199:680–683. doi: 10.1086/596632.

[34] Bray, J. R., & Curtis, J. T. (1957). An Ordination of the Upland Forest Communities of Southern Wisconsin. Ecological Monographs, 27(4), 325-349. doi:10.2307/1942268

[35] Bray–Curtis dissimilarity. (2017, March 02). Retrieved March 30, 2017, from https://en.wikipedia.org/wiki/Bray%E2%80%93Curtis_dissimilarity

[36] OTU picking strategies in QIIME. Retrieved March 22, 2017, from http://qiime.org/tutorials/otu_picking.html

6. APPENDIX

Script 1. Moving files into folders according to a mapping file

```
#!/bin/bash
input="/Pathname/pregnant_women_EIGCstudyIDlinks_091616.csv"
# Set "," as the field separator using $IFS
# and read line by line using while read combo
while IFS="," read -r f1 f2 f3 f4 f5
do
  if [ "$f4" == "Vaginal " ]; then
    cp /Pathname/PTB_0816/"$f1"* /Pathname/vaginal
  elif [ "$f4" == "Oral" ]; then
    cp /Pathname/PTB_0816/"$f1"* /Pathname/oral
  elif [ "$f4" == "Rectal" ]; then
    cp /Pathname/PTB_0816/"$f1"* /Pathname/rectal
  fi
done < "$input"
```

Script 2. Submission script for QIIME: qsub_py.sh

```
#!/bin/bash

	echo "#!/bin/bash" >> script.out
	echo "#$ -N QIIME" >> script.out
	echo "#$ -cwd" >> script.out
	echo "#$ -j y" >> script.out
	echo "export PATH=$PATH:/usr/local/python27/bin" >> script.out
	echo "export QIIME_CONFIG_FP=$HOME/.qiime_config" >> script.out
	echo "export TMPDIR=/Pathname/tmp" >> script.out
	echo "/home/jxu238/my_app/bin/python2.7 QIIME_vaginal.py
/Pathname/vaginal" >> script.out
	chmod +x script.out
	qsub -q gene.q ./script.out
	rm -rf script.out
```

Script 3. R code for DADA2 big data

```
library(dada2);packageVersion("dada2")

# File parsing
path <- "/Pathname/vaginal" # Change to the directory containing the fastq files

fns <- list.files(path)
fns <- sort(fns)

fastqFs <- fns[grepl("_R1",fns)] # Just the forward read files
fastqRs <- fns[grepl("_R2",fns)] # Just the reverse read files
if(length(fastqFs) != length(fastqRs)) stop("Forward and reverse files do not match.")

filtpathF <- file.path("/Pathname/vaginal_bigdata","FWD_filtered")
filtpathR <- file.path("/Pathname/vaginal_bigdata","REV_filtered")
dir.create(filtpathF)
dir.create(filtpathR)

######################
##    Quality Profiles    ##
######################
pdf("Quality_profiles_vaginal.pdf",width=7,height=7)
# Visualize the quality profile of the forward reads
plotQualityProfile(fastqFs[[1]])
plotQualityProfile(fastqFs[[2]])
# Visualize the quality profile of the reverse reads
plotQualityProfile(fastqRs[[1]])
plotQualityProfile(fastqRs[[2]])
dev.off()

################
##    Filtering    ##
################
for(i in seq_along(fastqFs)){
  fqF <- fastqFs[[i]]
  fqR <- fastqRs[[i]]
  fastqPairedFilter(c(file.path(path,fqF), file.path(path, fqR)),
            c(file.path(filtpathF, fqF), file.path(filtpathR, fqR)),
            trimLeft=c(0,0),truncLen=c(290,220), # trim to meet the lowest acceptable
quality score at 20
            maxEE=Inf, truncQ=0, maxN=0,
            compress=TRUE, verbose=TRUE)
}

####################
```

```
## Sample Inference ##
####################
filtFs <- list.files(filtpathF, full.names = TRUE)
filtRs <- list.files(filtpathR, full.names = TRUE)
sample.names <- sapply(strsplit(basename(filtFs),"_L001"),`[`,1) # Get sample names
from the forward read filenames
sample.namesR <- sapply(strsplit(basename(filtRs),"_L001"),`[`,1)
if(!identical(sample.names, sample.namesR)) stop("Forward and reverse files do not
match.")
names(filtFs) <- sample.names
names(filtRs) <- sample.names

set.seed(100)
# Learn forward error rates
NSAM.LEARN <- 25 # Choose enough samples to have at least 1M reads
drp.learnF <- derepFastq(sample(filtFs, NSAM.LEARN))
dd.learnF <- dada(drp.learnF, err=NULL, selfConsist=TRUE, multithread=TRUE)
errF <- dd.learnF[[1]]$err_out
rm(drp.learnF);rm(dd.learnF)
# Learn reverse error rates
drp.learnR <- derepFastq(sample(filtRs, NSAM.LEARN))
dd.learnR <- dada(drp.learnR, err=NULL, selfConsist=TRUE, multithread=TRUE)
errR <- dd.learnR[[1]]$err_out
rm(drp.learnR);rm(dd.learnR)

# Sample inference and merger of paired-end reads
mergers <- vector("list", length(sample.names))
names(mergers) <- sample.names
for(sam in sample.names) {
  cat("Processing:", sam, "\n")
    derepF <- derepFastq(filtFs[[sam]])
    ddF <- dada(derepF, err=errF, multithread=TRUE)
    derepR <- derepFastq(filtRs[[sam]])
    ddR <- dada(derepR, err=errR, multithread=TRUE)
    merger <- mergePairs(ddF, derepF, ddR, derepR)
    mergers[[sam]] <- merger
}
rm(derepF); rm(derepR)

# Construct sequence table and remove chimeras
seqtab <- makeSequenceTable(mergers)
seqtab <- removeBimeraDenovo(seqtab, multithread=TRUE)
saveRDS(seqtab, "/Pathname/vaginal_bigdata/seqtab_vaginal_bigdata.rds")

######################
##    Assign Taxonomy    ##
```

```
#######################
taxa.minus <- assignTaxonomy(seqtab,"/Pathname/rdp_train_set_14.fa.gz")
taxa <- addSpecies(taxa.minus, "/Pathname/rdp_species_assignment_14.fa.gz",
allowMultiple=TRUE,verbose=TRUE)
colnames(taxa) <- c("Kingdom", "Phylum", "Class", "Order", "Family",
"Genus","Species")
unname(head(taxa))

####################
##      Phyloseq      ##
####################
library(phyloseq)
# Make a data.frame holding the sample data
samples.out <- rownames(seqtab)
samdf <- data.frame(Subject=samples.out)
rownames(samdf)<- samples.out

# Construct phyloseq object (straightforward from dada2 outputs)
ps <- phyloseq(otu_table(seqtab,taxa_are_rows=FALSE),
               sample_data(samdf),
               tax_table(taxa))
ps
save(ps,file="phyloseq_vaginal_bigdata_species.RData")
```

Script 4. Submission script for DADA2: qsub_R.sh

```
#!/bin/bash

    echo "#!/bin/bash" >> script.out
    echo "#$ -N R" >> script.out
    echo "#$ -cwd" >> script.out
    echo "#$ -j y" >> script.out
    echo "R CMD BATCH ./dada2_vaginal_bigdata.R" >> script.out
    chmod +x script.out
    qsub -q gene.q ./script.out
    rm -rf script.out
```

Script 5. Making a mapping file: link.sh

```
#!/bin/bash
input="/Pathname/link_vaginal.csv"
while IFS="," read -r f1 f2
do
  var=$(echo $f2 | awk -F"_L001" '{print $1}')
  echo "$f1,$var"
done < "$input" > link1_vaginal.csv
```

Script 6. Phyloseq analysis for DADA2

```
# Install R package Phyloseq
#source('http://bioconductor.org/biocLite.R')
#biocLite('phyloseq')

library('phyloseq');packageVersion("phyloseq")
library("data.table");packageVersion("data.table")
library("ggplot2");packageVersion("ggplot2")
library("gridExtra");packageVersion("gridExtra")

set.seed(100)

# Read in the data file
setwd("/Pathname/Results")

load("phyloseq_vaginal_bigdata_species.RData")
ps.vaginal <- ps
load("phyloseq_oral_bigdata_species.RData")
ps.oral <- ps
load("phyloseq_rectal_bigdata_species.RData")
ps.rectal <- ps

#otu.table.test=otu_table(ps)
#tax.table.test=tax_table(ps)
#sample.data.test=sample_data(ps)

# Add SampleType to Sample Data
sample_data(ps.vaginal)$SampleType <- replicate(nsamples(ps.vaginal),"Vaginal")
sample_data(ps.oral)$SampleType <- replicate(nsamples(ps.oral),"Oral")
sample_data(ps.rectal)$SampleType <- replicate(nsamples(ps.rectal),"Rectal")

# Merge Three Phyloseq Objects
ps <- merge_phyloseq(ps.vaginal,ps.oral,ps.rectal)

# Sequencing Depth
#pdf("DADA2_SeqDep.pdf",width=7,height=7)
#seqdep = data.table(as(sample_data(ps), "data.frame"),
#                TotalReads = sample_sums(ps), keep.rownames = TRUE)
#setnames(seqdep, "rn", "SampleID")
#pSeqDepth = ggplot(seqdep, aes(TotalReads)) + geom_histogram() + ggtitle("DADA2:
Sequencing Depth")
#pSeqDepth
#dev.off()
# Separating by SampleType
#pdf("DADA2_SeqDepbyST.pdf",width=8,height=7)
```

```
#pSeqDepth + facet_wrap(~SampleType)
#dev.off()

# Library Size
quantile(sample_sums(ps))
mean(sample_sums(ps))
quantile(sample_sums(ps.vaginal))
mean(sample_sums(ps.vaginal))
quantile(sample_sums(ps.oral))
mean(sample_sums(ps.oral))
quantile(sample_sums(ps.rectal))
mean(sample_sums(ps.rectal))

# Taxa Total Counts Histogram
#pdf("DADA2_TTC.pdf",width=7,height=7)
ttc = data.table(tax_table(ps),
                 TotalCounts = taxa_sums(ps),
                 OTU = taxa_names(ps))
#ggplot(ttc, aes(TotalCounts)) + geom_histogram() + ggtitle("DADA2: Histogram of
Total Counts")
#dev.off()

quantile(taxa_sums(ps))
mean(taxa_sums(ps))
quantile(taxa_sums(ps.vaginal))
mean(taxa_sums(ps.vaginal))
quantile(taxa_sums(ps.oral))
mean(taxa_sums(ps.oral))
quantile(taxa_sums(ps.rectal))
mean(taxa_sums(ps.rectal))

# How many 0 counts?
ttc[(TotalCounts <= 0), .N]
# How many singletons (OTUs that occur in just one sample, one time)?
ttc[(TotalCounts == 1), .N]
# How many doubletons?
ttc[(TotalCounts == 2), .N]

# Remove samples with 0 counts
ps <- prune_samples(sample_sums(ps)>0,ps)
# Original
ps0 <- ps
# Removing singletons
ps1 <- prune_taxa(taxa_sums(ps) > 1, ps)
# Removing singletons and doubletons
ps2 <- prune_taxa(taxa_sums(ps) > 2, ps)
```

```
# Visualize Alpha Diversity
pdf("DADA2_alpha0.pdf",width=8,height=7)
plot_richness(ps0, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("DADA2: Alpha diversity for original OTUs")
dev.off()
pdf("DADA2_alpha1.pdf",width=8,height=7)
plot_richness(ps1, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("DADA2: Alpha diversity after removing
singletons")
dev.off()
pdf("DADA2_alpha2.pdf",width=8,height=7)
plot_richness(ps2, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("DADA2: Alpha diversity after removing
singletons and doubletons")
dev.off()

# Bar Plots
#pdf("DADA2_barbyST.pdf",width=7,height=7)
#plot_bar(ps, x="SampleType", fill="Phylum")
#dev.off()

# MDS-Bray-Curtis Ordination
pdf("DADA2_MDS0.pdf",width=7,height=7)
ord <- ordinate(ps0, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps0, ord, color="SampleType"),
        plot_ordination(ps0, ord, axes=c(3,4), color="SampleType"),
        top="DADA2: MDS-Bray-Curtis ordination for original OTUs")
dev.off()
pdf("DADA2_MDS1.pdf",width=7,height=7)
ord <- ordinate(ps1, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps1, ord, color="SampleType"),
        plot_ordination(ps1, ord, axes=c(3,4), color="SampleType"),
        top="DADA2: MDS-Bray-Curtis ordination after removing singletons")
dev.off()
pdf("DADA2_MDS2.pdf",width=7,height=7)
ord <- ordinate(ps2, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps2, ord, color="SampleType"),
        plot_ordination(ps2, ord, axes=c(3,4), color="SampleType"),
        top="DADA2: MDS-Bray-Curtis ordination after removing singletons and
doubletons")
dev.off()
```

Script 8. Phyloseq analysis for QIIME

```
library('phyloseq');packageVersion("phyloseq")
library("data.table");packageVersion("data.table")
library("ggplot2");packageVersion("ggplot2")
library("gridExtra");packageVersion("gridExtra")

library("biom");packageVersion("biom")
library("qiimer");packageVersion("qiimer")

set.seed(100)
options(stringsAsFactors = FALSE)

setwd("/Pathname/Results")

# Load vaginal data
vaginal.biom <- read_biom("vaginal.biom")
vaginal.otus <- as.matrix(biom_data(vaginal.biom))
vaginal.otus <- t(vaginal.otus)
map_file <- "mapping_vaginal.txt"
sample <- import_qiime_sample_data(map_file)
sample$SampleType <- replicate(nrow(vaginal.otus),"Vaginal")
row.names(sample) <- sample$Subject
name <- row.names(vaginal.otus)
for (i in 1:length(name)){
  for (j in 1:nrow(sample)){
    if (name[i]==sample[[j,1]]){
      name[i]=sample[[j,2]]
    }
  }
}
row.names(vaginal.otus) <- name
taxa <- biom_taxonomy(vaginal.biom, attr = "taxonomy")
taxa <- t(data.frame(taxa))
rname <- row.names(taxa)
rname1 <- sapply(strsplit(rname, "X"), `[`, 2)
row.names(taxa) <- rname1
colnames(taxa) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus",
"Species")
ps.vaginal <- phyloseq(otu_table(vaginal.otus, taxa_are_rows=FALSE),
               sample_data(sample),
               tax_table(taxa))
# Load oral data
oral.biom <- read_biom("oral.biom")
oral.otus <- as.matrix(biom_data(oral.biom))
oral.otus <- t(oral.otus)
```

```
map_file <- "mapping_oral.txt"
sample <- import_qiime_sample_data(map_file)
sample$SampleType <- replicate(nrow(oral.otus),"Oral")
row.names(sample) <- sample$Subject
name <- row.names(oral.otus)
for (i in 1:length(name)){
  for (j in 1:nrow(sample)){
    if (name[i]==sample[[j,1]]){
      name[i]=sample[[j,2]]
    }
  }
}
row.names(oral.otus) <- name
taxa <- biom_taxonomy(oral.biom, attr = "taxonomy")
taxa <- t(data.frame(taxa))
rname <- row.names(taxa)
rname1 <- sapply(strsplit(rname, "X"), `[`, 2)
row.names(taxa) <- rname1
colnames(taxa) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus",
"Species")
ps.oral <- phyloseq(otu_table(oral.otus, taxa_are_rows=FALSE),
                    sample_data(sample),
                    tax_table(taxa))
# Load rectal data
rectal.biom <- read_biom("rectal.biom")
rectal.otus <- as.matrix(biom_data(rectal.biom))
rectal.otus <- t(rectal.otus)
map_file <- "mapping_rectal.txt"
sample <- import_qiime_sample_data(map_file)
sample$SampleType <- replicate(nrow(rectal.otus),"Rectal")
row.names(sample) <- sample$Subject
name <- row.names(rectal.otus)
for (i in 1:length(name)){
  for (j in 1:nrow(sample)){
    if (name[i]==sample[[j,1]]){
      name[i]=sample[[j,2]]
    }
  }
}
row.names(rectal.otus) <- name
taxa <- biom_taxonomy(rectal.biom, attr = "taxonomy")
taxa <- t(data.frame(taxa))
rname <- row.names(taxa)
rname1 <- sapply(strsplit(rname, "X"), `[`, 2)
row.names(taxa) <- rname1
```

```
colnames(taxa) <- c("Kingdom", "Phylum", "Class", "Order", "Family", "Genus",
"Species")
ps.rectal <- phyloseq(otu_table(rectal.otus, taxa_are_rows=FALSE),
                      sample_data(sample),
                      tax_table(taxa))

# Merge Three Phyloseq Objects
ps <- merge_phyloseq(ps.vaginal,ps.oral,ps.rectal)
save(ps,file="phyloseq_QIIME.RData")
load("phyloseq_QIIME.RData")

# Sequencing Depth
#pdf("DADA2_SeqDep.pdf",width=7,height=7)
#seqdep = data.table(as(sample_data(ps), "data.frame"),
#                    TotalReads = sample_sums(ps), keep.rownames = TRUE)
#setnames(seqdep, "rn", "SampleID")
#pSeqDepth = ggplot(seqdep, aes(TotalReads)) + geom_histogram() + ggtitle("DADA2:
Sequencing Depth")
#pSeqDepth
#dev.off()
# Separating by SampleType
#pdf("DADA2_SeqDepbyST.pdf",width=8,height=7)
#pSeqDepth + facet_wrap(~SampleType)
#dev.off()

# Library Size
quantile(sample_sums(ps))
mean(sample_sums(ps))
quantile(sample_sums(ps.vaginal))
mean(sample_sums(ps.vaginal))
quantile(sample_sums(ps.oral))
mean(sample_sums(ps.oral))
quantile(sample_sums(ps.rectal))
mean(sample_sums(ps.rectal))

# Taxa Total Counts Histogram
#pdf("DADA2_TTC.pdf",width=7,height=7)
ttc = data.table(tax_table(ps),
                 TotalCounts = taxa_sums(ps),
                 OTU = taxa_names(ps))
#ggplot(ttc, aes(TotalCounts)) + geom_histogram() + ggtitle("DADA2: Histogram of
Total Counts")
#dev.off()

quantile(taxa_sums(ps))
mean(taxa_sums(ps))
```

```
quantile(taxa_sums(ps.vaginal))
mean(taxa_sums(ps.vaginal))
quantile(taxa_sums(ps.oral))
mean(taxa_sums(ps.oral))
quantile(taxa_sums(ps.rectal))
mean(taxa_sums(ps.rectal))

# How many 0 counts?
ttc[(TotalCounts <= 0), .N]
# How many singletons (OTUs that occur in just one sample, one time)?
ttc[(TotalCounts == 1), .N]
# How many doubletons?
ttc[(TotalCounts == 2), .N]

# Remove samples with 0 counts
ps <- prune_samples(sample_sums(ps)>0,ps)
# Original
ps0 <- ps
# Removing singletons
ps1 <- prune_taxa(taxa_sums(ps) > 1, ps)
# Removing singletons and doubletons
ps2 <- prune_taxa(taxa_sums(ps) > 2, ps)

# Visualize Alpha Diversity
pdf("QIIME_alpha0.pdf",width=8,height=7)
plot_richness(ps0, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("QIIME: Alpha diversity for original OTUs")
dev.off()
pdf("QIIME_alpha1.pdf",width=8,height=7)
plot_richness(ps1, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("QIIME: Alpha diversity after removing
singletons")
dev.off()
pdf("QIIME_alpha2.pdf",width=8,height=7)
plot_richness(ps2, x="SampleType", measures=c("Observed", "Chao1",
"Shannon"),color="SampleType")+ggtitle("QIIME: Alpha diversity after removing
singletons and doubletons")
dev.off()

# Bar Plots
#pdf("DADA2_barbyST.pdf",width=7,height=7)
#plot_bar(ps, x="SampleType", fill="Phylum")
#dev.off()

# Ordination
pdf("QIIME_MDS0.pdf",width=7,height=7)
```

```
ord <- ordinate(ps0, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps0, ord, color="SampleType"),
        plot_ordination(ps0, ord, axes=c(3,4), color="SampleType"),
        top="QIIME: MDS-Bray-Curtis ordination for original OTUs")
dev.off()
pdf("QIIME_MDS1.pdf",width=7,height=7)
ord <- ordinate(ps1, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps1, ord, color="SampleType"),
        plot_ordination(ps1, ord, axes=c(3,4), color="SampleType"),
        top="QIIME: MDS-Bray-Curtis ordination after removing singletons")
dev.off()
pdf("QIIME_MDS2.pdf",width=7,height=7)
ord <- ordinate(ps2, method="MDS", distance="bray")
grid.arrange(plot_ordination(ps2, ord, color="SampleType"),
        plot_ordination(ps2, ord, axes=c(3,4), color="SampleType"),
        top="QIIME: MDS-Bray-Curtis ordination after removing singletons and
doubletons")
dev.off()
```