# Metagenome Data Analysis

**EJ Garza**
BI-GY 7583
Summer 2019

**Project Overview**

Metagenome data analysis is an effective method for leveraging high throughput DNA sequencing to characterize microbial communities in environmental samples. Presented in this project is the application of an analysis pipeline to reproduce microbial community composition studies on publicly available data from a published research paper. Challenges that arose and feedback regarding the model are summarized as well as experiences implementing related bioinformatics processes.

## Introduction

In applied bioinformatics one should strive for computational pipelines which are easy to adopt, adapt, and automate. This work presents an attempt to recreate a published metagenomics workflow related to the study of microbial communities from environmental samples in Alaska by Hue et al in 2016. [1] ("the model" or "the reference"). The nature of the samples in the publication were intriguing because of possible similarity to those my lab may encounter when studying bacterial, archaeal, fungal, or viral diversity and gene function [2] of soil samples. First an overview of key metagenomics data analysis aspects as they apply to this pipeline will be described, then the methods in the Hue et al [1] pipeline are summarized, finally the results of my attempt to recreate the previously referenced model with two different types of metagenomics data (shotgun whole genome data and 16S marker gene data).

## Background

Metagenome analysis is a type of bioinformatics analysis that has increased in popularity and demand in no small part because of the economic feasibility and accessibility of high throughput DNA sequencing technology. For example, in a paper published by Nurul, Ashyikin Noor Ahmad et al.[4] titled "16S rRNA-Based metagenomic analysis of microbial communities associated with wild *Labroides dimidiatus* from Karah Island, Terengganu, Malaysia." the findings demonstrate the wide range of applications possible with metagenome data analysis. This is a single example among countless others. Computational metagenome analysis occurs after experimental design, environmental or patient sampling, and processing via high-throughput sequencing. Metagenome studies are also characterized by software which requires data input from the sequencing machine in the form of text files representing thousands or millions of text strings each with hundreds or thousands of characters where each character provides the identity for a single nucleotide base from some organism in the physical sample [5]. What makes metagenomes complicated to analyze are the unknown large number of organism DNA being decoded by the sequencer that must be "*denoised*" by algorithms [6]. Results for a metagenome project could be an inventory by taxonomy of the organisms found in the samples because their taxonomic classification could indicate some degree of function/role to the greater community. Other results may be abundance information counting how many times each species or strain was identified in the sample. Metagenome analysis can and should also be accompanied by metadata describing environmental conditions for comparative community analysis or multivariable correlation studies. Yet another focus of metagenome analysis could be to identify candidate organisms which are those that do not yet have a published annotated reference genome. For these as well as other types of metagenome data analysis there are multiple combinations of software packages and scripting languages that can be used to produce key findings and new discoveries.

## Pre-Processing

It's best practice and recommended to first examine and preprocess genomic reads before assembly or other computation [7]. These quality control steps ensure that the various algorithms which underlie most of bioinformatics use precious compute cycles on data that is likely to provide meaningful results. Short read quality control involves reviewing metrics about raw reads generated by the sequencer such as read length and base quality score profiles. It may be necessary to remove sequences known as adapter sequences that are not part of the sample but instead used by the machine to begin the chemical reaction of sequencing. Similarly, careful consideration should be given to remove bases which do not meet a certain quality and reads which do not satisfy a certain length. After quality filtering reads other changes may be necessary depending on how the sequencer produced the files and how downstream programs require those files be structured. Implementation specific format examples include paired-end FASTQ files with a certain file naming scheme or single-end FASTQ files among others.

## Assembly

Genome assembly of quality assessed short read sequencing data is an integral aspect of metagenome data analysis. There are many assemblers available in many programming languages and there are also a lot of research papers benchmarking them with synthetic and real-world data sets. Deciding which assembler to use will mostly depend on experimental design and sequencing platform. Assembler algorithms can work in many ways, but two common methods are clustering and denoising [6]. Clustering or OTU (Operational Taxonomic Unit) picking groups reads together based on a user defined similarity threshold (e.g., 97%). Sequences sharing similarity within the threshold are treated as the same taxonomic unit and the differences between them can be attributed as sequencing errors or biological differences between species or strains. On the other hand, Denoising incorporates error models that can be specific to the sequencing platform and quality scores for each nucleotide base to iteratively resolve amplified sequence variants and correct read errors. Two possible advantages of denoising over OTU picking is that the result of denoising is finer grained providing greater detail about the sequences as well as avoiding clustering thresholds that could differ between analysis runs if different parameters are used.

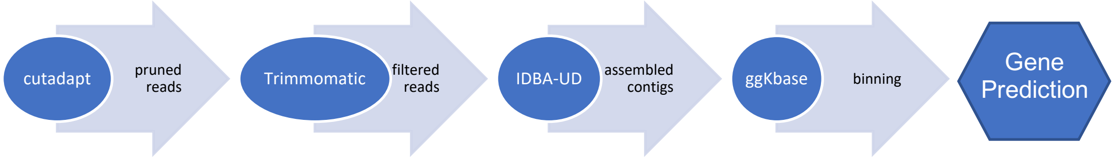
### Binning

After the assembly of high-quality reads into contiguous overlapping reads, metagenomic binning methods are used to derive the microbial taxonomies [8]. Binning strategies vary however most fall into approaches known as supervised, un-supervised, or hybrid. The supervised, or closed-reference, approach is accurate and should be faster than un-supervised approaches because a database of reference genomes is used to match sample contigs against. If a contig search against the reference database returns a hit, then the reference taxonomy name can be retrieved otherwise if there is no match from the database then the read may belong to an unpublished or unannotated gene. The unsupervised (also known as statistical or heuristic) approach lacking a reference database attempts to compute a fingerprint for reads from samples such as tetranucleotide frequencies or contig abundance correlation to group or bin the reads. Unsupervised binning methods are generally not as accurate as supervised methods and require a greater number of samples as input to start with for more accurate genomic fingerprinting. Some binning programs also require abundance data related to the contigs while others will compute that for you.

There is no off-the-shelf or one-size-fits-all approach to metagenomic data analysis and so far only a portion of all possible analysis steps are covered. Depending on the study, an alignment algorithm such as bowtie may also be required to map reads or contigs back onto a reference dataset. Gene prediction may be a goal of the study as well which requires other computation. Yet another objective may be pathway analysis or differential expression studies which can provide biological evidence to support nutrient transfer or functional behavior above and beyond the inventory of community taxonomies. Next, details of the target research pipeline are covered.

## Methods in Reference Pipeline

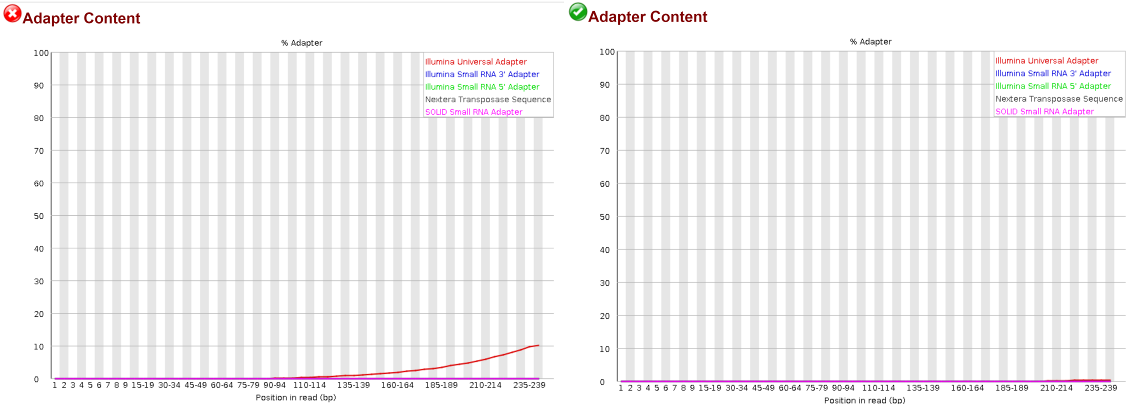
Hu P et al. [1] collected 6 samples from 3 oil well sites in Alaska and used Illumina technology to generate reads. The paper was published in 2016 but the samples were collected in 2011 and 2013 with the raw (straight from the sequencer and unfiltered) generated reads being deposited to NCBI Sequence Read Archive for public accessibility. Figure 1 of the publication shows a bar plot with color coded portions of each sample representing categories of bacterial community members and the methods section of the article describes the tools and versions used to generate community abundance visualizations. The paper listed cutadapt to remove adapter sequences from the raw reads, Trimmomatic to perform quality filtering, IDBA for genome assembly, EMIRGE for identification of 16S marker gene reads from the shotgun metagenome reads, and the ggKbase online platform to bin and classify genes. The ggKbase platform cites Meta-Prodigal as the gene prediction tool which reportedly includes the ability to perform functional pathway mapping. The high-level analysis approach used in the reference could help design a custom version of the same pipeline.



*Overview of Hue et al [1] data analysis*

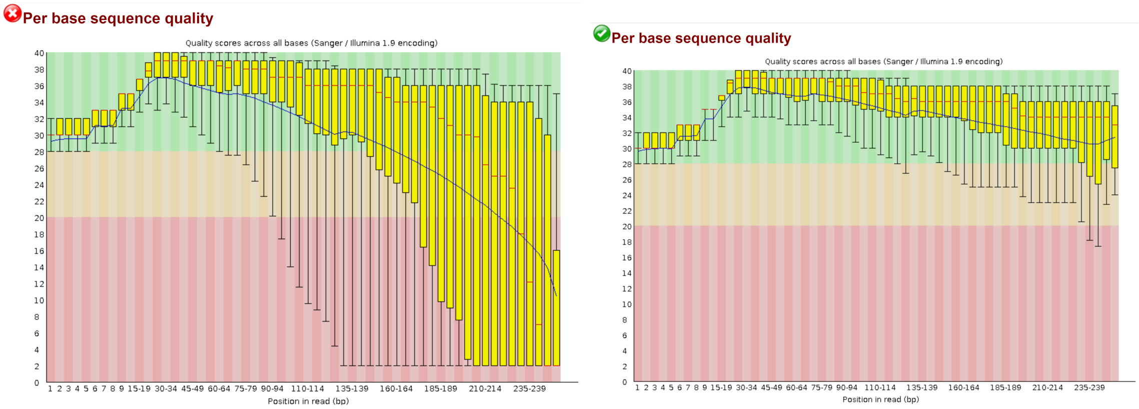
## Metagenome Sequence Analysis

Raw sequencing reads were acquired from the Sequence Read Archive at The National Center for Biotechnology Information using fastq-dump [9]. Six paired-end samples were downloaded from the repository with Accession IDs SRR1976948, SRR1977249, SRR1977296, SRR1977304, SRR1977357, and SRR1977365. Anaconda package manager and repository (https://anaconda.org/) was used for a majority of the installation of bioinformatics software packages and respective dependencies. FastQC was run in batch mode to generate reports on unrefined sequencing reads which identified adapter sequences needed to be pruned. Adapter removal was done with cutadapt version 2.3 [12] and the figures below show the adapter content report before (left side plot) and after (right side plot) processing one of the samples with cutadapt.



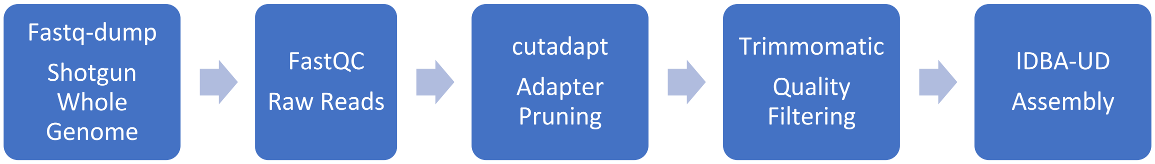
*FastQC Adapter Content Report for SRR1976948_2 indicating Illumina Universal Adapters were detected in the sequence before running cutadapt v2.3.*

Trimmomatic version 0.36 [13] filtered reads according to the parameters specified in the original article, “(parameters: leading, 3; trailing, 3; sliding window, 4; quality score, 15; minimum read length of 60 bases).” FastQC was run again to generate a new per base sequence quality report. On the left is the median PHRED base score distribution before trimming and the same report after trimming on the right.



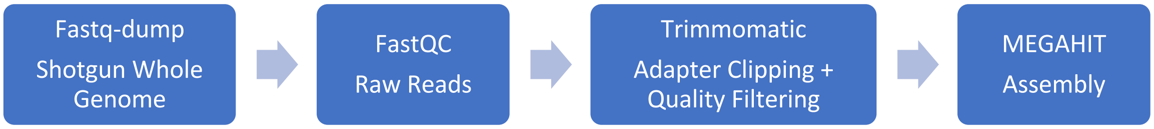
*FastQC Per base Sequence Quality Report for SRR1976948_2 indicating higher quality reads after filtering with Trimmomatic v0.36*

Assembly of the trimmed reads was performed using IDBA-UD [10] which is named after the graph theory approach it employs to solve genome assembly problems, the de Bruijn graph [18]. Once the reads were assembled into contigs it's assumed the paper follows steps prescribed by ggKbase (https://ggkbase-help.berkeley.edu/) that must be followed before upload to a ggKbase server for binning analysis. The ggKbase web service restricts access to submit project groups (which is simply a group of samples) to certain affiliated project members or user accounts creating a road block was not anticipate when first considering the project. The documentation on ggKbase lists quality control, sequence assembly, read alignment onto contigs (with bowtie), gene prediction and predicted gene annotation as the steps that must be completed before uploading. The gene prediction and annotation steps point to internally created python scripts and system paths which are also restricted. The gene annotation step uses KEGG, UniRef100, and UniProt reference databases with the usearch (https://www.drive5.com/usearch/) package which costs $1,500 for a 64-bit license and $0 for a 32-bit license. It's not clear what ggKbase does computationally after creating a project group and submitting data. Presumably, another pipeline could be designed based on the same binning functionality in ggKbase (if it were published) with newer software modules. It is worth noting that the ggKbase website mentions their software and procedures are all open source. After assembly the paper also describes another analysis workflow that uses the contigs as input to the EMIRGE [14] package to find 16S marker genes. EMIRGE requires usearch as a dependency as well but also requires python version 2.6 which is deprecated. Open source tools which will remain published, supported and are free of charge are important to identify and build around for the sake of easily reproducible analysis. In light of these new challenges progress could no longer proceed exactly as described in the reference article however other assemblers and data sets were identified to look at how contemporary metagenome pipelines may be put together. The diagram below is an overview of what this project accomplished in regard to reconstructing the pipeline described in the reference paper.



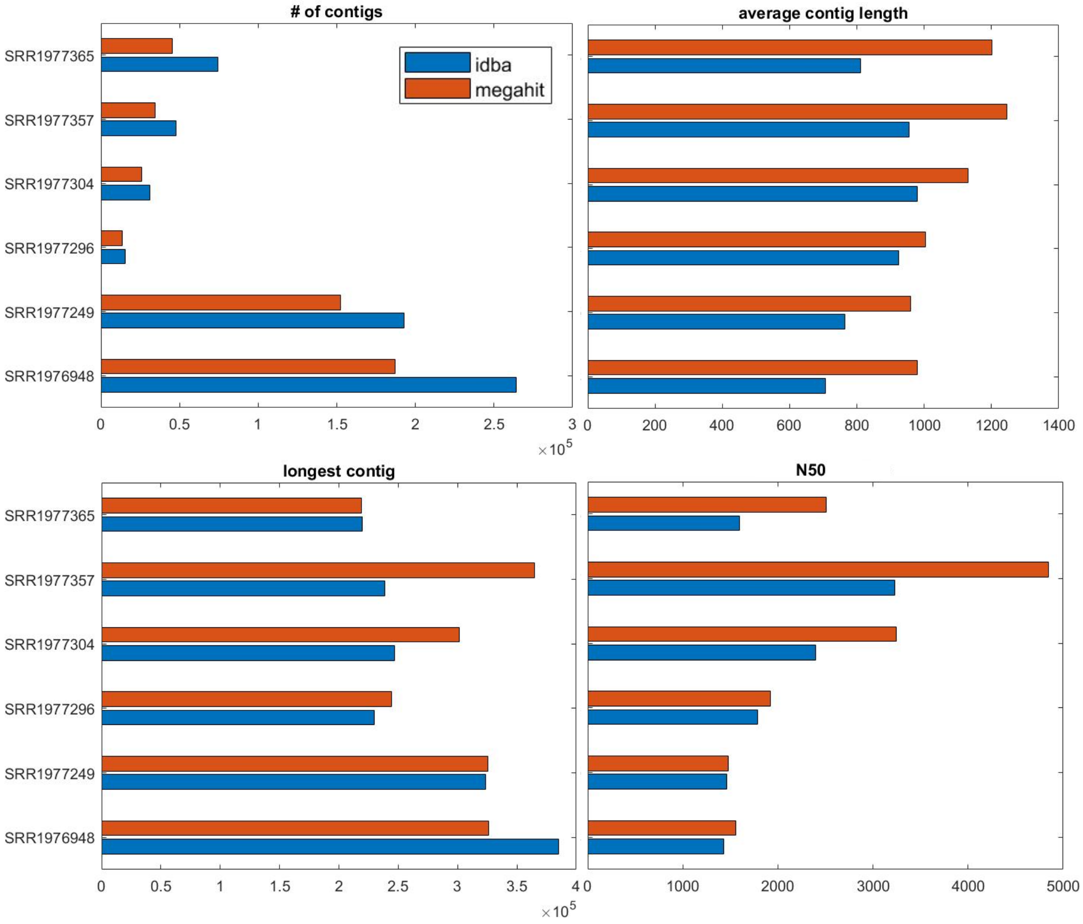
*Overview of completed steps from model pipeline achieved in this project*

Research into newer de Bruijn graph based genome assemblers capable of processing shotgun metagenome short reads produced by Illumina led to a tool named MEGAHIT [11]. MEGAHIT is capable of truncating and trimming reads but the reads were pre-filtered in much the same fashion the reference paper described. Below is a diagram depicting how MEGAHIT was used for this project.



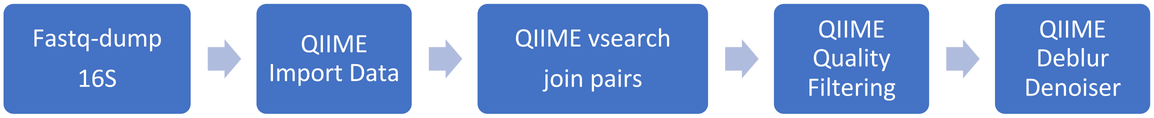
*Overview of a newer and simpler approach to metagenomic contig assembly using MEGAHIT [11]*

Completion time for both assemblers were comparable running on a single node with 55 GB of memory and 12 CPUs. Other measures are presented below.



MEGAHIT seems to outperform IDBA by assembling longer contigs from the same dataset. MEGAHIT produced larger N50 scores and also had a longer average contig length across all samples. Note that these metrics include all contigs and not just contigs over 500 base pairs in length as commonly measured in published literature.

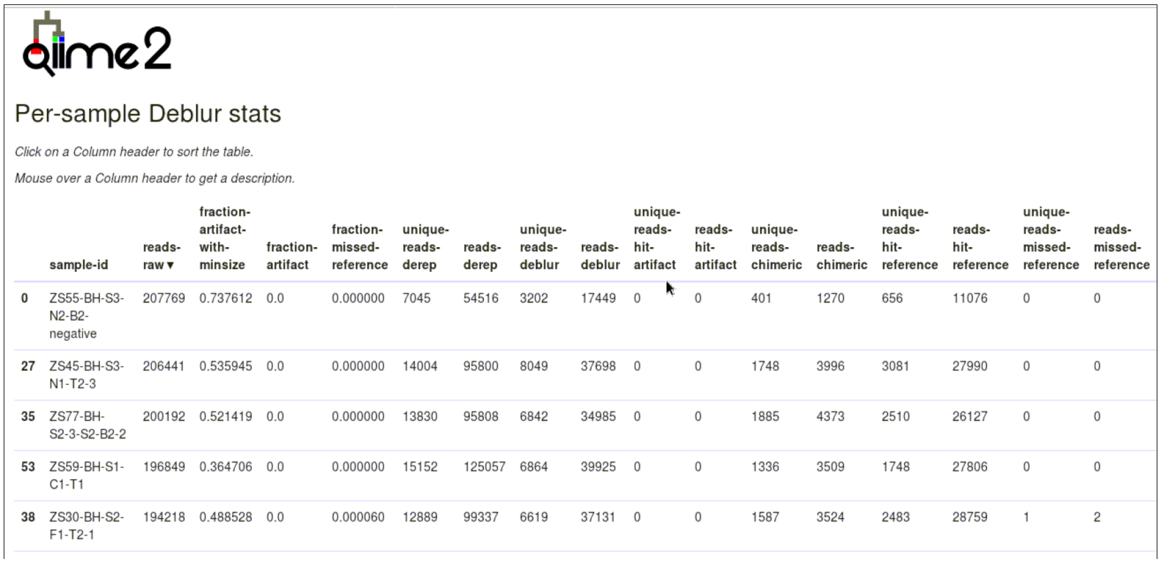
Analysis of 16S marker gene reads was performed on another SRA data set. QIIME2 command line interface (CLI) version 2019.4 [15] was used to convert raw reads into QIIME artifacts for assembly with two different QIIME plug-in denoising packages, DADA2 [3] and Deblur [16]. The Deblur assembler required preprocessing which was managed easily by QIIME plug-ins. Here is an overview of the process to running Deblur in QIIME2.



Fastq-dump 16S → QIIME Import Data → QIIME vsearch join pairs → QIIME Quality Filtering → QIIME Deblur Denoiser

*Process to run Deblur QIIME2 Plug-In using QIIME2 version 2019.4*

The QIIME2 package includes a feature to report on Deblur assembly statistics, below is part of a summary of the stats table sorted by raw read count.



qiime2

Per-sample Deblur stats

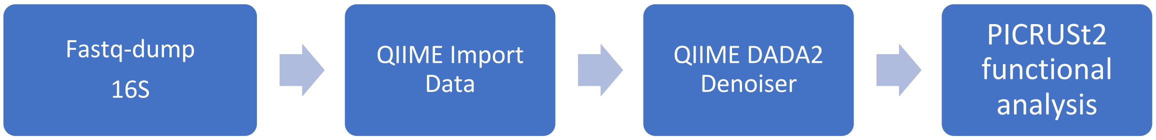
*Click on a Column header to sort the table.*

*Mouse over a Column header to get a description.*

| | sample-id | reads-raw ▼ | fraction-artifact-with-minsize | fraction-artifact | fraction-missed-reference | unique-reads-derep | reads-derep | unique-reads-deblur | reads-deblur | unique-reads-hit-artifact | reads-hit-artifact | unique-reads-chimeric | reads-chimeric | unique-reads-hit-reference | reads-hit-reference | unique-reads-missed-reference | reads-missed-reference |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | ZS55-BH-S3-N2-B2-negative | 207769 | 0.737612 | 0.0 | 0.000000 | 7045 | 54516 | 3202 | 17449 | 0 | 0 | 401 | 1270 | 656 | 11076 | 0 | 0 |
| **27** | ZS45-BH-S3-N1-T2-3 | 206441 | 0.535945 | 0.0 | 0.000000 | 14004 | 95800 | 8049 | 37698 | 0 | 0 | 1748 | 3996 | 3081 | 27990 | 0 | 0 |
| **35** | ZS77-BH-S2-3-S2-B2-2 | 200192 | 0.521419 | 0.0 | 0.000000 | 13830 | 95808 | 6842 | 34985 | 0 | 0 | 1885 | 4373 | 2510 | 26127 | 0 | 0 |
| **53** | ZS59-BH-S1-C1-T1 | 196849 | 0.364706 | 0.0 | 0.000000 | 15152 | 125057 | 6864 | 39925 | 0 | 0 | 1336 | 3509 | 1748 | 27806 | 0 | 0 |
| **38** | ZS30-BH-S2-F1-T2-1 | 194218 | 0.488528 | 0.0 | 0.000060 | 12889 | 99337 | 6619 | 37131 | 0 | 0 | 1587 | 3524 | 2483 | 28759 | 1 | 2 |

*Deblur statistics compiled by QIIME2 version 2019.4*

QIIME2 also provides denoising with DADA2. DADA2 requires less pre-work because it provides features that allow users to filter and assemble with one command line tool.  After DADA2 was used to create a feature table tracking the abundance of each amplicon sequence variant a tool name PICRUSt2 [17] was used to perform functional gene analysis. The output created by PICRUSt2 nearly meets the outputs of the reference pipeline because newer tools such as PICRSt2 can use DNA data to map reads to a reference phylogenetic tree and reconstruct what is missing from the tree after all community members have been aligned. This provides an estimated view of which organisms are present and what role they may be providing to the rest of the community organisms. Here is an overview of the steps to perform functional microbial community analysis with QIIME2, DADA2, and PICRUSt2.



*Functional analysis pipeline for 16S marker gene reads using QIIME2, DADA2, and PICRUSt2*

Working on a computational metagenomics analysis pipeline builds insights into the challenges of open-source software management, research article examination, software manual inspection, and scripting. After realizing the license costs for usearch which was used for both the gene prediction and 16S aspects of the model pipeline coupled with the black-box approach to binning used by the ggKbase platform, consideration into alternate methods to analyze the reference data became a priority. The lesson here is to look deeper into methods and results of papers before attempting to reconstruct their experiment. This technical skill is valuable because in many cases microbiologists or ecologists need an understanding of the computational framework however are too busy to deal with the low level implementation details. For these reasons a close working relationship between lab directors and senior scientists with infrastructure support and systems developers could prove immensely valuable.

### Source Code

All source code, scripts, and reports used for this analysis can be found in GitHub (https://github.com/ejgarza31/bio). Development originally took place on a private HPC cluster so references to file paths and software modules are specific to the private platform. It's possible the Anaconda package manager continues to be a preferred platform for open-source package management because of the ability to provide most bioinformatics development runtimes such as Python, R/CRAN/Bioconductor, and Perl.

## Future Work

Future work will consist of identifying and running a binning program, such as MaxBin [20], with the assembled contigs from all four assemblers, DADA2, IDBA-UD, MEGAHIT, and Deblur. Furthermore, to match the analysis carried out in the research article a substitute for EMIRGE to reconstruct marker gene sequences from raw metagenome reads needs to be identified, perhaps phyloFlash (https://github.com/HRGV/phyloFlash) although it often takes several attempts to find a suitable package. With the results from MaxBin and phyloFlash as input into QIIME2 artifacts plots such as those appearing in the publication should be possible to create as output providing a visual inspection of results between original findings and this analysis workflow.

### Acknowledgements

Thank you to some of the QIIME 2 developers who provided the training workshop on Microbial Communities Profiling via QIIME 2 and Qitta at Columbia University in New York City in June 2019. The experience was insightful and provided initial motivation for a project topic. I'd also like to acknowledge co-workers in my lab who are experts in sequence analysis and High Performance Computing and were kind enough to consult with me about my choice of project topic and offer general advice along the way.

# References

[1] Hu P *et al.,* "Genome-Resolved Metagenomic Analysis Reveals Roles for Candidate Phyla and Other Microbial Community Members in Biogeochemical Transformations in Oil Reservoirs.", *MBio*, 2016 Jan 19;7(1):e01669-15

[2] Mande, Sharmila S.; Monzoorul Haque Mohammed; Tarini Shankar Ghosh (2012). "Classification of metagenomic sequences: methods and challenges". Briefings in Bioinformatics. 13 (6): 669–81.

[3] Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. 2016. DADA2: high-resolution sample inference from Illumina amplicon data. Nat Methods 13:581–583. doi:10.1038/nmeth.3869.

[4] Nurul, Ashyikin Noor Ahmad et al. “16S rRNA-Based metagenomic analysis of microbial communities associated with wild *Labroides dimidiatus* from Karah Island, Terengganu, Malaysia.” *Biotechnology reports (Amsterdam, Netherlands)* vol. 21 e00303. 4 Jan. 2019, doi:10.1016/j.btre.2019.e00303

[5] Buffalo, Vince, “Bioinformatics Data Skills”, O’Reilly Media Inc., 2015.

[6] Nearing, J.T.; Douglas, G.M.; Comeau, A.M.; Langille, M.G. Denoising the Denoisers: An independent evaluation of microbiome sequence error-correction approaches. PeerJ 2018, 6, e5364.

[7] Flowers, Jonathan M. “Week3_short_read_QC” Methods in Next Generation Sequence Analysis BIGY 7653 6 Sep. 2018. New York University. Microsoft PowerPoint presentation.

[8] Yi Wang, Henry C.M. Leung, S.M. Yiu, Francis Y.L. Chin, MetaCluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample, *Bioinformatics*, Volume 28, Issue 18, 15 September 2012, Pages i356–i362, https://doi.org/10.1093/bioinformatics/bts397

[9] Leinonen, Rasko et al. “The sequence read archive.” *Nucleic acids research* vol. 39, Database issue (2011): D19-21. doi:10.1093/nar/gkq1019

[10] Peng Y, Leung HC, Yiu SM, Chin FY. 2012. IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. Bioinformatics 28:1420–1428. http://dx.doi.org/10.1093/bioinformatics/bts174.

[11] MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, Tak-Wah Lam. Bioinformatics. 2015 May 15; 31(10): 1674–1676. Published online 2015 Jan 20. doi: 10.1093/bioinformatics/btv033

[12] MARTIN, Marcel. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, [S.l.], v. 17, n. 1, p. pp. 10-12, may 2011. ISSN 2226-6089. Available at: <http://journal.embnet.org/index.php/embnetjournal/article/view/200>. Date accessed: 11 aug. 2019. doi:https://doi.org/10.14806/ej.17.1.200.

[13] Anthony M. Bolger, Marc Lohse, Bjoern Usadel, Trimmomatic: a flexible trimmer for Illumina sequence data, Bioinformatics, Volume 30, Issue 15, 1 August 2014, Pages 2114–2120, https://doi.org/10.1093/bioinformatics/btu170

[14] Miller CS, Baker BJ, Thomas BC, Singer SW, Banfield JF (2011) EMIRGE: reconstruction of full-length ribosomal genes from microbial community short read sequencing data. Genome biology 12: R44. doi:10.1186/gb-2011-12-5-r44.

[15] Bolyen E, Rideout JR, Dillon MR, Bokulich NA, Abnet C, Al-Ghalith GA, Alexander H, Alm EJ, Arumugam M, Asnicar F, Bai Y, Bisanz JE, Bittinger K, Brejnrod A, Brislawn CJ, Brown CT, Callahan BJ, Caraballo-Rodríguez AM, Chase J, Cope E, Da Silva R, Dorrestein PC, Douglas GM, Durall DM, Duvallet C, Edwardson CF, Ernst M, Estaki M, Fouquier J, Gauglitz JM, Gibson DL, Gonzalez A, Gorlick K, Guo J, Hillmann B, Holmes S, Holste H, Huttenhower C, Huttley G, Janssen S, Jarmusch AK, Jiang L, Kaehler B, Kang KB, Keefe CR, Keim P, Kelley ST, Knights D, Koester I, Kosciolek T, Kreps J, Langille MG, Lee J, Ley R, Liu Y, Loftfield E, Lozupone C, Maher M, Marotz C, Martin BD, McDonald D, McIver LJ, Melnik AV, Metcalf JL, Morgan SC, Morton J, Naimey AT, Navas-Molina JA, Nothias LF, Orchanian SB, Pearson T, Peoples SL, Petras D, Preuss ML, Pruesse E, Rasmussen LB, Rivers A, Robeson, II MS, Rosenthal P, Segata N,

Shaffer M, Shiffer A, Sinha R, Song SJ, Spear JR, Swafford AD, Thompson LR, Torres PJ, Trinh P, Tripathi A, Turnbaugh PJ, Ul-Hasan S, van der Hooft JJ, Vargas F, Vázquez-Baeza Y, Vogtmann E, von Hippel M, Walters W, Wan Y, Wang M, Warren J, Weber KC, Williamson CH, Willis AD, Xu ZZ, Zaneveld JR, Zhang Y, Zhu Q, Knight R, Caporaso JG. 2018. QIIME 2: Reproducible, interactive, scalable, and extensible microbiome data science. PeerJ Preprints 6:e27295v2 https://doi.org/10.7287/peerj.preprints.27295v2

[16] Amir A, McDonald D, Navas-Molina JA, Kopylova E, Morton JT, Zech Xu Z, Kightley EP, Thompson LR, Hyde ER, Gonzalez A, Knight R. 2017. Deblur rapidly resolves single-nucleotide community sequence patterns. mSystems 2:e00191-16. https://doi.org/10.1128/mSystems.00191-16.

[17] Gavin M. Douglas, Vincent J. Maffei, Jesse Zaneveld, Svetlana N. Yurgel, James R. Brown, Christopher M. Taylor, Curtis Huttenhower, Morgan G. I. Langille. 2019. PICRUSt2: An improved and extensible approach for metagenome inference. bioRxiv, doi: https://doi.org/10.1101/672295.

[18] Mishra, Bud. “Algorithms and Data Structures: For Bioinformatics Lecture #5” Algorithms and Data Structures for Bioinformatics BIGY 7453 27 Feb. 2018. New York University. Microsoft PowerPoint presentation.

[19] MetaBAT, an efficient tool for accurately reconstructing single genomes from complex microbial communities. 2015. doi:10.7717/peerj.1165.

[20] Wu et al.: MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. Microbiome 2014 2:26.

[21] Microbiome/Metagenome Analysis Workshop: Introduction to Metagenomics on YouTube posted by Brown University on April 13, 2018 viewed on August 3, 2019.