Research and Testing Laboratory

Data Analysis Methodology

Last Updated: 11/07/2014

The most recent version of this file can be downloaded from
http://www.researchandtesting.com/docs/Data_Analysis_Methodology.pdf

Contents

Version Changelog

Current Version:

Version 2.2.4 (11/07/2014)

- Updated the file descriptions to include the file type and recommended program for viewing the data.
- Updated the file descriptions to include all files provided by RTL.
- Added new section to cover recommended programs for viewing provided data.

Previous Versions:

Version 2.2.3 (09/16/2014)

- Updated the data archive to split files too large to fit on our webserver.
- Included instructions for how to handle split zip archives.

Version 2.2.2 (09/03/2014)

- Added the OTUs folder to the Analysis archive.
- Moved OTUmap.txt from Analysis/OTUMap.txt to Analysis/OTUs/OTUMap.txt
- Added OTUs.fas to the Analysis/OTUs archive.
- Corrected the otus.tre file. All ';' within sequence definitions have been changed to '_'.

Version 2.2.1 (08/29/2014)

- Added the customer data retention policy.

Version 2.2.0 (07/09/2014)

- Added phylogenetic tree construction using MUSCLE and FastTree.
- Added Krona visualization to the Taxonomic Analysis pipeline.

- Added phylogenetic tree, multiple sequence alignment, and Krona visualizations to the analysis zip archive.
- Updated 454 and Ion Torrent PGM processing to run using the same workflow as MiSeq.
- Added description for the OTUMap.txt file in the Analysis Folder.

Version 2.1.1 (05/20/2014)

- Updated OTU Selection. Trimming to shortest sequence now performed before UPARSE OTU Selection.

Version 2.1.0 (02/28/2014)

- Updated denoiser to use PEAR for paired-end read merging in place of USEARCH.

Version 2.0.0 (01/28/2014)

- Updated denoiser to use USEARCH 7, replacing USEARCH 5.
- Methodology now accounts for processing of 454, Ion Torrent PGM and Illumina MiSeq data.

Term Definitions

Terms used within this guide are defined as follows:

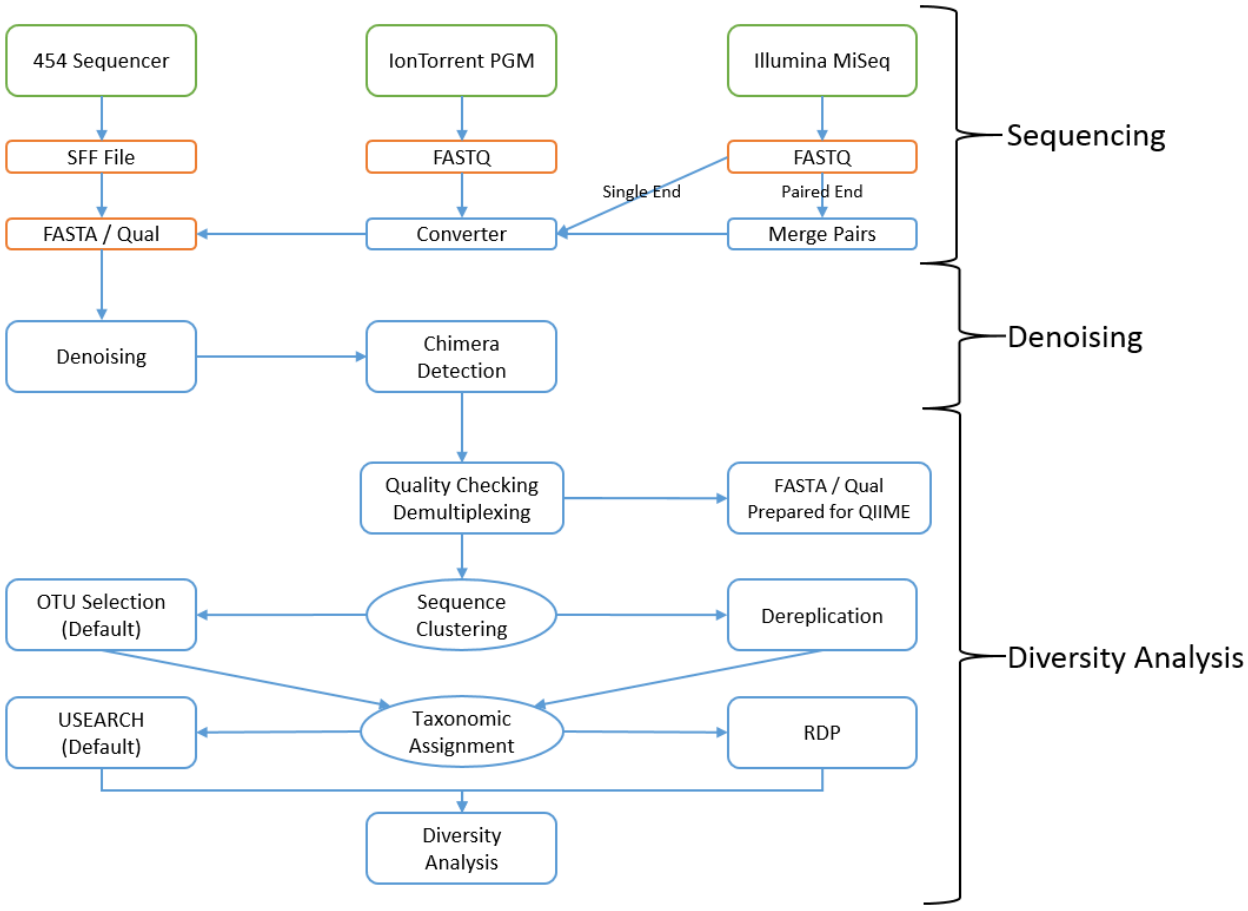
- Tag
  - The term tag refers to the 8-10 bp sequence at the 5' end of the sequence read.
  - The tag is also known as the barcode in some programs.
- ASCII value
  - ASCII (American Standard Code for Information Interchange) is a character encoding scheme based on the English alphabet to encode the following: the numbers 0-9, the letters a-z, the letters A-Z, basic punctuation, control codes (such as new line), and the blank space.
  - Each letter, number and punctuation mark on a keyboard is assigned a numeric value (mostly between 0 and 127) using the ASCII table in order to create a way of encoding/decoding character symbols into computer readable digital bit patterns.

Client Data Retention Policy

Data will made available for download (typically via a 12 month temporary link) upon completion of your project. RTL will make every reasonable effort to store all electronic data for your project for a period of 24 months from the date of notification that the project has been completed. If you have any questions regarding your data or if you need to discuss longer term storage, please contact us.

Data Analysis Methodology

Visual Overview of the Data Analysis Process



Overview of the Data Analysis Process

Once sequencing of your data has completed, the data analysis pipeline will begin processing the data. The data analysis pipeline consists of two major stages, the denoising and chimera detection stage and the microbial diversity analysis stage. During the denoising and chimera detection stage, denoising is performed using various techniques to remove short sequences, singleton sequences, and noisy reads. With the bad reads removed, chimera detection is performed to aid in the removal of chimeric

sequences. Lastly, remaining sequences are then corrected base by base to help remove noise from within each sequence. During the diversity analysis stage, each sample is run through our analysis pipeline to determine the taxonomic information for each constituent read and then this information is collected for each sample. This stage is performed for all customers whose data is sequenced using primers targeting the 16S, 18S, 23S, ITS or SSU regions. Analysis can performed on other regions but may require additional charges.

The data analysis pipeline is broken down into the following steps, each of which is discussed more thoroughly in the sections below:

- Denoising and Chimera Checking
    1. Denoising
    2. Chimera Checking
    3. SFF File Generation (454 only) – FASTQ File Generation (Ion Torrent & Illumina)

- Microbial Diversity Analysis
    1. Quality Checking and FASTA Formatted Sequence/Quality File Generation
    2. Sequence Clustering
    3. Taxonomic Identification
    4. Data Analysis

Denoising and Chimera Checking

Denoising

The process of denoising is used to correct errors in reads from next-generation sequencing technologies. According to the paper “Accuracy and quality of massively parallel DNA pyrosequencing” by Susan Huse, et al. and “Removing noise from pyrosequenced amplicons” by Christopher Quince, et al. the per base error rates from 454 pyrosequencing attain an accuracy rate of 99.5% [1] [2]. The paper “A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers” by Michael Quail, et al. states that the observed error rates generated by the Illumina MiSeq is less than .4% while the Ion Torrent PGM has an error rate of 1.78% [3]. Due to the large number of reads and even higher number of base calls per sequencing run, the total number of noisy reads can be quite substantial. In order to determine true diversity it becomes critical to determine which reads are good and which reads contain noise introduced by the experimental

procedure. The Research and Testing Laboratory analysis pipeline attempts to correct this issue by denoising entire regions of data prior to performing any other steps of the pipeline.

The Research and Testing Laboratory analysis pipeline performs denoising by performing the following steps on each region:

1. The forward and reverse reads are taken in FASTQ format and are merged together using the PEAR Illumina paired-end read merger [4]. (Illumina MiSeq Paired End Sequencing Only)
2. The FASTQ (Illumina MiSeq and Ion Torrent PGM Only) and SFF (454 Only) formatted files are converted into FASTA formatted sequence and quality files.
3. Reads are run through an internally developed quality trimming algorithm. During this stage each read has a running average taken across the sequence and is trimmed back at the last base where the total average is greater than 25.
4. Sequence reads are then sorted by length from longest to shortest.
5. Prefix dereplication is performed using the USEARCH [5] algorithm. Prefix dereplication groups reads into clusters such that each sequence of equal or shorter length to the centroid sequence must be a 100% match to the centroid sequence for the length of the sequence. Each cluster is marked with the total number of member sequences. Sequences < 100bp in length are not written to the output file, however no minimum cluster size restriction is applied which will allow singleton clusters to exist in the output.
6. Clustering at a 4% divergence (454 & Illumina) or 6% divergence (IonTorrent) is performed using the USEARCH [5] clustering algorithm. The result of this stage is the consensus sequence from each new cluster, with each tagged to show their total number of member sequences (dereplicated + clustered). Clusters that contain <2 members (singleton clusters) are not added to the output file, thus removing them from the data set.
7. OTU Selection is performed using the UPARSE OTU selection algorithm [6] to classify the large number of clusters into OTUs.
8. Chimera checking, which is explained in more detail below in the section entitled “Chimera Checking”, is performed on the selected OTUs using the UCHIME chimera detection software executed in *de novo* mode [7].
9. Each clustered centroid from step 6 listed above is then mapped to their corresponding OTUs and then marked as either Chimeric or Non-Chimeric. All Chimeric sequences are then removed.
10. Each read from step 3 is then mapped to their corresponding nonchimeric cluster using the USEARCH global alignment algorithm [5].

11. Using the consensus sequence for each centroid as a guide, each sequence in a cluster is then aligned to the consensus sequence and each base is then corrected using the following rules where C is the consensus sequence and S if the aligned sequence:
    a. If the current base pair in S is marked to be deleted, then the base is removed from the sequence if the quality score for that base is less than 30.
    b. If the current position in S is marked to have a base from C inserted, then the base is inserted into the sequence if the mean quality score from all sequences that mark the base as existing is greater than 30.
    c. If the current position in S is marked as a match to C but the bases are different, then the base in S is changed if the quality score for that base is less than 30.
    d. If a base was inserted or changed, the quality score for that position is updated. If the base was deleted the quality score for that position is removed.
    e. Otherwise, leave the base in S alone and move to the next position.
12. The corrected sequences are then written to the output file.

Chimera Checking

As discussed in the paper "Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons" by Brian Haas, et al. the formation of chimeric sequences occurs when an aborted sequence extension is misidentified as a primer and is extended upon incorrectly in subsequent PCR cycles [8]. This can be seen in Figure 1, shown below.

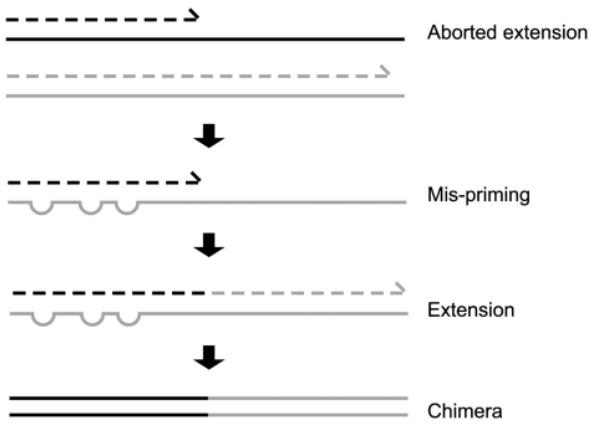


Figure 1.

Formation of chimeric sequences during PCR. An aborted extension product from an earlier cycle of PCR can function as a primer in a subsequent PCR cycle. If this aborted extension product anneals to and primes DNA synthesis from an improper template, a chimeric molecule is formed. Figure and description taken directly from "Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons" by Brian Haas, et al. **[8]**.

Because amplification produces chimeric sequences that stem from the combination of two or more original sequences [7], we will perform chimera detection using the *de novo* method built into UCHIME.

The Research and Testing Laboratory analysis pipeline performs chimera detection and removal by executing UCHIME [7] in *de novo* mode on the clustered data that was output by our denoising methods. By using this method we can determine chimeras across entire region of data even after accounting for noise and removing low quality sequences.

Microbial Diversity Analysis

In order to determine the identity of each remaining sequence, the sequences must first be quality checked and demultiplexed using the denoised data generated previously. These sequences are then clustered into OTUs using the UPARSE [6] algorithm. The centroid sequence from each cluster is then run against either the USEARCH global alignment algorithm or the RDP Classifier against a database of high quality sequences derived from the NCBI database. The output is then analyzed using an internally developed python program that assigns taxonomic information to each sequence and then computes and writes the final analysis files.

Quality Checking and FASTA Formatted Sequence/Quality File Generation

The denoised and chimera checked reads generated during sequencing are condensed into a single FASTA formatted file such that each read contains a one line descriptor and one to many lines of sequence/quality scores. The Research and Testing Laboratory analysis pipeline takes the FASTA formatted sequence and quality files and removes any sequence which fails to meet the following quality control requirements:

1. Sequences must be at least ½ the expected length given the primer sets used.
2. Sequences must contain a valid error free barcode.

Sequences that pass the quality control screening are condensed into a single FASTA formatted sequence and quality file such that each read has a one line descriptor followed by a single line of sequence/quality data. The descriptor line in both files has been altered to contain the samples name followed by the original descriptor line, separated with a unique delimiter (::).

This stage of the pipeline creates the FASTA reads archive which contains the following files:

1. The sequence reads from all samples concatenated into a single sequence file. The original tags have been removed from each sequence and an "artificial tag" has been added in its place. The title of the file will be <name>_<order ID>.fas.
2. The quality scores from all samples concatenated into a single quality file. The scores are labeled with the corresponding sample name and will have a matching line in the .fas file. Since the original tags were removed from the sequence and an "artificial tag" was put into its place, the quality scores have been similarly altered such that the original scores for the tag have been removed and an "artificial quality tag" has been added in its place. The artificial quality tag consists of Q30s for the length of the tag. This file will be labeled <name>_<order ID>.qual.
3. A mapping file consisting of sample names included in the analysis. This file contains the information for each sample such that each line has the sample name, tag and primer used for the sample. This file will be labeled as: <name>_<order ID>.txt

Sequence Clustering

OTU selection clusters sequences into clusters using either an OTU selection program or dereplication depending on the needs of the customer. By default, the OTU selection method is used to determine OTUs and uses the centroid sequence for each OTU to determine taxonomic information. However, if the customer requests that we use the dereplication method, then the clusters will instead represent 100% identity clusters and taxonomic information will be assigned to each of these cluster's centroid sequence instead.

OTU Selection (Default)

OTU selection is performed using the guidelines discussed in the paper "UPARSE: Highly accurate OTU sequences from microbial amplicon reads" by Robert Edgar [6]. In that paper, the following methodology is laid out in order to select OTUs:

1. Perform dereplication on the sequences.
2. Remove all singleton clusters from the data set and sort the data by abundance.
3. Trim all sequences to the same length.
4. Perform OTU clustering using UPARSE.
5. Map original reads to the OTUs

Dereplication of sequences is performed using the USEARCH prefix dereplication method [5]. Once complete we removed all singleton clusters and sorted the remaining sequences by cluster size from largest to smallest. The sequences are then run through a trimming algorithm that trims each sequence down to the same size. It should be noted that the sequences are only trimmed for UPARSE and the final taxonomic analysis is based upon the full length sequences. Next we use the UPARSE algorithm to select OTUs [6]. Using the USEARCH global alignment algorithm [5] we then assign each of the original reads back to their OTUs and write the mapping data to an OTU map and OTU table file.

Dereplication

Some customers would prefer to not have their data go through UPARSE if they are interested in the taxonomic information of the singleton sequences. For these customers we have the pipeline replace the OTU selection stage with a dereplication step using the USEARCH prefix dereplication algorithm [5]. Once the dereplication is complete a dereplication mapping table and dereplication table (same format as the OTU table) are both created and the centroid sequences are written to a file for taxonomic assignment.

Tree Building

Once OTU selection has been performed, a phylogenetic tree in Newick format is constructed. In order to construct the phylogenetic tree, multiple sequence alignment must be done on the OTU sequences in order to generate equal length aligned sequences. The multiple sequence aligner MUSCLE [9] [10], developed by Robert Edgar, is used with a maximum of 2 iterations in order to perform the alignment of the OTU data. The finished multiple sequence alignment is then passed into FastTree [11] [12], developed by Morgan Price at the Lawrence Berkeley National Lab, a program used to infer approximately-maximum-likelihood phylogenetic trees from aligned sequence data. If you would like to learn more about how FastTree works, please visit the following link: http://www.microbesonline.org/fasttree/#How.

Taxonomic Identification

In order to determine the taxonomic information for each remaining sequence, the sequences must be run through either the USEARCH global alignment program or the RDP classifier. By default the

USEARCH based method is employed however the RDP classifier can be substituted if a customer has requested that we use the RDP classifier instead. In either case the data is identified using a database of high quality sequences derived from NCBI that is maintained in house. If a customer would prefer we classify their data using a different database such as GreenGenes then we can substitute that database in place of our own. If a non-standard database is requested that requires Research and Testing Laboratory to spend time converting or creating, then a small fee may be charged.

USEARCH Global Search (Default)

The global search method uses a mixture of the USEARCH global search algorithm along with a python program to then determine the actual taxonomic assignment that is assigned to each read. This method is described in the paper "An extensible framework for optimizing classification enhances short-amplicon taxonomic assignments" by Nicholas Bokulich, et al. [13]. The paper describes a methodology in which a high quality database is used in pair with USEARCH rapidly find the top 6 matches in the database for a given sequence. From these 6 sequences you then assign a confidence value to each taxonomic level (kingdom, phylum, class, order, family, genus and species) by taking the number of taxonomic matches that agree with the top match and then divide by the number of total matches, e.g. Bacteria is the top kingdom match and 5 hits state Bacteria and 1 hit shows another kingdom, this would assign a confidence of 5/6 = .83. Once confidence values are assigned for each sequence an RDP formatted output file is generated to be used by our final analysis program.

RDP Classifier

The RDP Classifier is naïve Bayesian classifier than can rapidly determine taxonomic information for sequences while automatically determining the confidence it has at each taxonomic level [14]. The RDP classifier is run against an internally maintained database or a customer requested database along with a taxonomic file to help determine confidence values by giving the classifier a taxonomic tree.

Diversity Analysis

Regardless of the classifier that was used, the data next enters the diversity analysis program. This program takes the OTU/Derep table output from sequence clustering along with the output generated during taxonomic identification and begins the process of generating a new OTU table with the

taxonomic information tied to each cluster. This updated OTU table is then written to the output analysis folder with both the trimmed and full taxonomic information for each cluster. For each taxonomic level (kingdom, phylum, class, order, family, genus and species) four files are generated which contain the number of sequences per full taxonomic match per sample, the percentage per full taxonomic match per sample, the number of sequences per trimmed taxonomic match per sample and the percentage per trimmed taxonomic match per sample. These files are all described in more detail below.

File Descriptions and Formatting

Zip Archives

The following archives will be passed along to you upon completion of your order:

- <Name>_<OrderNumber>Raw<SequencingDate>.zip
  - This archive contains the raw FASTQ and/or raw SFF as described in the "Raw Sequence Data File Formats" section found on page 17.
- <Name>_<OrderNumber>Fasta<SequencingDate>.zip
  - This archive contains the denoised sequence data for your entire order in FASTA/Qual format as described in the "FASTA Archive File Descriptions" section found on page 20.
- <Name>_<OrderNumber>Analysis<SequencingDate>.zip
  - This archive contains the analysis data described in the "Analysis Archive File Descriptions" section found on page 21.
  - This archive will only be sent if you used a standard primer set that we have a working database for. Custom assays will likely not be analyzed.

Split Zip Archives

If any zip archive is larger than 10GB in size, we will be unable to upload the file to our file server without breaking the file into smaller chunks. In order for you to open these files you will need to download each file in the archives set (denoted with <ArchiveName>.zip.XXX where XXX is a number starting at 001 and counting upwards) and then stitch them back together prior to unzipping the archive. The following commands can be used to rebuild the zip file prior to unzipping.

Windows

Stitching the files together in Windows requires you to do the following:

1. Open a command/DOS prompt
   - In most versions of windows go to "Start menu" then type in cmd and run cmd.exe.
2. Navigate to the folder you downloaded the files into.
3. Type in the following: copy /B ArchiveName.zip.* ArchiveName.zip
4. Unzip the ArchiveName.zip file as you normally would.

Linux / Mac

Stitching the files together in Linux or Mac requires you to do the following:

1. Open a command terminal.
2. Navigate to the folder you downloaded the files into.
3. Type in the following: cat ArchiveName.zip.* > ArchiveName.zip
4. Unzip the ArchiveName.zip file as you normally would.

Raw Sequence Data File Formats

SFF File Generation

An sff file is a binary file containing detailed information regarding each read in a single file. For each read, the sff contains a flowgram, quality score and sequence with defined lengths from QC measures performed by the machine. The sff represents the raw data and includes many reads that may have been excluded due to length or chimera detection or any other filter requested for custom processing. Since the files are binary, they cannot be opened with standard text editors. Special programs like Mothur [15] or BioPython [16] are able to load their data into human readable formats and output fasta, qual, flowgram or text (sff.txt) versions. Sff files or their derivatives can then be used for further processing of the data. Sff files provided may be of two forms. In the case of an entire region containing a single investigator's samples, the entire region plus mapping file is provided. In cases where multiple investigators had samples on a single region, each sample is demultiplexed from the sff file using the Roche sffinfo tool by providing its barcode, effectively eliminating it from any read extracted. The split sff can then be used for raw data or submitted directly to archives like the NCBI's SRA. In cases where a single sff for all samples is desired but an entire quadrant is not used, an investigator may request a single sff for a nominal charge. Alternatively, it is possible to use the provided split sff files for denoising/chimera removal by modifying the mapping files. Additional instructions are available if you wish to do so.

FASTQ File Generation

FASTQ files are text based formatted data files that store the nucleotide sequences generated by the sequencer and their corresponding quality scores encoded as ASCII characters. A FASTQ file contains 4 lines per read that contain the following information:

- Line 1 contains the sequence ID (read definition) and is prepended with an “at” symbol, ‘@’.
- Line 2 contains the sequence data.
- Line 3 acts as a separator line between the sequence data and the quality score, it contains a single plus sign, ‘+’.
- Line 4 encodes the quality values for the sequence in line 2 with each quality score being represented by a single character. As such Line 2 and Line 4 must be the same length.

Decoding of the quality scores requires you to know the phred score offset that was used when the file was generated. Once you know the offset, you can take the ASCII value for the given character and subtract the offset value to obtain the quality score. For example, if the phred offset is +33 and the character ‘B’ is encountered, then the quality score for that position would be 33 as ‘B’ is represented by the ASCII value 66 and the offset is 33 (66 – 33 = 33). Using the same logic, ‘A’ (represented by the value 65) would be 65-33=32 meaning the ‘A’ character represents a quality score of 32. A free to view ASCII table can be found here: http://www.ascii-code.com/.

Roche 454

The Roche 454 sequencers produce single SFF files for each region of a run, where runs here at RTL are broken into either 2 or 4 regions per run. Customers who opt to purchase an entire region or run of sequencing on the 454 will receive these files. Customers who opt to pay per sample will be given a single SFF file for each of their samples. SFF files that are generated one per sample will have their barcodes trimmed back by the SFF file generator. Some programs capable of reading SFF files will be able to see the original barcode and other programs will continue to ignore it. Please see the documentation for the program you are using to determine which method their software uses.

IonTorrent PGM

The IonTorrent PGM produces numerous file formats for their sequence data. We prefer to stick with the machine producing a single SFF file for the entire run that we then break down into one SFF file per sample, similar to how we run Roche 454 data. For information on what to expect when dealing with your SFF files, please see the section titled “Roche 454”, found on page 18.

Illumina MiSeq

The Illumina MiSeq produces FASTQ files with a phred offset of +33. While the FASTQ file(s) generated by a MiSeq do contain all of the raw sequence data generated by the sequencer, they **do not** contain any information regarding the primer (forward or reverse). Unlike other next generation sequencing technologies, the MiSeq does not sequence the primer, instead it begins sequencing at the first base pair following the forward or reverse primer. This can make processing of your data difficult if the post processing program you decide to use requires it be able to see the primer on the sequence, however most modern programs have removed this restriction due to the prevalence of Illumina data. FASTQ files generated by the Illumina MiSeq come in two forms depending on the sequencing – either paired end or single end. Single end reads are stored in a single FASTQ file with each read in the file representing a read from the sequencer. Paired end reads, however, are slightly more complex and are covered in following section. Reads from the Illumina MiSeq are stored per sequence and are demultiplexed by the Illumina Software, thus your raw data will be missing all barcode information.

Paired End FASTQ Files

Paired end reads are stored in two FASTQ files with the first file storing the forward "half" of the read and the second file which stores the reverse "half" of the read. It should be noted that both reads are provided in forward order, meaning if you wish to link the two reads together you will first need to take the reverse complement of the reads in the second file. Depending on the insertion size and sequencing read length, the forward and reverse reads may or may not overlap at some point. Unlike other FASTQ files, the order of reads within these files must be kept in a specific order to avoid issues with most post processing programs. The reads within these two files must stay the same between the two files, meaning if you remove or move a sequence in one file, you must remove it or move it to same place in the other file to preserve the same order in both files.

Because the insert size for a paired end sequence matters, we provide two examples of how your sequences may, or may not, line up. In both examples it is assumed that you have already taken the reverse complement of the reverse reads.

Example 1 – Insert Size approx. 500bp using a 2x300 kit.

Forward----------

~250 BP Forward Only | ~50 BP Alignment | ~250 BP Reverse Only

----------Reverse

Example 2 – Insert Size approx. 800bp using a 2x300 kit.

Forward----------

300 BP Forward Only | 0 BP Alignment | 300 BP Reverse Only

----------Reverse

FASTA Archive File Descriptions

The FASTA archive you receive with your data will contain the following files:

- <Name>_<OrderNumber>.fna
  - This file contains your sequence data after it has undergone quality trimming, denoising and quality checking. Each sequence in this file has had an 8 nucleotide barcode prepended to the front of it. This barcode is not the barcode that was originally used during sequencing, instead we use a faux barcode for each sample to ensure that they each have a unique barcode.
  - **File Type:** FASTA formatted sequence data. Sequence definition lines begin with a '>' (greater than) character followed by 1 or more lines of sequence data.
  - **Recommended Program:** Any text editor.
- <Name>_<OrderNumber>.qual
  - This file contains the quality scores for your sequence data after it has undergone quality trimming, denoising and quality checking. Each quality score set in this file has had 8 fake 40 scores prepended to the front of it to account for the faux barcode added to the sequence file.
  - **File Type:** FASTA formatted sequence data. Sequence definition lines begin with a '>' (greater than) character followed by 1 or more lines of sequence quality data.
  - **Recommended Program:** Any text editor.
- <Name>_<OrderNumber>.txt
  - This file stores a table that maps each faux barcode generated in the sequence file to the sample it corresponds to. The table contains four columns and has been formatted to be compatible with QIIME, however some changes may be required if your sample names contain characters or symbols that are disallowed in QIIME.
  - The table contains the following four columns:
    - Column 1 contains the sample name (SampleID) for your sample.
    - Column 2 contains the faux barcode sequence added to you sequence file.
    - Column 3 contains the primer sequence that was used for that sample.

        - Column 4 contains the description of the sample which is required for QIIME. Because we do not know exactly what your samples are, we instead just fill this column with the sample name again.
    - **File Type:** Table written in tab-separated value (TSV) format.
    - **Recommended Program:** Any spreadsheet program or text editor.

Analysis Archive File Descriptions

The analysis archive you receive with your data will contain the following files:

- For each taxonomic level (<level>) where level is Kingdom, Phylum, Class, Order, Genus or Species.
    - FullTaxa.<level>.counts.txt
        - This file contains a table with the columns representing each sample in your order and the rows representing each unique taxonomic information for the top hit listed down to <level>, e.g. if <level> is Phylum then it will give each unique Kingdom/Phylum combination.
        - Each row/column intersection defines the number of sequences in the sample that matched that particular unique taxonomic information.
        - Keep in mind that the Full Taxa data shows only the taxonomic information for the top hit, regardless of what the confidence values were.
        - **File Type:** Table written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - FullTaxa.<level>.percent.txt
        - This file contains a table with the columns representing each sample in your order and the rows representing each unique taxonomic information for the top hit listed down to <level>, e.g. if <level> is Phylum then it will give each unique Kingdom/Phylum combination.
        - Each row/column intersection defines the percent of sequences in the sample that matched that particular unique taxonomic information.
        - Keep in mind that the Full Taxa data shows only the taxonomic information for the top hit, regardless of what the confidence values were.
        - **File Type:** Table written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - TrimmedTaxa.<level>.counts.txt

        - This file contains a table with the columns representing each sample in your order and the rows representing each unique taxonomic information for the top hit listed down to <level>, e.g. if <level> is Phylum then it will give each unique Kingdom/Phylum combination.
        - Each row/column intersection defines the number of sequences in the sample that matched that particular unique taxonomic information.
        - Keep in mind that the Trimmed Taxa data shows the taxonomic information after the confidence values are taken into account. The USEARCH method rejects the taxonomic information at a level if the confidence is below 51% while the RDPClassifier uses a minimum confidence of 80%.
        - **File Type:** Table written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - TrimmedTaxa.<level>.percent.txt
        - This file contains a table with the columns representing each sample in your order and the rows representing each unique taxonomic information for the top hit listed down to <level>, e.g. if <level> is Phylum then it will give each unique Kingdom/Phylum combination.
        - Each row/column intersection defines the percent of sequences in the sample that matched that particular unique taxonomic information.
        - Keep in mind that the Trimmed Taxa data shows the taxonomic information after the confidence values are taken into account. The USEARCH method rejects the taxonomic information at a level if the confidence is below 51% while the RDPClassifier uses a minimum confidence of 80%.
        - **File Type:** Table written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
- OTU/Derep tables
    - FullTaxa.otu_table.txt
        - This file contains a table with the columns representing each sample in your order and the rows representing each unique OTU or Dereplication Cluster. The final column contains the taxonomic information for that particular OTU/Cluster listed down to the Species level.
        - Keep in mind that the Full Taxa data shows only the taxonomic information for the top hit, regardless of what the confidence values were.
        - **File Type:** Table written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - FullTaxa.otu_table.biom

        - This file contains a BIOM formatted copy of the OTU Table stored in FullTaxa.otu_table.txt.
        - Keep in mind that the Full Taxa data shows only the taxonomic information for the top hit, regardless of what the confidence values were.
        - **File Type:** OTU Table in BIOM Format.
        - **Recommended Program:** Any text editor or program that accepts BIOM files (e.g. QIIME).
    - TrimmedTaxa.otu_table.txt
        - This file contains a table with the columns representing each sample in your order and the rows representing each unique OTU or Dereplication Cluster. The final column contains the taxonomic information for that particular OTU/Cluster listed down to the Species level.
        - Keep in mind that the Trimmed Taxa data shows the taxonomic information after the confidence values are taken into account. The USEARCH method rejects the taxonomic information at a level if the confidence is below 51% while the RDPClassifier uses a minimum confidence of 80%.
        - **File Type:** Tables written in tab-separated value (TSV) format.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - TrimmedTaxa.otu_table.biom
        - This file contains a BIOM formatted copy of the OTU Table stored in TrimmedTaxa.otu_table.txt.
        - Keep in mind that the Trimmed Taxa data shows the taxonomic information after the confidence values are taken into account. The USEARCH method rejects the taxonomic information at a level if the confidence is below 51% while the RDPClassifier uses a minimum confidence of 80%.
        - **File Type:** OTU table in BIOM format.
        - **Recommended Program:** Any Text Editor or program that accepts BIOM files (e.g. QIIME).
- **Krona Folder**
    - Raw Data Folder
        - This folder contains the raw data files that were passed to Krona in order to generate the FullTaxa and TrimmedTaxa Krona HTML files. These files were derived directly from the FullTaxa.species.counts.txt and TrimmedTaxa.species.counts.txt files. These files are provided for transparency purposes regarding how your visualization data was created.
    - FullTaxa.krona.html

        - This file contains the Krona visualization of the FullTaxa.species.count.txt file. The visualization file is a standard HTML file and should be accessible using any web browser. This visualization was generated using the FullTaxa.species.counts.txt file described before and contains data on all samples found in that file. You are able to switch between samples using the menu on the left hand side of the screen. Keep in mind that the "Collapse" checkbox is checked by default which can cause your taxonomic levels to look incorrect.
        - **This file requires an internet connection in order to work.**
        - **File Type:** HTML file with embedded code from Krona.
        - **Recommended Program:** Any Web Browser, however the developers suggest using FireFox.
    - TrimmedTaxa.krona.html
        - This file contains the Krona visualization of the FullTaxa.species.count.txt. The visualization file is a standard HTML file and should be accessible using any web browser. This visualization was generated using the TrimmedTaxa.species.counts.txt file described before and contains data on all samples found in that file. You are able to switch between samples using the menu on the left hand side of the screen. Keep in mind that the "Collapse" checkbox is checked by default which can cause your taxonomic levels to look incorrect.
        - **This file requires an internet connection in order to work.**
        - **File Type:** HTML file with embedded code from Krona.
        - **Recommended Program:** Any Web Browser, however the developers suggest using FireFox.
- **OTUs Folder**
    - OtuMap.txt
        - This file contains the mapping of each OTU identification number used in the OTU Table files. Each line contains the following information separated by tabs: the OTU identification number, the number of sequences that make up the OTU, the sequence definition for the seed sequence and then the sequence definition for each member sequence.
        - **File Type:** Table written in tab-separated value (TSV) format. The first column is the OTU #, the 2nd can be ignored, the 3rd is the OTU centroid sequence definition and columns 4+ contain each member sequence definition.
        - **Recommended Program:** Any spreadsheet program or text editor.
    - OTUs.fas

- This file contains the OTU sequences selected during sequence clustering in fasta format. For information regarding how this file was generated please see Sequence Clustering on page 12.
- **File Type:** FASTA formatted sequence data. Sequence definition lines begin with a '>' (greater than) character followed by 1 or more lines of sequence data.
- **Recommended Program:** Any text editor.

- **TreeData Folder**
  - otu_map.condensed.txt
    - This file contains a condensed version of the OtuMap.txt file. Each line contains two columns separated by tabs. The first column gives the OTU identification number and the second column contains the sequence definition for the seed sequence. Thus this file contains the equivalent of columns #1 and #3 from the OTU Map.
    - **File Type:** Table written in tab-separated value (TSV) format
    - **Recommended Program:** Any Spreadsheet program or Text Editor
  - otus.msa
    - This file contains the multiple sequence alignment for each OTU described in the OTU table and OTU map. This file was generated using MUSCLE as described above in the section titled Tree Building.
    - **File Type:** FASTA formatted sequence data. Sequence definition lines begin with a '>' (greater than) character followed 1 or more lines of sequence data.
    - **Recommended Program:** This file is best viewed using a MSA viewer but can also be viewed using any text editor.
  - otus.tre
    - This file contains the phylogenetic tree in Newick tree format created using the otus.msa file. This file was generated using FastTree as described in the section titled Tree Building on page 13.
    - **File Type:** Newick tree formatted phylogenetic tree.
    - **Recommended Program:** This file is best viewed using a phylogenetic tree viewer but can also be viewed using any text editor.
    - **Disclaimer:** Numerous tree viewers apply restrictions on the Newick format that are not standard. As such this file may not be readable by all tree viewers. We do try to constrain the data to work with as many viewers as possible, but we can in no way make it work for all.

Recommended Programs

This section will give a brief description of some programs that we suggest you use in order to view or edit the data we have passed along.

- Text Editors
  - Let us first note that a text editor and a word processor are vastly different programs. When we say text editor we are discussing programs that often do not allow for changes in font, size or stylization. As such we strongly advise you to avoid using Microsoft Word, OpenOffice Writer, Wordpad or Google Docs to view or edit any file we provide.
  - Recommended Text Editors (Paid Usage)
    - UltraEdit: http://www.ultraedit.com/
  - Recommended Text Editors (Free)
    - Notepad++: http://notepad-plus-plus.org/
    - Unix & Linux Text Editors: GEdit / vi / emacs
- Spreadsheet Applications
  - Recommended Spreadsheet Applications (Paid Usage)
    - Microsoft Excel: http://products.office.com/en-us/excel
    - Apple Numbers: http://www.apple.com/mac/numbers/
  - Recommended Spreadsheet Applications (Free)
    - OpenOffice Calc: http://www.openoffice.org/
- Browser
  - Krona supports most browsers however the developers suggest using FireFox.
  - FireFox: https://www.mozilla.org/en-US/firefox/new/
- Tree Viewer
  - Recommended Tree Viewer (Local Installation)
    - MEGA: http://www.megasoftware.net/
  - Recommended Tree Viewer (Web Based)
    - ETE Toolkit Tree Viewer: http://etetoolkit.org/treeview/
  - For a more comprehensive list of options, see the Wikipedia article on the topic: http://en.wikipedia.org/wiki/List_of_phylogenetic_tree_visualization_software
- Multiple Sequence Alignment Viewer
  - Recommended MSA Viewer (Local Installation)
    - MEGA: http://www.megasoftware.net/
  - Recommended MSA Viewer (Web Based)
    - MView: http://www.ebi.ac.uk/Tools/msa/mview/

- For a more comprehensive list of options, see the Wikipedia article on the topic: http://en.wikipedia.org/wiki/List_of_alignment_visualization_software

References

[1] S. M. Huse, J. A. Huber, H. G. Morrison, M. L. Sogin and D. M. Welch, "Accuracy and quality of massively parallel DNA pyrosequencing.," *Genome Biology,* vol. 8, no. 7, 2007.

[2] C. Quince, A. Lanzen, R. J. Davenport and P. J. Turnbaugh, "Removing Noise From Pyrosequenced Amplicons," *BMC Bioinformatics ,* vol. 12, no. 38, 2011.

[3] M. A. Quail, M. Smith, P. Coupland, T. D. Otto, S. R. Harris, T. R. Connor, A. Bertoni, H. P. Swerdlow and Y. Gu, "A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers," *BMC Genomics,* 2012.

[4] J. Zhang, K. Kobert, T. Flouri and A. Stamatakis, "PEAR: A fast and accurate Illumina Paired-End reAd mergeR," *Bioinformatics,* 2013.

[5] R. C. Edgar, "Search and clustering orders of magnitude faster than BLAST," *Bioinformatics,* pp. 1-3, 12 August 2010.

[6] R. C. Edgar, "UPARSE: highly accurate OTU sequences from microbial amplicon reads," *Nature Methods,* vol. 10, pp. 996-998, 2013.

[7] R. C. Edgar, B. J. Haas, J. C. Clemente, C. Quince and R. Knight, "UCHIME improves sensitivity and speed of chimera detection," *Oxford Journal of Bioinformatics,* vol. 27, no. 16, pp. 2194-2200, 2011.

[8] B. J. Haas, D. Gevers, A. M. Earl, M. Feldgarden, D. V. Ward, G. Giannoukos, D. Ciulla, D. Tabbaa, S. K. Highlander, E. Sodergren, B. Methé, T. Z. DeSantis, The Human Microbiome Consortium, J. F. Petrosino, R. Knight and B. W. Birren, "Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons," *Genome Research,* 2011.

[9] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research,* vol. 32, no. 5, pp. 1792-1797, 2004.

[10] R. C. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," *BMC Bioinformatics,* 2004.

[11] M. N. Price, P. S. Dehal and A. P. Arkin, "FastTree: computing large minimum evolution trees with profiles instead of a distance matrix.," *Molecular biology and evolution,* vol. 26, no. 7, pp. 1641-1650, 2009.

[12] M. N. Price, P. S. Dehal and A. P. Arkin, "FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments," *PLOS One,* 2010.

[13] N. A. Bokulich, J. R. Rideout, K. Patnode, Z. Ellett, D. McDonald, B. Wolfe, C. F. Maurice, R. J. Dutton, P. J. Turnbaugh, R. Knight and J. G. Caporaso, "An extensible framework for optimizing classification enhances short-amplicon taxonomic assignments," *Not Yet Published,* 2014.

[14] Q. Wang, G. M. Garrity, J. M. Tiedje and J. R. Cole, "Naive Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy.," *Applied and Environmental Microbiology,* vol. 73, no. 16, pp. 5261-5267, 2007.

[15] P. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G. Thallinger, D. J. V. Horn and C. F. Weber, "Introducing mothur: Open-source, platform-independent, community-supported software for describing and comparing microbial communities," *Appl Environ Microbiol,* vol. 75, no. 23, pp. 7537-41, 2009.

[16] P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski and M. J. L. d. Hoon, "Biopython: freely available Python tools for computational molecular biology and bioinformatics," *Bioinformatics,* vol. 25, no. 11, 2009.