Sequence analysis methods.

Sequence curation.

16S

The quality of all illumiuna R1 and R2 reads was assessed visually using fastqc [1]. Generally we observed a significant drop in read quality in the last 50-100bp of R2 and the last 10bp of R1. We trim the 5' end of R1 by 10bp and the 5' end of R2 by 70bp (we chose to trim as many bp as possible while still leaving an overlap that allowed reliable merging of R1 and R2 reads. Reads were then merged using FLASH [2]. After merging several hundred sequence were merged manually and the results compared to the FLASH merges to ensure efficacy of FLASH.

Following merging, fasta format sequences were extracted from fastq files and sequences that contained N's, homopolymer runs > 8bp or that were shorter than 400bp were removed. The remaining sequences were then passed to our open reference OTU picking and assigning workflow.

18S

Illumina R1 and R2 sequences returned included primers and adapters, so R1 and R2 reads were both trimmed by 30 bp (5' end). The reads were then merged using FLASH and the results compared to a random subsample of sequences that were merged by hand. Following merging, fasta format sequences were extracted from fastq files and sequences that contained N's, homopolymer runs > 8bp or that were shorter than 400bp were removed. The remaining sequences were then passed to our open reference OTU picking and assigning workflow.

ITS

For ITS, R1 sequences only were used. R1 included the ITS1 region, upon which we've based our workflow. Fasta files were extractd from fastq files and complete ITS1 regions were extracted from reads using ITSx [3]. Sequences that contained full ITS1 regions were passed to the OTU picking and assigning workflow.

Open OTU picking and assignment

All amplicons were submitted to the same workflow to pick OTU's and assign abundance of reads to an OTU x Sample matrix. Ostensibly we followed a similar conceptual outline to that advocated in the QIIME open reference OTU picking pipeline (http://qiime.org/tutorials/open_reference_illumina_processing.html). Differences were a) we didn't

assign to reference OTU's initially via a round of closed reference picking, but instead pick de_novo otus (we classify OTU's later), b) in order make compute time for de novo picking manageable (because we didn't perform the closed round) we initially cluster OTU's on the numerically dominant sequences only (that is sequences that have > 4 representatives when we dereplicate the dataset), c) instead of randomly picking sequences that fail to be recruited to OTU's for subsequent clustering, we use all sequences with >2 representatives after dereplicating. We primarily used USEARCH to perform our analyses, but other programs could be equally efficacious. Our workflow, thus, follows these step

1. Ensure all sequences have appropriate identifiers for the tools to be employed downstream (e.g., if using usearch add "barcodelabel=#;" etc.
2. Dereplicate sequences
3. Sort sequences by abundance and keep sequences with >4 representatives*
4. Cluster sequences into OTU's (97%) using USEARCH -cluster_otus [4]. Chimera checking is done at this stage. Output both representative OTU sequence file, but also uparse file.
5. Cluster chimeric sequences to produce a representative sequences file for each OTU cluster (97%) [5]. Do this by using the .uparse output from (4) to obtain chimeric reads and cluster these using usearch -cluster_fast.
6. Concatenate denovo OTU's from (4) and chimeric OTU's from (5) into a single OTU.fasta mapping file.
7. Map reads in the original dataset of quality checked seqences (1) against this database using ueasearch -usearch_global
8. Get mapped reads (hits) from .uc output and split these into chimeric and non-chimeric .uc files
9. Get non-mapped reads (misses) from .uc output, and retrieve these from the original data to create a dataset of nonmapped/nonchimeric reads.
10. Repeat the process from step (2) with this new file, reducing the number of required representatives per sequence at step (3) appropriately (e.g., from 4 to 2).
11. After all clustering and mapping is done concatenate the resultant .uc files from all mapped hits
12. Convert this file to an OTU table
13. Concatenate all representative OTU sequence files and identify OTU's using your favorite database (we used Green genes for Bacteria (16S) , Silva for Eukaryotes (18S) and archaea (16S) and Unite for fungi (ITS). OTU fasta sequences are available on the BASE data repository and can be re-classified by users as required.

*It should be noted that 4 may not be the best number for you to use, depedning on the size of your dataset and compute resources.

1. Andrews, S., *FastQC A Quality Control tool for High Throughput Sequence Data.* http://www.bioinformatics.babraham.ac.uk/projects/fastqc/, 2010.
2. Magoč, T. and S.L. Salzberg, *FLASH: Fast Length Adjustment of Short Reads to Improve Genome Assemblies.* Bioinformatics, 2011.
3. Bengtsson-Palme, J., et al., *Improved software detection and extraction of ITS1 and ITS2 from ribosomal ITS sequences of fungi and other eukaryotes for analysis of environmental sequencing data.* Methods in Ecology and Evolution, 2013. **4**(10): p. 914-919.
4. Edgar, R.C., *UPARSE: highly accurate OTU sequences from microbial amplicon reads.* Nat Meth, 2013. **10**(10): p. 996-998.
5. Edgar, R.C., *Search and clustering orders of magnitude faster than BLAST.* Bioinformatics, 2010. **26**(19): p. 2460-2461.