LOTUS: A WEB-BASED COMPUTATIONAL TOOL FOR THE PRELIMINARY INVESTIGATION OF A NOVEL MST METHOD UTILIZING A LIBRARY OF 16S RRNA *BACTEROIDES* OTUS

A Thesis

presented to

the Faculty of California Polytechnic State University,

San Luis Obispo

In Partial Fulfillment

of the Requirements for the Degree

Master of Science in Computer Science

by

Ginger DeWitte

May 2021

COMMITTEE MEMBERSHIP

TITLE:	LOTUS: A Web-Based Computational Tool for the Preliminary Investigation of a Novel MST Method Utilizing a Library of 16S rRNA *Bacteroides* OTUs
AUTHOR:	Ginger DeWitte
DATE SUBMITTED:	May 2021
COMMITTEE CHAIR:	Alexander Dekhtyar, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Chris Kitts, Ph.D. Professor of Biological Sciences
COMMITTEE MEMBER:	Michael Black, Ph.D. Professor of Biological Sciences
COMMITTEE MEMBER:	Lubomir Stanchev, Ph.D. Professor of Computer Science
COMMITTEE MEMBER:	Paul Anderson, Ph.D. Professor of Computer Science

ABSTRACT

LOTUS: A Web-Based Computational Tool for the Preliminary Investigation of a Novel MST Method Utilizing a Library of 16S rRNA *Bacteroides* OTUs

Ginger DeWitte

Microbial Source Tracking (MST) is a field of study that attempts to identify the source of fecal contamination in waterways in order to assist with development of remediation strategies. Biologists at Cal Poly Center for Applications in Biotechnology (CAB) are developing a new MST method using microbes from the genus *Bacteroides*. *Bacteroides* species are host-specific microorganisms that can theoretically be used to trace back to a single host species. After fecal samples are collected, biologists use Next-Generation Sequencing (NGS) techniques to obtain only the genetic sequences of microorganisms belonging to the phylum Bacteroidetes. Investigators hypothesize that similar sequences belong to the same phlyogenetic group (i.e., the same genus) and can therefore be computationally clustered. Each cluster of related sequences, typically 97% similar, is called an Operational Taxonomic Unit (OTU). Theoretically, an OTU acts as a molecular signature that can be traced back to a specific host genus. This thesis presents LOTUS, the **L**ibrary of **OTUs**, a web-based computational tool for the preliminary investigation of the use of the *Bacteroides* OTU library as an MST method. This work discusses the four contributions of LOTUS: a database design which accurately models OTUs and the underlying relationships necessary for source tracking, a pipeline to create OTUs from raw sequencing reads, a method of assigning taxonomy to OTUs, and a web-based user interface. In preliminary testing for a reference library of twelve samples, LOTUS produced 1,431 OTUs, of which 891 were single-source (OTUs derived from sequences from a single host species). Using these OTUs, LOTUS was able to accurately taxonomically match four of five unknown test samples, showing promise for using OTUs as an MST method.

ACKNOWLEDGMENTS

I would like to thank the following people who all had a part in helping me achieve my goals:

- My family for their endless patience, unconditional love, unwavering support, and steadfast encouragement
- Michael Wagner for opening a new world to me by encouraging me to explore a Master's in CS
- Alex Dekhtyar for his enthusiasm for bioinformatics, his warm-hearted welcome into the grad program, brilliant advice, gentle humor in dealing with my procrastination and freak outs, his unflagging support, and being the greatest advisor I could have hoped for
- Dr. Chris Kitts & Dr. Michael Black for granting me the privilege of working with them on this project, for their encouragement, astute advice, and especially for their patience in answering my biology questions and in seeing this thesis to completion
- Kurt Mammen for giving me the awesome experience of being a T.A. and for fun times grading 357 tests
- Tram Lai for being a friend from the beginning of our CS journey together, for getting me through the hard classes, for staying up coding through the night, and for making me laugh through it all
- Di Hoang for being my go-to class partner, for introducing me to escape rooms, and for becoming a lasting friend

- Leanne Fiorentino, the woman behind the curtain whose magic makes the CS department work, for her knowledge, wit, humor, and friendship which made my CS journey not only possible but fun
- Diana Wilson, the maven of the CSL, who became a friend through many early mornings spent together while I was working on project deadlines
- Kat Axelsson, Eric Thorndyke & Tim Scott for sticking with me and teaching me the real world side of CS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Illnesses from food and water contamination are a major public health concern. Studies used by the Centers for Disease Control and Prevention (CDC) estimate 477,000 illnesses and almost 7,000 deaths annually due to waterborne diseases in the United States alone [21, 3, 54]. A broader 2018 study of recreational waterborne illness puts the estimate at 90 million illnesses annually [29]. Several known disease-causing organisms called pathogens are transmitted via water including *Campylobacter*, *Cryptosporidium*, *Giardia*, *Escherichia coli* O157:H7, *Salmonella* (the causative agent of typhoid), *Leptospira*, and *Vibrio* (the causative agent of cholera) [84, 108, 54, 133, 55, 46]. These pathogens can be excreted in the feces of animal or human sources and contaminate recreational or drinking water [84, 50]. The World Health Organization (WHO) estimates that 1.8 billion people worldwide use drinking water contaminated by feces causing an estimated 1.9 million preventable deaths [154, 10, 155]. Identifying the source of fecal contamination in the food and water supply is of paramount importance for developing prevention and remediation strategies to reduce both the human health risk and economic impact of these diseases [134].

Since public health is concerned with the prevention of disease, resource managers would preferably like to measure pathogens in the water. But given the diversity of pathogens and the potential difficulty in their culture[1] and identification, direct pathogen testing is infeasible; so instead, legislative solutions have focused on fecal indicator bacteria (FIB) [47, 95]. These bacteria do just what their name implies - indicate the presence of fecal matter. Scott et al. [134] and Field et al. [47] describe criteria for ideal indicator bacteria as: 1) strong association with the presence

[1] In biology, growing a colony of a bacteria on a growth medium (such as an agar plate) in an incubator is called culturing.

of pathogens, 2) rapidly and easily detectable, 3) non-pathogenic, and 4) similar survival characteristics to the pathogens of interest (i.e., the indicator should not reproduce in the environment outside of the host). Traditional FIB include total and fecal coliforms, *Escherichia coli* (abbreviated as *E. coli*), and fecal enterocooci [47, 62, 12, 138, 134, 142]. These FIB can co-exist with pathogenic organisms in the gastrointestinal tracts of humans and animals and are used as proxies to determine fecal contamination by pathogens. Hence, the presence of FIB in the water can be predictive of public health risk. Importantly, FIB alone cannot identify the source of fecal contamination as the FIB mentioned above are generally found across multiple species. However, if either FIB themselves or particular strains or other attributes of FIB can be associated with a specific host species, then the source can be identified [47, 62, 12].

Microbial Source Tracking (MST) is an active area of biological research that includes a variety of forensic methods using genotype, phenotype, or other chemical or biological characteristics to trace an environmental microorganism back to its specific animal host, or more precisely, the host animal's species [12, 121, 47, 126, 142]. MST methods that utilize FIB are based on the premise that microorganisms exist which are specific to their host species and that these microbes have some characteristic which can be used as a marker for fecal contamination from the host species [12, 62, 47, 148]. In simple terms, the presence of FIB answers the question of "Is the water contaminated?" while MST techniques attempt to answer the question "What (or who) is the source of the contamination?". It should be noted that although MST was developed for and has been primarily used in aquatic environments, it is also applicable to agricultural and food production environments [133, 116, 53]. Section 2.1.2 provides a more in-depth look at MST methods.

The impetus behind MST is determining the total maximum daily load (TMDL)[2]

[2]TMDL or Total Maximum Daily Load is the maximum amount of a pollutant that a water body

as defined in Section 303(c) of the 1972 Clean Water Act [12, 121]. The traditional FIB mentioned above were originally used due to rapid, inexpensive, and easy detection in order to help legislators and resource managers establish the TMDL for a given body of water [95]. However, as the focus shifted to source tracking for better control of fecal pollution, and as newer molecular detection methods were discovered, alternative FIB such as *Bacteroides*, *Bifidobacterium*, *Rhodococcus*, and *Methanobrevibacter* were investigated for use in MST [12, 157, 126, 95].

Since human pathogens are assumed to be the greatest risk to human health, it is important to distinguish between human and animal sources of contamination [47, 134]. The primary focus of source tracking has traditionally been diseases caused by human pathogens, such as typhoid and cholera, however zoonotic diseases[3] are an increasing concern [62]. *Cryptosporidium* and *E. coli* O157:H7 are often shed in the feces of infected cattle while *Campylobacter* appears to come from poultry and the parasite *Giardia* is commonly found in multiple animal hosts [55, 62, 84]. Knowledge of source contamination is vital for effective remediation strategies. For example, typhoid has largely been eliminated in the US due to disinfection and filtration treatments and *E. coli* O157:H7 can be removed by chlorination, but the zoonoses *Cryptosporidium* and *Giardia* are resistant to standard filtration measures, and preventative rather than remediative measures are recommended [84, 133].

MST methods seek to use different characteristics of FIB to differentiate between human and animal hosts, and because of zoonotic concerns, also to differentiate between individual animal species [121, 14]. Again in plain terms, MST answers the question "Which genus (or species) is contaminating the environment?". Library-dependent MST methods involve collecting samples from known hosts and building a library of molecular signatures (also known as fingerprints). Unknown samples

can receive, and still meet water quality standards [12, 121].

[3]Zoonotic disease are diseases transmitted from animals to humans.

can then be "fingerprinted" and matched against known fingerprints in the library [126, 121, 62, 101]. Library-independent methods are newer and typically use the presence or absence of a known host-associated marker to determine contamination [121, 126, 62, 142, 157].

Dr. Michael Black and Dr. Chris Kitts, biologists at Cal Poly Center for Applications in Biotechnology (CAB), are investigating using *Bacteroides* as an FIB in a new library-dependent, culture-independent MST method. This developing method is library-based, which means that samples need to be collected and a database of known fingerprints needs to be built. However, this method is culture-independent and does not require growth and storage of bacterial cultures since only the DNA sequences are needed. *Bacteroides* has been investigated as an organism of interest in MST library-independent methods due to its high degree of host-specificity, inability to grow well in the environment, and relative abundance in fecal samples [48, 121, 109]. As an obligate anaerobe, *Bacteroides* is hard to cultivate in a laboratory setting and only with the advent of newer molecular technologies, such as Polymerase Chain Reaction (PCR), has it been looked at for MST. PCR makes numerous copies of a specific DNA region of interest through the use of **primers** which are short segments of DNA (or RNA) that act as a starting point for DNA synthesis. Further details on primers and an explanation of the PCR process are discussed in Section 2.2.2.

In 2000, Bernhard and Field [11] developed a *Bacteroides* PCR primer for 16S rRNA that distinguished between ruminant and human sources. Since that time, many more primers have been developed and tested for different MST assays. Table 1.1 lists a small sampling of the numerous *Bacteroides* primer assays that have been created for specific host targets in recent years. Each primer is used to trace a specific *Bacteroides* strain and therefore, a specific host.

Table 1.1: Selected examples of *Bacteroides* PCR primer assays developed in the past 20 years grouped by host target.

Primer Assay	Host Target	Reference
AllBac	*Bacteroides* genus	Layton et al. (2006) [82]
BacUni-UCD	*Bacteroidales* order	Kildare et al. (2007) [76]
CF193	Ruminant	Bernhard & Field (2000) [11]
HF183	Human	Bernhard & Field (2000) [11]
HF183 SYBR	Human	Seurinck et al. (2005) [135]
HuBac	Human	Layton et al. (2006) [82]
BacHum-UCD	Human	Kildare et al. (2007) [76]
Human-Bac1	Human	Okabe et al. (2007) [106]
BoBac	Cow	Layton et al. (2006) [82]
BacCow-UCD	Cow	Kildare et al. (2007) [76]
Cow-Bac1	Cow	Okabe et al. (2007) [106]
HoF597F	Horse	Dick et al. (2005) [31]
Hor-Bac	Horse	Tambalo et al. (2012) [145]
PF163F	Pig	Dick et al. (2005) [31]
Pig-Bac1	Pig	Okabe et al. (2007) [106]
Pig-Bac2	Pig	Okabe et al. (2007) [106]
Pig-1-Bac	Pig	Mieszkin et al. (2009) [98]
BacCan-UCD	Dog	Kildare et al. (2007) [76]
CB-R2-80	Chicken	Lu et al. (2007) [87]
CGOFG1-Bac	Canada Goose	Fremaux et al. (2010) [52]
Gull-2	Gull	Lu et al. (2008) [88]
MuCa01	Muskrat	Marti et al. (2011) [91]
Beapol01	Beaver	Marti et al. (2013) [92]

The current library-independent approach utilizing *Bacteroides* would require running multiple different PCR primer assays to target each individual source in an environmental sample. Even for a specific host, multiple assays may be required. One problem is that the bacterial strain targeted by the primer may not be found in a given individual host [47]. As a simple example, a human *Bacteroides* primer may only be found in 5 out of 6 humans. So that assay would exclude the human without that specific strain, but maybe a different primer entirely or a combination of two primers would cover all six humans. Both Layton et al. [83] and Shanks et al. [137] concluded that since no assay was 100% specific, multiple assays are needed for the confirmation of human fecal pollution. Another issue is misidentification and cross-reactivity of primers. In the study by Shanks et al. [136] comparing bovine markers, one bovine-specific assay had a 47.4% specificity, meaning it was not exclusive to cows and targeted other hosts. Because the goal of MST is to identify "who is pooping in the water", optimal MST methods should find all sources of contamination and identify those sources accurately.

Individual assays are not ideal as a comprehensive MST method since they would exclude many sources of contamination. Cal Poly CAB researchers hypothesize that by using a primer that targets a higher taxonomic level, a single assay can provide all the source information needed from an environmental sample. The new method will use Next-Generation Sequencing (NGS) techniques to obtain the sequence information of all organisms belonging to the phylum Bacteroidetes, and then use computational tools to cluster similar sequences at a genus level. Each cluster of related sequences is referred to as an **Operational Taxonomic Unit (OTU)**. As OTUs are a computational concept, they are not explicitly found in nature, but are constructed *in silico*. An OTU will theoretically act as a molecular signature that can be traced back to a specific host.

CAB researchers hope to create a library of OTUs that can be used for iden-

tification of the sources of fecal contamination in unknown environmental samples. The OTUs would be the "fingerprint" in this library. As envisioned, this OTU-based method would remove the need for multiple assays using multiple primers. A second advantage of the new method is its culture-independent aspect. Utilizing only DNA sequences for analysis eliminates the necessity to culture and store indicator organisms, saving both the time and financial resources required in culture-dependent methods. Finally, using OTUs as molecular signatures also has a key advantage regarding unclassified samples. Current library-independent techniques require the development of a specific primer to find a specific host. If a primer has not been found for a host, that host remains unclassified. For example, if a water sample contains feces from humans, cows, and raccoons, current primers only identify humans and cows, leaving out the unclassified raccoon source and consequently any potential pathogens carried by raccoons. The OTU-based method could hypothetically find an OTU for raccoon even though there is no current raccoon-specific primer. Therefore, this new method can potentially help identify unknown host species that have no specific *Bacteroides* primers developed, resulting in a more complete picture of fecal contamination. NGS methodology is discussed in Section 2.2.3 and detailed information about OTUs in Section 2.3.3.

As with all library-dependent methods, the new OTU-based method is contingent upon the information, the "fingerprints", stored in the library. Using the example from above, a raccoon OTU can be found only if a raccoon sample was collected and a raccoon-associated OTU was added to the database. Only sources that are in the library can be identified, so constructing a diverse library is essential to the proper function of this MST method. Though an adequate library size has not been established, Brown et al. [18] suggest that more than 12 samples per animal type group are needed in OTU-based libraries for reliable source information.

To address the biologists' needs for the computational aspects of the new OTU-

based MST method, this thesis presents LOTUS, the **L**ibrary of **OTUs**, a web-based database application that creates and stores OTUs for use in fecal source identification in environmental samples. LOTUS consists of four main components which act collectively as a support tool for the new MST method. These components are: a database (the reference "library"), a standardized pipeline for creating OTUs, a method for determining taxonomy of OTUs, and a web-based user interface.

The database component stores data regarding OTUs and any associated metadata, providing a knowledge base of *Bacteroides* OTUs present in the environment. The website interface allows researchers to easily add known samples to the OTU library or to match unknown samples to the library without requiring significant technical skills. As OTUs are abstract concepts and exist only in the database, OTUs must be created from the samples before comparison with the database. The key feature within LOTUS is a multi-step pipeline for processing OTUs from raw sequence files. This pipeline uses existing open source bioinformatics software packages to produce high quality, consistent, standardized OTUs for comparative analysis. Once created, the OTUs must be assigned a taxonomy to enable source tracking. The design of the LOTUS database and the specifics of the pipeline are discussed in Chapter 3.

This work discusses a preliminary investigation into using *Bacteroides* as an indicator species for MST. The computational tools presented in this thesis are necessary for investigators to determine if using a library of *Bacteroides* OTUs as molecular signatures will be a viable MST method. This thesis will help investigators conclude if this is possible and allow them to continue development of a faster, more cost effective MST method.

To this point, the main focus of this thesis answers the question "Can source-specific OTUs, termed **single-source OTUs**, be found that can be used as molecular signatures for source tracking?". Validating a multi-component tool such as LOTUS

requires evaluating the final output to make this determination. LOTUS has two essential pieces of functionality: 1) a reference library of OTUs, and 2) a procedure for matching an OTU from a sample from an unknown host to the reference library. The reference library itself must be created and evaluated for the presence of single-source OTUs. The quality of the OTU clusters used as the reference library was evaluated using purity and entropy[4]. The methodology that produced the highest purity and lowest entropy OTU clusters was used to create the reference library OTUs which were then used to evaluate unknown matching accuracy. The best case reference library had 1,431 OTUs produced from 1,563,411 sequences from 12 samples from known hosts. Of these, 994 OTUs were classified as single-source using 75% purity as the cutoff threshold. Five separate samples were used as the test unknowns for source tracking. Using 75% purity as a cutoff, LOTUS was able to match four of the five samples, achieving matching accuracy of 62.86% for the horse sample, 33.99% for the dog sample, and 62.86% for the two human samples, suggesting that OTU-based MST is a promising method.

The contributions of this thesis are:

- A database design that models *Bacteroides* sequence data, OTUs, and associated metadata for storage, maintenance, and analysis to act as the library component of the MST method.
- A standardized pipeline that constructs consistent, high quality OTUs from raw Illumina reads using existing open source software packages.
- Determination of the taxonomic assignment of library OTUs for subsequent unknown matching at the genus level in order to test the use of OTUs in potential MST method.

[4]Purity and entropy are measures of the extent to which a cluster contains a single class (i.e., sequences from the same *Bacteroides* strain) [147].

- A web-based user interface tool for researchers to easily access the database, add new samples to the library, and match unknown samples with the library.

The remainder of this document is organized as follows: Chapter 2 explains the necessary biological background relevant for understanding the context of this thesis. As this work primarily functions as support tools for use by CAB biologists, it is essential to understand the biological underpinnings before discussion of computational solutions. Chapter 3 describes the relational database design necessary for source tracking as well as the pipeline to construct OTUs and user requirements for LOTUS as a web-based application. Chapter 4 discusses the implementation of the different components of LOTUS. Chapter 5 shows the validation and evaluation of LOTUS as a tool for creating OTUs for further investigation. Finally, Chapter 6 concludes this thesis with a summary of the work so far and ideas for future work.

Chapter 2

BACKGROUND & RELATED WORK

This thesis discusses a computational support tool used in the development and research of a new MST method. To understand the role and context of LOTUS, it is necessary to understand the biological half of the MST method and the underlying biological concepts that are represented by the data. This chapter explains both the biological and computational basis for the OTU-Based MST Method which will henceforth be referred to as OBMM.

2.1 Biological Background

2.1.1 Biology Concepts and Terminology

A review of molecular biology is helpful to understand the role of the support tools discussed in this thesis and the terminology used throughout this document. This summary section is based on information found in Chapter 1 of *Next Generation Sequencing Technologies and Challenges in Sequence Assembly* [42].

Deoxyribonucleic acid (DNA) is a double-stranded molecule that contains the unique genetic information of every living creature. This genetic code is the cellular blueprint used for gene expression. DNA is composed of 4 nucleotides: Adenine (A), Cytosine (C), Thymine (T), and Guanine (G). Adenine and Guanine are purines while Cytosine and Thymine are pyrimidines. These nucleotides (nt) can bind to each other and form complementary base pairs. These base pairs (bp) consist of a purine bound to a pyrimidine and bind as follows: $\mathbf{A} \longleftrightarrow \mathbf{T}$ and $\mathbf{C} \longleftrightarrow \mathbf{G}$. The order of nucleotides is called a **sequence**. Within sequences are specific regions called genes which most often encode for proteins. Genes can be hundreds or thousands of

base pairs long.

A single strand of DNA sequence can be represented as a string of letters consisting of the nucleotide alphabet. DNA is read from the 5′ end to the 3′ end[1]. Ribonucleic Acid (RNA) is a nucleic acid that is used to build proteins. RNA is similar to DNA except that RNA is single-stranded and uses Uracil (U) in place of Thymine (T). RNA is replicated from DNA in a process called transcription. During transcription, an enzyme called RNA polymerase travels along the DNA strand and builds a complementary RNA strand. Figure 2.1 shows an example of a DNA template sequence and the complementary strands that can be built from it.

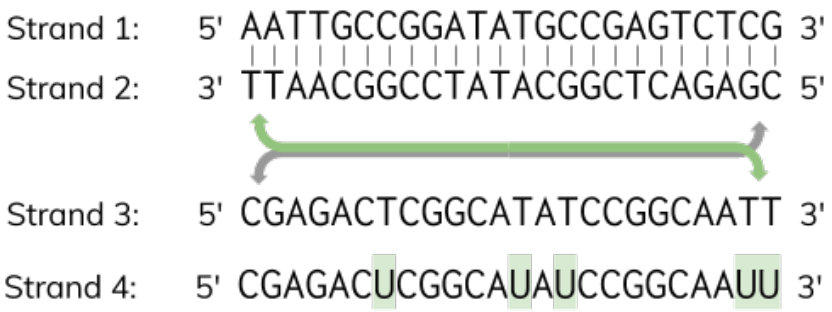


Figure 2.1: Visualization of complementary base pairs of a 24 bp sequence. Strand 1 is the original (template) strand read from 5′ end to 3′ end. Strand 2, the paired strand, shows how the nucleotides line up when they bind (or pair). Strand 3, the complementary strand, is the paired strand read in the proper 5′ to 3′ orientation. Strand 4 is the complementary RNA strand with U in place of T.

While there are different types of RNA in the cell, molecular studies often use ribosomal RNA (rRNA). A ribosome is an intracellular structure necessary for protein synthesis. The ribosome is made of two subunits of rRNA. In bacteria, the **16S rRNA gene** is a major part of one of these subunits, called the small subunit [146]. Bacterial studies often use the 16S rRNA gene (sometimes called the SSU rRNA gene) as a target sequence for genetic research because it is highly conserved and yet has regions of variability [95, 105, 107]. The highly conserved regions are present in all bacteria

[1]The 5′ end is pronounced "five prime end" and refers to the sugar backbone of DNA.

making it a useful target for PCR amplification, while the nine hypervariable regions (e.g., V2, V3, V4) can be used to differentiate between organisms [105, 45, 51]. The 16S rRNA gene is around 1500 bp long [115, 51]. RNA can always be determined from DNA due to complementary base pairing, so biologists using 16S rRNA genes refer only to the four nucleotides found in DNA. This chapter explains how biologists use this sequence information to develop an OTU-Based MST Method.

2.1.2 Microbial Source Tracking (MST)

Microbial Source Tracking (MST) is a discipline in Biology that attempts to identify the source of fecal contamination in bodies of water, particularly water used for human consumption or recreation [12, 62, 121, 134]. A major branch of MST focuses on tracking fecal indicator bacteria (FIB). This particular field of investigation is based on the premise that specific strains of microorganisms are associated with specific hosts [60, 62]. As this is an ever evolving area of research, there are currently many methods available. MST methods can be broadly divided into two main categories: Library-Dependent and Library-Independent [12, 60, 121, 126, 142].

Library-Dependent Methods

Library-dependent methods, as the name implies, require the construction of a library of characteristics (or "fingerprints") of fecal isolates from known samples which can then be used for comparison with unknown isolates [121, 101, 126]. The terms "known" and "unknown" as used here refer to sample provenance information, so known samples are samples taken from individual hosts of specific known species while unknown samples are taken from environmental sources without a specific known species. Library-dependent methods can use either phenotypic (biochemical) or genotypic (molecular) techniques [121, 126, 12, 60]. Phenotypic methods use observable

characteristics of a microorganism such as biochemical properties like outer membrane proteins or serology. These methods include Antibiotic Resistance Analysis (ARA), Carbon Utilization Profile, and Nutrient Utilization Pattern [121, 126, 12, 142]. Genotypic methods use the genetic information of a microorganism and include Ribotyping, Pulse-Field Gel Electrophoresis (PFGE), and Repetitive Palindromic Polymerase Chain Reactions (rep-PCR) [121, 126, 12, 142]. To be useful for fecal identification, the library has to contain a sufficiently large collection of known samples. For *E. coli*, this number is suggested to be between 900 – 2000 isolates [126]. All library-dependent methods are reliant on the size and composition of the library. This limitation is due to the fact that only the host species in the library can be used for identification. The host range is determined by the samples collected from known sources during construction of the library [62]. This has the effect that libraries are more focused on local areas where the known samples were collected and are potentially not as viable in other geographic locations [126, 12, 121, 148]. Another consideration for library-dependent methods is the stability of the fingerprints in the library over time [101, 148].

Library-Independent Methods

Library-independent methods have been developed more recently in an attempt to reduce the reliance on a library in favor of directly finding the marker of interest. These methods utilize a unique genetic marker from a host-specific organism [62]. There is no need for a library since this marker is only ever found in a specific microorganism and that microorganism itself is specific to one host species [121, 126, 157, 12, 142]. Library-independent methods include bacteriophages, bacterial PCR, F^+ RNA coliphage, and viral markers [126, 12, 62]. In this way, only the presence or absence of the marker is needed to determine the source of fecal contamination [142]. As library-independent methods each focus on a single species, they rely on a

separate method for each host and lack the ability to identify multiple sources in an unknown sample [62].

Culture-Dependent and Culture-Independent Methods

Both library-dependent and library-independent MST methods can be further subdivided into culture-dependent or culture-independent [47, 121, 157]. Culture-dependent methods such as ARA grow bacterial cultures on agar plates in the laboratory. Culturing allows researchers to "isolate" or grow only the bacteria of interest from all the other microorganisms present in a sample. Cultures can take 1 – 2 days for the bacterial colonies to grow costing researchers time as well as financial resources for labor and supplies such as growth plates and incubators [47]. Newer molecular technologies allow for culture-independent methods such as bacterial or viral PCR [62, 121]. These methods target genetic information, so for a given sample, researchers can look for a sequence using genetic markers rather than culturing to find if the bacteria is present.

2.1.3 Related Work

Pyroprinting

Dr. Black and Dr. Kitts, in collaboration with the Computer Science department, previously developed a novel library-dependent MST method called pyroprinting [13]. Pyroprinting uses **strains**[2] of *E. coli* to identify sources of fecal contamination. The biologists chose *E. coli* as it is used by most regulatory agencies as an indicator of fecal contamination. As with any library-dependent method, pyroprinting required a large collection of bacterial samples from known host species to create the library. After each fecal sample was collected, isolated colonies (called **isolates**) were cultured in the lab. Polymerase Chain Reaction (PCR) was run on the isolates to amplify the

[2] A *strain* is defined as a group of isolates that have similar DNA fingerprints [100].

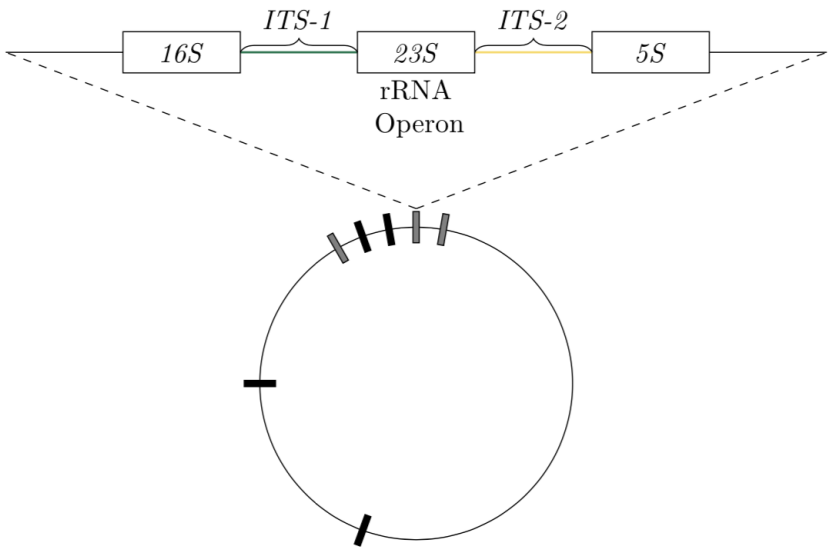


Figure 2.2: A diagram of the *E. coli* genome, showing the seven copies of the rRNA operon. Each operon contains the two ITS regions: *ITS-1* (between the 16S and 23S genes) and *ITS-2* (between the 23S and 5S genes).

DNA sequences of interest. For *E. coli*, the DNA sequences of interest were two **intergenic transcribed spacer (ITS)** regions in the rRNA operon[3]. The *ITS* regions are non-coding regions of the genome between two genes as seen in Figure 2.2. The *E. coli* genome contains seven copies of the rRNA operon containing both regions *ITS-1* and *ITS-2*. One PCR primer will amplify all seven rRNA operons, giving seven potentially different templates for each *ITS* region [100, 141, 94, 81].

After PCR, this mixed template DNA was run through the Pyromark machine to generate a pyroprint for each *ITS* region. It is important to note that a pyroprint does not give the exact DNA sequence of the *ITS* region; but rather represents an aggregate of all seven template DNA strands, providing as Lai [81] says, a "unique identifier for that region of a species that can later be used for strain discrimination". McGovern [94] defines a pyroprint as "a vector representing the peak light values of the pyrosequencing of one of the *ITS* regions in the seven loci of the *E. coli*

[3]An operon is a group of related genes that are transcribed together to produce a single mRNA [43].

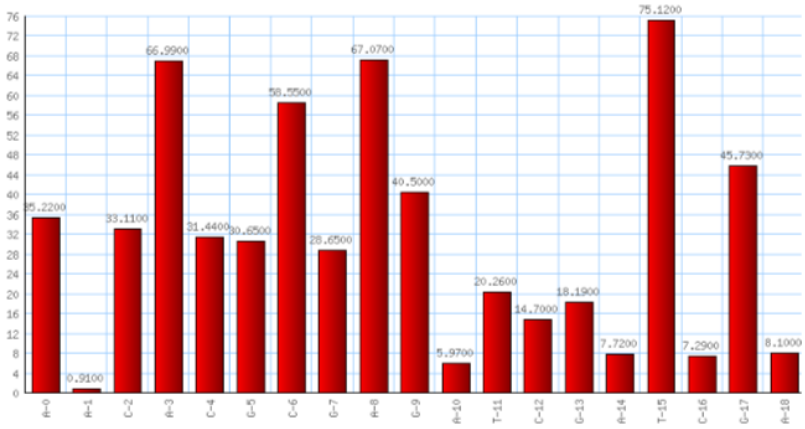


Figure 2.3: Graphical example of a pyroprint. A pyroprint is a vector of peak heights. Peak heights represent the light intensity emitted during each nucleotide dispensation of the pyrosequencing process.

genome." A discussion of pyrosequencing is outside the scope of this thesis, but the peak light values are the light intensities released as a nucleotide is dispensed during the sequencing process. The pyroprint itself is a vector of numerical values, but a visualization of a pyroprint is shown in Figure 2.3. Variations of nucleotides in the template strands will result in differences in peak height intensities at different nucleotide dispensations, resulting in a unique set of values that create a distinct pyroprint.

Finally, the pyroprint was stored in a database along with the metadata that included host species from which the isolates were collected. The reader is referred to Black et al. [13], Montana [100], and Soliman [141] for a detailed discussion of the pyroprinting process.

The pyroprinting method requires the use of a library. Students and faculty of the Cal Poly Computer Science Department implemented the Cal Poly Library Of Pyroprints (**CPLOP**) to provide storage and analytical support for pyroprinting [140, 141]. CPLOP provides computational support for the use of pyroprinting as an MST method. The CPLOP database was designed to store pyroprints as well

as metadata which provides contextual information about a given pyroprint such as isolate, host, host species, and sample [140]. Similarity between pyroprints is measured using Pearson correlation [141]. CPLOP also provides functionality to cluster isolates into strains based on similarity using OHClust! [100], DBSCAN [73], k-RAP [94], and HAP [81] algorithms. Pyroprints from unknown isolates can in this way be clustered into strains which are then associated with a specific host species [100].

As CPLOP was used and amassed more data, certain issues came to light. First, pyroprinting is a culture-dependent MST method and so is subject to the time constraints and financial limitations of this dependency. Second, pyroprinting (along with other MST methods) is assumed to work based on the idea that certain strains are host-specific [121, 126]. As more data was collected in CPLOP, more variation was found, indicating that strains of *E. coli* seemed to be very transient. These findings suggested that *E. coli* traveled in and out of different hosts, which made tracing a strain back to a specific host species difficult in some instances. As an example, bat strains were very specific, but strains that traced back to chickens also traced back to other host species [94, 81].

Bacteroides

The transient nature of strains of *E. coli* led to the investigation of a different FIB. The genus *Bacteroides* has been looked at as an alternative indicator organism for use in library-independent, culture-independent MST methods [12, 62, 121, 126, 142, 157]. It is an obligate anaerobe, meaning it cannot grow in the presence of oxygen. This property makes *Bacteroides* difficult to cultivate in laboratory settings, thus resulting in a need to use genotypic techniques that do not require culture for MST methods. As an anaerobe, *Bacteroides* is not generally found in the environment, but is abundant

in mammalian gastrointestinal (GI) tracts [121], so its presence in environmental samples is indicative of recent fecal pollution [109]. Wuertz et al. [157] show that the human intestinal bacterial population comes from only nine different phyla; and of those, the two prominent pyhla Firmicutes and Bacteroidetes make up 98% of that population. This composition means that *Bacteroides* species are present in fecal samples at much larger concentrations than other microorganisms. For example, *Bacteroides* is present at 2–3 orders of magnitude higher concentration than *E. coli* in mammalian feces [157]. This abundance gives *Bacteroides* a lower detection threshold than other FIB. Another feature that makes *Bacteroides* useful for MST is the high level of host-specificity [62, 121, 126, 157]. This quality means that one species of *Bacteroides* is generally found in only one host species, in contrast to the overlapping or mixed results that can happen at the strain level of *E. coli*.

Bacteroides primers have been developed for specific species as previously presented in Table 1.1. Primers are short segments of DNA used in PCR and described in Section 2.2.2. Various studies have evaluated these types of primer assays in different ways. Some studies compared markers against others that target the same host to evaluate performance and efficiency [137, 136, 83]. Other studies looked at the performance of primer assays in different geographical locations [109, 114, 117, 9, 77]. Finally, some studies tested the usefulness of these assays as a source tracking tool using different fecal source types such as fresh feces, sewage, marine water, fresh water, and stormwater [127, 137, 114, 83, 77]. Numerous and continuing studies underscore the point that there is still no one MST method that is preferred over the others and no single method can determine all sources of fecal contamination [12, 121, 126].

2.2 Cal Poly OTU-Based MST Method (OBMM)

The new OBMM is based on *Bacteroides* primer assays developed for library-independent techniques. By using primers targeting a higher taxonomic level, biologists can gather information about multiple species in a single assay and then theoretically can computationally cluster related sequence information back into genus level groups (OTUs). The primer is no longer host-specific, but the OTUs can be used as molecular signatures to identify hosts. The OTUs require a library for comparison since these clusters only exist in the computer. Though this method is library-dependent, the underlying primer assays that generated the information are geographically stable (i.e., these assays work in different geographic locations) [47], so the library should be effective outside the local environment.

Dr. Black and Dr. Kitts have requested a tool similar to CPLOP that will allow them to investigate using OTUs constructed from *Bacteroides* for MST. This section discusses the four steps in the workflow of their new OBMM: 1) data collection, 2) PCR, 3) Next-Generation Sequencing (NGS) with the Illumina Miniseq platform, and 4) data processing. Steps 1 – 3 are the biological component of the OBMM and step 4 is the computational component.

2.2.1 Data Collection

The first step in any MST method is data collection. For the new OBMM, fecal samples are first collected from known host sources to build the library. Each host is given a unique identifier and may contribute multiple samples (e.g., at different times). Biologists collect fecal material from different animal hosts and record contextual information about each host such as species classification, date of sampling, and geographic location including latitude and longitude. This contextual information (or

metadata) is recorded by investigators during sample collection. The metadata allows researchers to understand the *Bacteroides* microbiome in the sampled environment which provides a foundation for further avenues of study. Once a library has been constructed from a sufficient number of samples, biologists can collect environmental water samples and record associated metadata to test for unknown sources.

2.2.2 Polymerase Chain Reaction (PCR)

After sample collection, the next step is to use an *in vitro* process called Polymerase Chain Reaction (PCR) to amplify a target gene sequence, in this study the 16S rRNA gene [121, 126, 142]. PCR utilizes **primers** which are short sequences of nucleotides that bind to a target region of a DNA sequence. Primers act as a starting point for synthesis of the complementary strand [45]. There are two primers: a forward primer and a reverse primer. Figure 2.4 shows an overview of the steps in the PCR process.

Biologists design the PCR primers to detect the target sequence of interest. For the OBMM, the primers target sequences common to the Bacteroidetes phylum. PCR targets a portion of the gene sequence using primers and then creates multiple copies of that sequence, a process known as amplification [97]. During amplification, the three-step process is repeated in successive iterations as shown in Figure 2.5. PCR enables MST methods to lower the detection threshold for the gene of interest since it allows even small amounts of target gene to be found and amplified. The sequence copies that result from PCR are termed **amplicons**.

2.2.3 Next-Generation Sequencing (NGS)

After the PCR step, the amplicons need to be "sequenced" for downstream analysis. DNA sequencing is a process that allows scientists to define the order of the nucleotides in a DNA molecule [59, 80]. Sanger sequencing, first introduced in 1977,

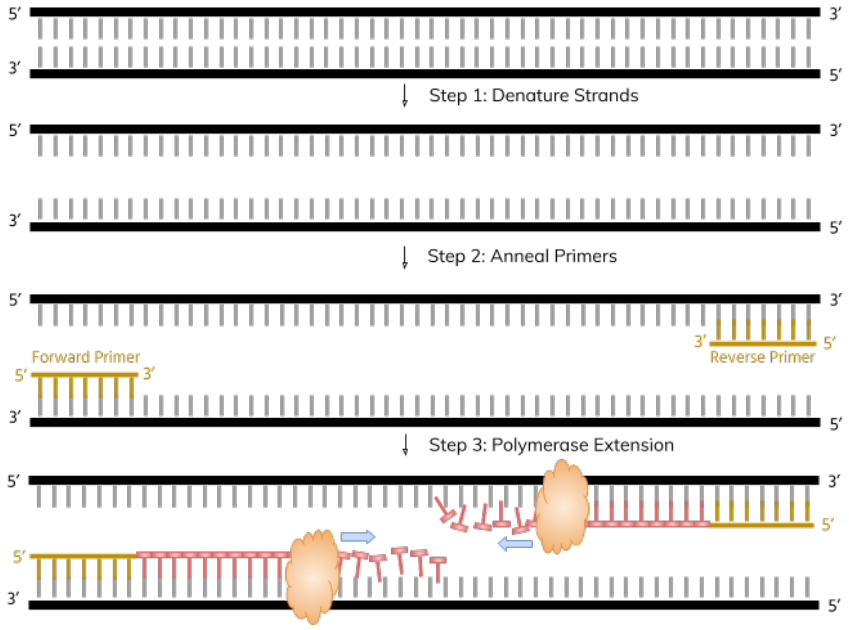


Figure 2.4: An overview of the three steps in a single PCR cycle. Step 1: The two strands of DNA are separated (denatured). Step 2: the primers are attached (annealed) to the target region. The forward primer attaches to the bottom strand and the reverse primer attaches to the top strand. Step 3: Polymerase enzyme starts at the primer and synthesizes a new complementary strand of DNA from 5′ end to 3′ end direction.

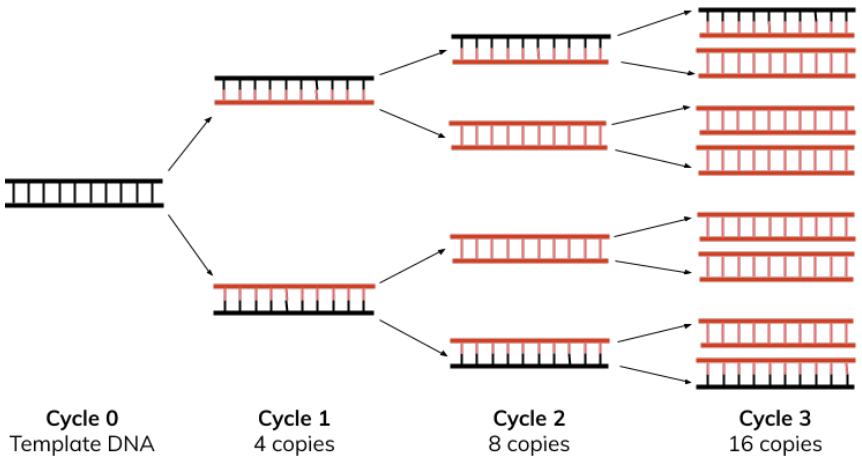


Figure 2.5: Process of PCR Amplification. Black strands represent the original DNA template strands. Red strands are the copies produced during amplification (amplicons). A single strand is doubled every cycle. After 25 cycles, there are $2^{25} = 33{,}554{,}432$ amplicons from a single strand.

is able to determine the nucleotide order for one DNA fragment at a time [125]. This first-generation sequencing method has been used for decades in studies such as the Human Genome Project [80]. **Next-Generation Sequencing (NGS)** techniques have been developed since 2005 to allow massively parallel sequencing of millions of fragments at one time [69]. To put this in perspective, the first human genome took 15 years and $3 billion dollars to sequence, but with NGS, over 45 human genomes can be run in one day at a cost of about $1,000 per genome [59, 69].

There are several different NGS platforms including the Roche 454 GS FLX+, Ion Torrent Personal Genome Machine (PGM), Illumina MiSeq, Pacific Biosciences (PacBio) Single Molecule Real-Time (SMRT), and Oxford Nanopore GridION [112, 5, 107]. Each platform has its particular advantages and disadvantages [118]. All sequencing platforms "read" a DNA fragment from one end to the other. The output sequences are thus referred to as **reads**.

Read length is an important factor in microbial studies. First-generation Sanger sequencing produced average reads of 500 – 1,000 bp, but NGS typically produces short reads (100 – 300 bp) [71, 99]. Short reads give less information so are a limitation in studies looking at genes or whole genomes which can be thousands or millions of base pairs long [99]. The 16S rRNA gene is 1500 bp and the 300 bp short reads produced by NGS only cover one or two hypervariable regions [102]. The newest third-generation NGS technologies are capable of producing long reads but are not widely adopted due to higher cost and error rates. These include the PacBio SMRT (10,000 bp reads) and Oxford Nanopore (50,000 bp reads) [80, 39, 107, 118]. Despite producing short reads, Illumina systems (100 – 300 bp) have replaced 454 pyrosequencing systems (up to 1,000 bp) due mainly to increased throughput, high quality reads, and relative low cost [107, 146, 5, 148, 151]. As of 2016, Roche discontinued the 454 platform, although it is still in use in many laboratories [5, 102].

Of the currently available NGS technologies, Illumina systems are the platform of choice for most metagenomic studies with an estimated 70% of the market [28, 102, 107, 71, 80]. **Metagenomics**[4], more accurately "shotgun metagenomics", is the genetic analysis of all microbes (including fungi and viruses) in an environment without the need for culturing [61, 107]. The field of metagenomics has expanded to include targeted 16S rRNA gene studies in what is called "marker gene amplification metagenomics" or 16S metagenomics [107, 86]. Compared to shotgun metagenomics, 16S studies are a rapid and affordable way to characterize bacterial diversity in an environmental sample. Illumina offers a variety of platforms optimized for shotgun or targeted sequencing studies [118, 69].

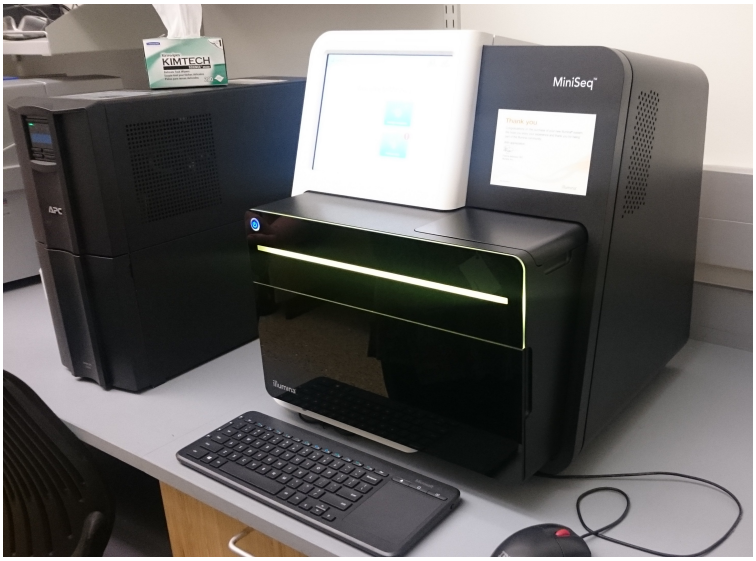
The choice of NGS platform along with NGS sample preparation determines read length, run time, error profile, and read quality which effects the processing of the output data [5]. Cal Poly has invested in a benchtop sequencing platform called the Illumina MiniSeq as seen in Figure 2.6. The Illumina MiniSeq produces high quality short reads up to 150 bp for single-end sequencing or 75 bp for paired-end sequencing with a run time from 7 – 24 hours [66]. This system is designed to support targeted sequencing studies (i.e., 16S studies) and can output 1.8 – 7.5 Gigabytes (Gb) of data per run [69].

A full discussion of the Sequencing By Synthesis process used by Illumina is outside the scope of this paper and can be explored further in Illumina's documentation [69] and website[5]. Nevertheless, it is important to understand certain aspects of the process in order to interpret the output data correctly.

Illumina's main advantage is the ability to produce vast amounts of sequence data from a given sample at a reasonable price [107]. This capability is known as

[4]Metagenomics is also called environmental genomics, community genomics, or population genomics.

[5]`https://www.illumina.com/science/technology/next-generation-sequencing/sequencing-technology.html`



(a) Illumina MiniSeq™ benchtop sequencer.

(b) Reagent cartridge for samples.

Figure 2.6: The Illumina MiniSeq system setup for Next-Generation Sequencing at Cal Poly.

High-Throughput Sequencing (HTS), and it enables Illumina to overcome the limitations of short reads. For comparison, the Illumina MiSeq produces 25 million reads of 300 bp in length where 454 pyrosequencing produces 1 million reads of 1,000 bp in a single sequencing run [107, 151]. HTS is achieved in Illumina's workflow through the use of paired-end sequencing and multiplexing. **Paired-end (PE) sequencing** is a technique that sequences both ends of a DNA fragment [69]. PE sequencing produces twice as many reads as single-end (SE) sequencing and can be merged to create longer reads. The technique is shown in Figure 2.7 [40, 69].

Multiplexing enables Illumina to greatly increase throughput by allowing pooled samples to be run simultaneously. During sample preparation, unique index sequences (**barcodes**) are added to each DNA fragment so that each read can be mapped to a specific sample [69]. This saves time and reduces costs by allowing more samples to be sequenced in a single run using the same preparation reagents. After sequencing, the reads must be **demultiplexed**, a computational process that associates each read back to its sample of origin [107, 58]. Multiplexing and demultiplexing are

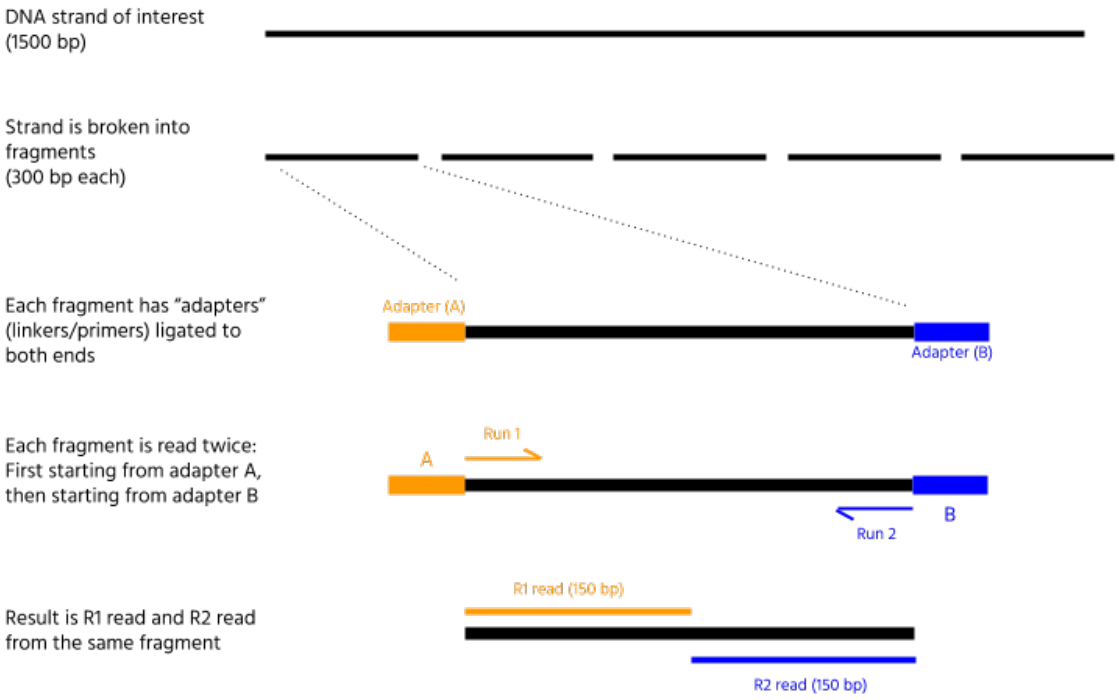


Figure 2.7: Illumina Paired-End Sequencing.

illustrated in Figure 2.8. An example read produced by Illumina with the barcode from multiplexing and the adapter from PE sequencing is shown in Figure 2.9. The use of PE sequencing and multiplexing greatly increases throughput, though it does add some computational complexity during data analysis.

2.2.4 Data Processing

Illumina HTS platforms produce extremely large datasets. Targeted 16S metagenomic studies produce millions of amplicons which translates to Gb of data [105, 86]. Bioinformatics is a field that utilizes computer resources for evaluation of biological data. In NGS applications, bioinformatics software is a necessary component to transform the overwhelming quantity of raw data into usable information for meaningful analysis. The data processing portion of the OBMM is discussed further in the next section.

Figure 2.8: Multiplexing/Demultiplexing. The multiplexing process involves the addition of unique barcodes to each sample during NGS preparation. This allows a sequencer to run multiple (pooled) samples in the same sequencing run. The output data contains reads from all samples. The reads are mapped back to their original samples in a process called demultiplexing. For simplicity, adapters are not included in this example.

Figure 2.9: Depiction of a single read produced by Illumina.

2.3 Computational Background

The vast amount of data produced by NGS platforms has necessitated bioinformatic software solutions that reduce the complexity of the raw data to enable interpretation and insights into microbial composition and function [86]. A common approach to reducing the size of NGS data in 16S metagenomic studies is to cluster or bin the data into groups called **Operational Taxonomic Units (OTUs)** [79]. Collapsing reads into clusters simplifies the analysis and transforms the data into a more manageable form for computational resources [70]. This section highlights the data processing considerations in transforming raw Illumina output data into OTUs. OTUs are discussed in detail in Section 2.3.3.

2.3.1 Raw Sequence Data

Illumina platforms output the reads to base call (bcl) files which are converted to a standardized file format called a `fastq` file. Fastq files are text files that contain both the actual sequence information and the corresponding quality information for each nucleotide of the sequence. Each `fastq` entry is composed of four lines [39]:

Line 1: Illumina sequence identifier (Begins with "`@`")

Line 2: The actual nucleotide sequence (includes barcodes and primers)

Line 3: Separator ("`+`") with optional identifier

Line 4: Quality scores in Phred-33 ASCII encoding

An example `fastq` output file is shown in Figure 2.10. The file, as shown, requires further processing before a human researcher can draw meaningful conclusions from the data.

```
@M01522:151:000000000-B9DJG:1:2110:7582:6405 1:N:0:GTTTCG
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTT
AGGCACGTGTGCCTTTTTGTATGTACCGTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGA
TCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAA
CCGTAAAATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTA
GATATCACGAAGAACTCCGATTG
+
CCCCCGG>FGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGCFGJJJJJJJJJJJJJJJJJJJJJJJ@JJJJJJJJJJJJJJ
JJ?>JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGGGFGGGGGGGGGGGGGGGGGFCCGGGGGGGGGGGGGGGCCCCC
@M01522:151:000000000-B9DJG:1:1118:15273:9358 1:N:0:GTTTCG
CTCTAGTCCTGAACCAGCCAAGTAGCGTGAAGGATGAAGGCTCTATGGGTCGTAAACTTCTTTTATATGGGAATAAAGTT
TTCCACGTGTGGAATTTTGTATGTACCATATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGA
TCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGTGGATTGTTAAGTCAGTTGTGAAAGTTTGCGGCTCAA
CCGTAAAATTGCAGTTGAAACTGGCAGTCTTGAGTACAGTAGAGGTGGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTA
GATATCACGAAGAACTCCGAT
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGJ;@JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ>JJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGGGGGFGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGCCCCC
@M01522:151:000000000-B9DJG:1:1104:28363:12846 1:N:0:GTTTCG
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATATGGGAATAAAGTT
GTCCACGTGTGGATTTTTGTATGTACCATATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGA
TCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATTGTTAAGTCAGTTGTGAAAGTTTGCGGCTCAA
CCGTAAAATTGCAGTTGATACTGGCAGTCTTGAGTGCAGTAGAGGTGGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTA
GATATCACGAAGAACTCCGATTG
+
<CCCCGFEFGGGFGGGGGGGGGEFGGGGEDGGGGG?<EFFG@GGJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJ=JJ=JJJJJJJJJJJJ=JJJJJJJJJJJJJJJGGEF:@9GGGFECF8GGGG
EFECGGGGGGFEFC:GF<CCCCC
@M01522:151:000000000-B9DJG:1:2104:12501:20661 1:N:0:GTTTCG
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATAAAGGAATAAAGTC
GGGTATGCATACCCGTTTGCATGTACTTTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGA
TCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGATGGATGTTTAAGTCAGTTGTGAAAGTTTGCGGCTCAA
CCGTAAAATTGCAGTTGATACTGGATATCTTGAGTGCAGTTGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTA
GATATCACGAAGAACTCCGATTG
+
CCCCCGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGFGGGGGFJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ<JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGGGGGGGGGGGGGGGGGGGGGGG
GGGGGGGGGGGGGGGGGGCCCCC
@M01522:151:000000000-B9DJG:1:1108:6117:4024 1:N:0:GTTTCG
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTT
AGGCACGTGTGCCTTTTTGTATGTACCGTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGA
TCCGAGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAA
CCGTAAAATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTA
GATATCACGAAGAACTCCGATTG
+
CCCCCGGGGGGFGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGGFGDFGFGGGFGGGGGGGFG
GCGGGGGGGGGGGGGGGGCCCCC
```

Figure 2.10: Example `fastq` file output by Illumina MiSeq showing first five entries. Each entry consists of four lines. The line numbers are not part of the file and are shown for reference only.

2.3.2 Corrected Sequence Data

The raw sequence data in `fastq` files represents the reads produced *by the sequencing process*, not the actual true biological sequences of interest. There are several quality filtering steps that reads must undergo before the underlying data more correctly represents the biological reality and can be clustered into OTUs.

Each step of sample processing from collection through DNA sequencing has the potential to introduce errors (or bias) which will be carried forward through the ensuing analysis [45, 129]. Errors create inflated estimates of diversity in 16S studies, meaning it appears there are more species than are actually there [15]. This inflation is of particular importance in the "rare biosphere" with new or rare species [65]. Artificial errors can create the impression of a novel species or find a species that is not actually present. For the OBMM, errors can result in false identification of a host species that is not in actuality a source of environmental fecal contamination.

PCR Errors

Errors can occur at any stage of sample processing, but PCR and sequencing errors are two well-known sources of bias for which software solutions have been developed [129, 65, 111, 146, 107]. PCR errors include substitution errors, short reads, and chimeras [129].

Substitution errors occur when an incorrect nucleotide is incorporated during polymerase extension. These errors are usually not reproduced across the millions of fragments being amplified and result in **singletons**. Robert Edgar is an authority on OTUs having developed several bioinformatics algorithms including MUSCLE [34], USEARCH [35], and UCHIME [38] which are available on his website[6]. In the manual for USEARCH, Edgar defines a singleton as "a read with a sequence that is

[6] Robert Edgar's website: `https://drive5.com`

present exactly once" [33]. After PCR amplification and quality filtering, a singleton is either a rare sequence or a substitution error. While it is possible for a unique sequence to represent a low abundance read from a rare source, PCR polymerases are known to produce substitution errors at a rate of about 1 in every $10^5 - 10^6$ bases [26, 129]. In practical terms, amplifying a 100nt fragment to 1,000 copies gives 100,000 (10^5) base pairs. Out of the 1,000 copies, the polymerase errors in 1 of those 100,000 base pairs, meaning 1 out of the 1,000 reads is a bad read. The other 999 reads are all identical and the unique read (the singleton) contains the error. Additionally, Illumina platforms are susceptible to substitution errors, particularly in GC rich regions [102, 118, 146]. Given the error rate of polymerases and the error profile of the sequencing platform, singletons are presumed to be errors. Edgar recommends discarding any singletons that remain after quality filtering to reduce false OTUs [33].

Short reads can be a by-product of PCR amplification and removing reads below a minimum length is recommended [33]. Another source of PCR errors is chimeras. A **chimera** is a hybrid DNA fragment that is made from two different DNA fragments [38, 102]. Chimeras result from incomplete amplification during PCR and do not represent a real sequence [58]. An example of chimera formation is shown in Figure 2.11. Undetected chimeras can make up a large portion of unique sequences and are a major cause of increased diversity since they can be interpreted as novel species [102]. Software that identifies and removes chimeras includes Bellerophon, ChimeraSlayer, Perseus, and UCHIME [129, 102, 58].

Sequencing Errors

Sequencing errors can occur during any of the sequencing steps: DNA fragmentation, multiplexing, bridge amplification, and base calling. The bridge amplification step

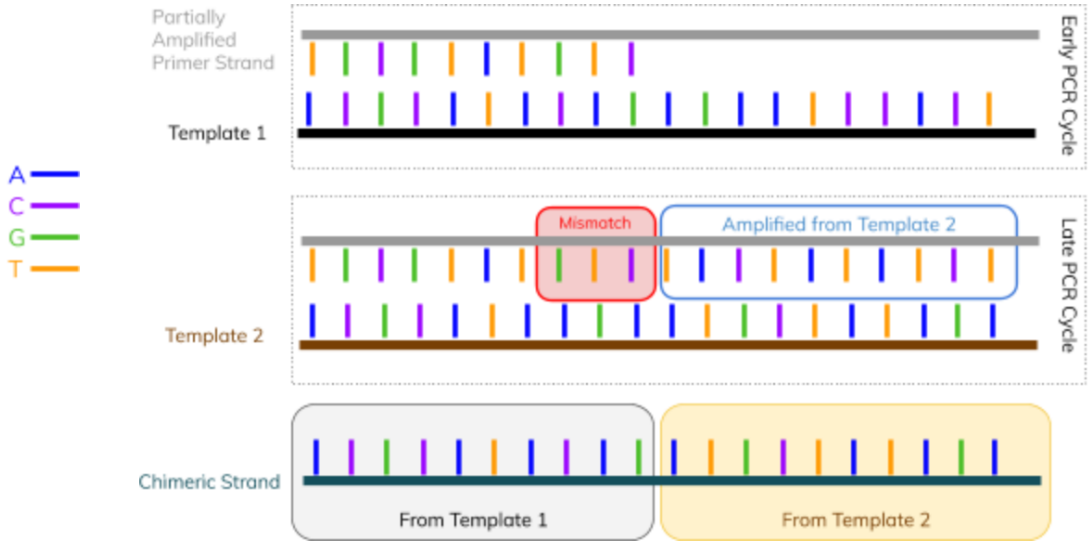


Figure 2.11: Illustrated explanation of Chimeras. This diagram shows how chimeras are formed during PCR amplification. Incomplete amplification results in a partial strand. During the next cycle, the partially amplified strand binds to a strand from a different template. The result is a chimeric strand that is made from two different DNA fragments.

uses PCR and is subject to the PCR errors described above. Sequencing errors can be mitigated by using the quality scores that are embedded in the output `fastq` data. The `fastq` files contain quality scores as a way for the sequencer to indicate the probability that the correct nucleotide base was called at the correct position. Quality scores of 30 or higher are considered the benchmark for NGS platforms [68]. Quality scores are discussed in detail in Appendix A.

Homopolymers[7] are another source of sequencing errors, though this applies to the 454 sequencing platforms [129]. Because of the ubiquitous use of the 454 platforms in earlier sequencing studies, trimming or removing reads with homopolymers remains a standard part of quality filtering.

[7]Homopolymers are stretches of a continuous single nucleotide base within a DNA fragment. For example: `AAAAAAAA` is a homopolymer of size 8.

Illumina-Specific Processing Steps

As described previously, the Illumina sequencing process includes multiplexing and PE sequencing. Multiplexed reads include unique barcodes which allow pooled samples to be run simultaneously. As shown in Figure 2.8, the reads must be demultiplexed, a process in which the barcode is mapped back to the sample identifier to associate each read with its source sample.

Paired-end sequencing is specific to Illumina platforms and produces two files: reads from the forward primer are output to the R1 fastq file and concurrent reads from the reverse primer are output to the R2 fastq file [71]. Forward (R1) reads are often higher quality than reverse (R2) reads [58, 32]. Joining the paired ends creates longer reads and allows for the identification of indels[8] which improves the overall quality of the reads [69, 33]. An example of joining paired ends is shown in Figure 2.12. The forward and reverse reads can only be joined if there is overlap between the paired sequences, a factor determined by the choice of fragment length during NGS sample preparation [40]. Although there are studies that have investigated the use of non-overlapping reads [71] or the hybrid use of both single and paired end reads [25], most applications use the joined reads if possible or the forward reads alone if not [151]. Software tools developed to join paired ends include PEAR [1], fastq-join [8], PandaSeq [93], and SeqPrep [2].

2.3.3 Operational Taxonomic Unit (OTU)

Once the reads have undergone the processing steps described above, they can be aggregated into clusters of similar sequences. It is assumed that more similar sequences are related phylogenetically [63]. Clustering the reads reduces the computer

[8]Indel is a biology term referring to either an **in**sertion or a **del**etion of nucleotides in a DNA sequence.

(a) R1 forward read

(b) R2 reverse read

(c) Joining R1 and R2 reads

Figure 2.12: Expanded example showing the use of paired-end, multiplexed reads in processing a read for sequence analysis. (a) The R1 forward read has a forward adapter. (b) The R2 reverse read has a reverse adapter. (c) If there is an overlapping region, the R1 and R2 reads can be joined to form a longer sequence. The barcodes and adapters will be removed to produce the actual sequence.

resources needed and allows investigation between different studies [63, 57, 152]. The most common approach for analyzing NGS data in 16S studies is binning the reads into computational representations of microbial taxa called Operational Taxonomic Units (OTUs) [70, 51]. A commonly accepted threshold for clustering sequences into "species" groups is 97% similarity (or 3% distance) using the sequence similarity metric [58, 105, 131, 70, 115, 152]. Sequence similarity (or percent identity) is a quantitative metric used for the comparison of two aligned DNA sequences. Sequences must be aligned before they can be compared, a non-trivial process that is computationally expensive [79]. Sequence alignment is detailed in Appendix B. The actual percent identity calculation is discussed in Appendix C. Figure 2.13 illustrates the concept of an OTU.

For the OBMM, OTUs are clustered into groups of closely related sequences at the species level, meaning that ideally an OTU should consist of sequences that come from the same host species. Such an OTU is referred to in this thesis as a "single-

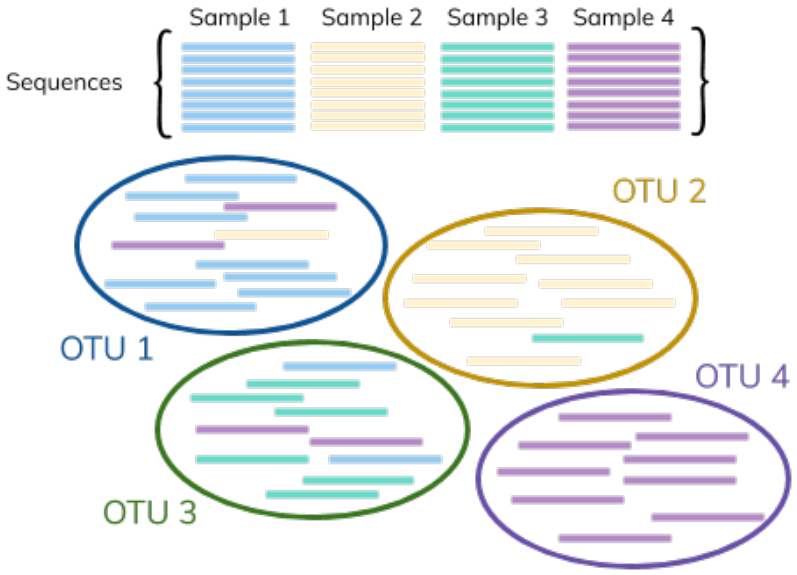


Figure 2.13: Conceptualization of Operational Taxonomic Unit (OTU). Related sequences from each sample are clustered into OTUs at 97% similarity threshold. OTU 4 depicts a single source OTU, while OTU 3 exemplifies a multi-source OTU.

source" OTU. In this way, Cal Poly biologists hope to use an OTU as a molecular signature that can be traced to a specific species for MST.

The use of OTUs is standard practice in metagenomic analysis and many OTU clustering algorithms have been developed over the past two decades. Creating clusters of OTUs is also called OTU picking [19]. The choice of OTU picking method can have a significant impact on downstream analysis and estimates of diversity [58, 143]. There are three main approaches to OTU picking: 1) *de novo*, 2) closed-reference, and 3) open-reference [119, 152, 70, 58, 110, 79, 103].

De Novo Picking

In *de novo* OTU picking, the dataset sequences are clustered based on similarity to each other without the use of an external reference database [79, 119]. Sequences within a certain similarity threshold are clustered into the same OTU. *De novo* algorithms can be further subdivided into agglomerative hierarchical clustering and greedy heuristic (centroid-based) clustering [143].

The hierarchical clustering algorithms include single-linkage, average-linkage, and complete-linkage. Hierarchical clustering methods utilize pairwise distance between sequences. **Single-linkage** clustering (also called nearest neighbor) requires that each sequence be within a pairwise distance threshold of one other sequence in the OTU cluster [63, 79, 19, 152]. **Complete-linkage** (also called furthest neighbor) requires every sequence in an OTU cluster to be within a pairwise distance threshold of every other sequence in the cluster [63, 19, 152]. **Average-linkage** requires the average of the pairwise distances for all sequences in one OTU cluster to be within a given distance threshold [79, 103]. Schloss and Westcott found that the average-linkage method produces higher quality OTU clusters over the other hierarchical clustering methods as measured by Matthew's correlation coefficient [131].

Software tools such as DOTUR [130], mothur [132], and ESPRIT [144] use hierarchical OTU clustering. The advantage of hierarchical clustering is more accurate OTU clusters. The main disadvantage is that it requires the calculation of a pairwise distance matrix between all sequences [103]. This takes an $O(n^2)$ space and time complexity where n is the number of input sequences. According to Sun et al. [143], 1 million reads produces a 7500 GB matrix which is too large to fit into the memory of most computers. Even a sparse matrix down to hundreds of GB still will not fit in memory. As a result, hierarchical clustering methods do not scale to large datasets, although improvements such as HCluster inside ESPRIT and OptiClust within mothur have been developed to overcome this limitation [143, 153]. DOTUR was rolled into mothur (available at `https://mothur.org/`) which has been maintained and improved since it was first released in 2009 [128].

A more computationally efficient *de novo* alternative is heuristic **greedy centroid-based** clustering [63]. Heuristic algorithms speed up performance but do not guarantee optimal clusters. For greedy clustering, sequences are evaluated one at a time, negating the need for a large pairwise distance matrix. The first sequence is designated as a centroid of an OTU and subsequent sequences are clustered into that OTU if they are above a similarity threshold to the centroid. If not, the new sequence becomes a new centroid and the process repeats, comparing subsequent sequences to each of the centroids [33]. If the new sequence is within the threshold to multiple centroids, it can be clustered either with the most abundant centroid or with the closest distance centroid. The computational complexity of greedy clustering is $O(mn)$ where m is the number of centroids and n is the number of input sequences. Since $m << n$, greedy clustering is far more computationally efficient than hierarchical clustering [143]. One *caveat* is that greedy clustering is highly dependent on the order of the sequences being processed, since the first sequence to be processed becomes a centroid. He et al. [63] found that abundance-based greedy clustering produced

more stable OTUs than other *de novo* methods.

The most well-known greedy centroid-based algorithm is UCLUST [35] developed by Robert Edgar in 2010. UCLUST uses USEARCH to compare new sequences to centroids. USEARCH is a heuristic that uses overlapping k-mers to identify a small number of centroids which are closest to the new sequence hence decreasing the size of m even further [35, 33]. This technique greatly speeds the performance of UCLUST while maintaining sensitivity according to Edgar [35]. The USEARCH software tool is available from `www.drive5.com`, but it is commercial and closed source.

Other greedy algorithms are variations of UCLUST. CD-HIT [85] uses the same centroid-based approach, but does not have the fast heuristic employed by USEARCH. SUMACLUST [96] uses exact alignment at each step which provides accurate clusters, but is very slow. VSEARCH [123] uses the same techniques as described in USEARCH and was developed as an open source alternative to USEARCH.

OTU picking is an ongoing area of research with new *de novo* clustering algorithms being developed continually. Aside from the established methods described above, several new techniques are being developed as listed in Table 2.1. With the increasing amounts of NGS data produced, newer faster methods need to be explored further.

The main advantage of *de novo* methods is that every input sequence is used which allows identification and classification of unknown or rare species [70]. The main drawback is execution time. As *de novo* methods are not parallelizable, processing time for very large data sets can be prohibitive [152, 103]. Another disadvantage is the inability to compare OTUs between studies since *de novo* OTUs are created within each individual study or sequencing run [79]. Further, *de novo* picking requires comparison of the same gene region [158, 25]. In other words, *de novo* OTUs created with sequences from the V2 hypervariable region cannot be compared with *de novo* OTUs created from the V4 hypervariable region.

Table 2.1: Examples of *de novo* OTU Picking algorithms developed within the past decade arranged by year.

OTU Picking Algorithm	Year	Reference
Two-Stage Clustering	2012	Jiang et al. [72]
M-Pick	2013	Wang et al. [149]
TreeOTU	2013	Wu et al. [156]
Distribution-Based Clustering	2013	Preheim et al. [111]
SWARM	2014	Mahé et al. [89]
bioOTU	2016	Chen et al. [23]
DMclust	2017	Wei et al. [150]
OptiClust	2017	Westcott & Schloss [153]
HmmUFOtu	2018	Zheng et al. [159]

Closed-Reference Picking

In closed-reference OTU picking, the sequences of interest are matched against a reference database and any sequences that do not match at a given similarity threshold are discarded. Common reference databases used for metagenomic studies are Greengenes [30], SILVA [113], and RDP [27]. These databases contain taxonomically annotated copies of the entire 1500 nt 16S gene, although there are differences both between the databases and within each database between release versions [58, 103].

Closed-reference picking is best suited for well-studied microbiomes such as the human oral cavity [79]. The main advantage of closed-reference picking is that it can be parallelized and be very fast even for large datasets [119]. A second advantage is that reference databases allow for comparison of OTUs between studies [152]. The primary disadvantage is the inability to handle sequences from new species since anything not already in the reference database will be discarded [119].

Open-Reference Picking

Open-reference OTU picking is a hybrid of the closed and *de novo* approaches. In this method, sequences of interest are first matched against a reference database as in closed-reference picking, but any unmatched sequences are clustered into *de novo* OTUs instead of being discarded [119]. Open-reference picking therefore uses all the sequences of interest and is partially parallelizable which improves performance [79].

OTU Picking Summary

Along with other steps in NGS data processing, OTU clustering methods have a significant impact on diversity analysis [65, 115, 146, 24, 143]. Hence the choice of clustering method is important for a given study.

Each of the three approaches has its strengths and weaknesses, depending upon the desired outcome of the investigation. Various criteria have been used to measure the "success" of OTU clusters such as: OTU structure, computational efficiency, low OTU artificial diversity inflation, comparison with mock community data, heritability, and consistency or stability [24, 152, 70, 63].

The varied metrics of success have been used to provide recommendations for each of the approaches. Jackson et al. [70] recommend *de novo* picking as producing the most heritable OTUs where heritability "quantifies the percentage of phenotypic variation that is attributable to genetic variability". Westcott and Schloss [152] recommend *de novo* methods as providing the highest quality OTUs based on whether or not a sequence is assigned to the correct OTU as measured by the similarity between sequences.

In contrast, He et al. [63] found that *de novo* picking produces the most unstable OTUs where stability is defined as OTUs that contain the same sequences within OTU

clusters. The authors found closed-reference to produce the most stable OTUs, but as this approach cannot be used with novel sequences, instead opt for open-reference picking. Several studies recommend open-reference picking as the best compromise between speed and the inclusion of new sequences [58, 110, 119].

In general, closed-reference picking is recommended for well-characterized environments like human or mouse gastrointestinal tract. *De novo* picking is necessary for building an initial reference database or for microbiomes that are not present in the reference database such as soil or marine environments. Finally, open-reference picking is recommended for most other situations if the data allows (e.g., if the data is from the same hypervariable region) [79, 110].

2.3.4 Relevant Bioinformatics Software

OTU picking is the last step in a sequential multi-step metagenomic analysis "pipeline" for producing OTUs from raw sequencing data. Specialized bioinformatic algorithms and tools have been developed for each stage of this NGS data processing pipeline [102, 75]. As a result, there is no one standardized way to perform 16S analysis. Put another way, there is no gold standard for creating OTUs. Each investigation is tailored by the biologists conducting it. Researchers are required to consider all existing tools from a confusing if not overwhelming array of options and decide which is appropriate for the study, an approach that often involves considerable time and technical expertise [57, 28]. Efforts have been made to simplify choices for investigators by consolidating the numerous tools into a centralized pipeline program. To this end, several software pipelines have been developed to aid researchers in analyzing 16S data, some examples of which are seen in Table 2.2.

Of these pipelines, the most mature and established are QIIME, mothur, and USEARCH (which was rolled into UPARSE) [123, 110, 58, 28, 120]. Newer pipelines

Table 2.2: Non-exhaustive list of OTU software pipelines arranged by year.

Pipeline	Year	Reference
mothur	2009	Schloss et al. [132]
USEARCH	2010	Edgar [35]
QIIME	2010	Caporaso et al. [20]
CloVR	2011	Angiuoli et al. [7]
Genboree Microbiome Toolset	2012	Riehle et al. [120]
UPARSE	2013	Edgar [36]
mPUMA	2013	Links et al. [86]
Phoenix2	2013	Soh et al. [139]
LotuS	2014	Hildebrand et al. [64]
VSEARCH	2016	Rognes et al. [123]
NINJA-OPS	2016	Al-Ghalith et al. [4]
NG-Tax	2016	Ramiro-Garcia et al. [115]
OCToPUS	2017	Mysara et al. [102]
Hybrid-denovo	2018	Chen et al. [25]
Qiita	2018	Gonzalez et al. [57]

attempt different ways to increase efficiency; but the benefit of using mature pipelines is thoroughly tested functionality, extensive documentation, and up-to-date maintenance by a large collection of developers and experts.

This thesis expands upon QIIME and VSEARCH as they are integral to LOTUS and central to creating an OTU processing pipeline for the OBMM.

QIIME

QIIME [20] (pronounced "chime") stands for **Q**uantitative **I**nsights **I**nto **M**icrobial **E**cology. It is an open source suite of tools written in Python to aide researchers in metagenomic studies. Version 1 was released in 2010 and is available at `www.`

qiime.org. QIIME is a wrapper pipeline, meaning it wraps many other third party bioinformatics tools into one package [103]. The advantage of this approach is that the tools have been developed and benchmarked for a specific purpose. QIIME provides extensive capabilities for 16S analysis, but it must be downloaded locally to each computer and used on the command line [7].

As of this writing, QIIME is still one of the most popular and widely used tools in metagenomic analysis [123, 103, 5]. QIIME version 1 offers a wide array of Python scripts suitable for many tasks in metagenomic analysis from demultiplexing to quality filtering to OTU picking to graphical analysis. However, the flexibility of the tools for use with many different platforms and inputs can create confusion in determining the correct scripts to use and the order in which to use them for a given study. Researchers must invest significant time and technical skills in using QIIME for their studies.

QIIME offers several methods of *de novo* OTU picking including: CD-HIT, BLAST, mothur, SWARM, SUMACLUST, UCLUST, and USEARCH (v5.2 or v6.1). The default is UCLUST although on his website[9], Robert Edgar himself suggests USEARCH (now replaced by UPARSE) over UCLUST for OTU clustering. Edgar developed USEARCH to manage OTU clustering in a reasonable time [35]. USEARCH uses a greedy-centroid based clustering algorithm that uses an initial local alignment of k-mers to identify close matches before doing pairwise alignment to obtain the percent identity. For use in QIIME 1, USEARCH v6.1 must be downloaded separately. As previously documented, USEARCH is available for commercial use[10] and the source code is proprietary.

[9] http://drive5.com/usearch/manual/uclust_algo.html

[10] A 32-bit version of USEARCH is freely available for individual use.

VSEARCH

VSEARCH [123] was developed as an open source alternative to USEARCH. VSEARCH is a suite of command line tools written in C++ for processing metagenomic data. VSEARCH is licensed under the GNU General Public License version 3 and is freely available at `https://github.com/torognes/vsearch`. Westcott and Schloss [152] found VSEARCH performance to be comparable to USEARCH.

VSEARCH can perform several functions including fasta/fastq processing, clustering, searching, and chimera detection (using the UCHIME algorithm developed by Edgar[38]). VSEARCH clusters OTUs using the greedy centroid-based approach. Like USEARCH, VSEARCH uses an initial heuristic to find close matches, then uses a Needleman-Wunsch [104] pairwise aligner with biological modifications to find the optimal alignment and obtain the percent similarity. Percent identity and similarity are used interchangeably and detailed in Appendix C.

Chapter 3

DESIGN

The previous chapter provided the biological background and computational foundation for the novel MST method being investigated by Cal Poly biologists. This chapter details the design of the four components that constitute the computational aspect of the OBMM which is referred to as LOTUS:

1. A relational database to store OTUs and enable source tracking analysis
2. A pipeline (C3PO) to create OTUs from raw Illumina NGS data
3. A method for taxonomic assignment of the library OTUs which allows source classification of unknown samples
4. A web-based user interface for facilitating user access and analysis

LOTUS stands for the **L**ibrary of **OTUs**, but the acronym refers to all the components that collectively function as the computational tool.

3.1 LOTUS Requirements

The purpose of the computational half of the OBMM is to take the raw sequencing data generated by Cal Poly's Illumina MiniSeq and create OTUs either for incorporation into the reference library if the samples are from known sources or for matching analysis if the samples are from unknown sources.

Dr. Black & Dr. Kitts began with four requests for the computational tool:

1. A reference library of OTUs to be built and modified in-house

2. The ability to create OTUs from raw sequencing data

3. The ability to match OTUs from an "unknown" sample to the in-house OTU library

4. A web-based tool similar to CPLOP [141, 140, 13] for ease of use

During development, additional secondary objectives of LOTUS were elucidated including:

- **Consistency and Accuracy.** LOTUS shall produce OTUs in a consistent and accurate manner by utilizing the same pipeline processing and default parameter configuration across all project submissions.

- **Time efficiency.** LOTUS web functionality shall not be impeded by background script processing. LOTUS shall produce all results in a reasonable time frame. Processing time directly correlates with the number of sequences being processed. LOTUS processing for building the reference library or unknown matching shall not exceed 4 hours for a project that contains up to 1 million sequences. *De novo* reclustering of the library shall not exceed 24 hours for up to 1 million sequences. User notification shall occur within 24 hours of file submission.

- **Maintainability.** LOTUS shall include documentation and code commentary for readability and maintenance by future developers. LOTUS shall maintain version control in a centralized repository such as github.

- **User accessibility.** LOTUS shall provide users the complex task of creating OTUs, maintaining the library, and analyzing results with an easy to use interface that does not require a significant amount of time or technical expertise.

- **User configurations.** LOTUS shall have a user-centric design and functionality which will store specific project information on a per user basis. Results and analysis are per user and not between users.

- **Future Experimentation.** LOTUS shall be designed to take into account potential future studies by allowing parameterization and storage of raw files. For example, the web functionality has default parameter settings that do not currently leverage all the input options for C3PO.

- **Future Studies.** LOTUS shall provide a way to store configuration information at the reference library level as well as the user project level to act as the groundwork for comparison between studies in the future.

The biologists' requirements are fulfilled by the four main components of LOTUS. The following sections in this chapter discuss each of the components in detail.

3.2 Reference Library

As a library-dependent MST method, the OBMM requires a reference library of the molecular signatures under investigation. Therefore, the main focus of LOTUS is a reference library of OTUs created and curated by Cal Poly biologists. Existing OTU reference libraries such as Greengenes or SILVA do not contain marine and soil environments [58, 79] and therefore are unsuitable for the OBMM since these environments are explicitly of interest to Cal Poly biologists. This unsuitability makes the requirements for an in-house reference library a necessity. Going forward, the term "reference library" will refer to the LOTUS database library rather than outside reference libraries mentioned earlier in this document.

There is a *caveat* to the OTUs created and stored in LOTUS. Recall that existing libraries consist of OTUs constructed from the entire 1500 nt 16S rRNA gene.

However, the nature of short read sequencing means that the OTUs constructed for LOTUS will not cover the whole genome, but instead will only be for the V3 and V4 hypervariable regions of a few hundred nucleotides. As mentioned in section 2.3.3, *de novo* picking requires comparison of the same gene region and the same principle applies for comparing an unknown to a reference library (e.g., OTUs created from the V2 region must be compared with other OTUs created from the V2 region and not with OTUs created from the V4 region) [158, 25]. In other words, the LOTUS reference library is hypervariable region specific to V3 and V4, which has relevance both in terms of building the library and in matching unknown source OTUs to the library.

The reference library is implemented through a relational database structure which allows for storage, maintenance, and analysis of data. To reiterate a salient point, an OTU is a ***computational* concept**. It is an aggregate of sequence data that represents a molecular signature which is hypothesized to be used for microbial source tracking. The database allows association of this computational concept, the OTU, with the associated metadata of the underlying sequences, and hence allows an OTU to be used for source tracking.

The database design seen in Figure 3.1 models the data as core relations which include HostSpecies, Hosts, Sites, Samples, Sequences, and OTUs. A HostSpecies is the species of an individual animal host from which a fecal sample was collected. A Host is an identified specific individual animal or a population of animals from which known fecal samples are collected. A Site is a specific location from which an unknown environmental sample is collected. A Sample is the substance being tested, either fecal matter if it is collected from a known host or a water sample if it is collected from a geographic location. A single Host can contribute multiple Samples. The Sample table contains the metadata information for both known and unknown samples. A Sequence is a non-singleton, non-chimeric unique DNA sequence produced

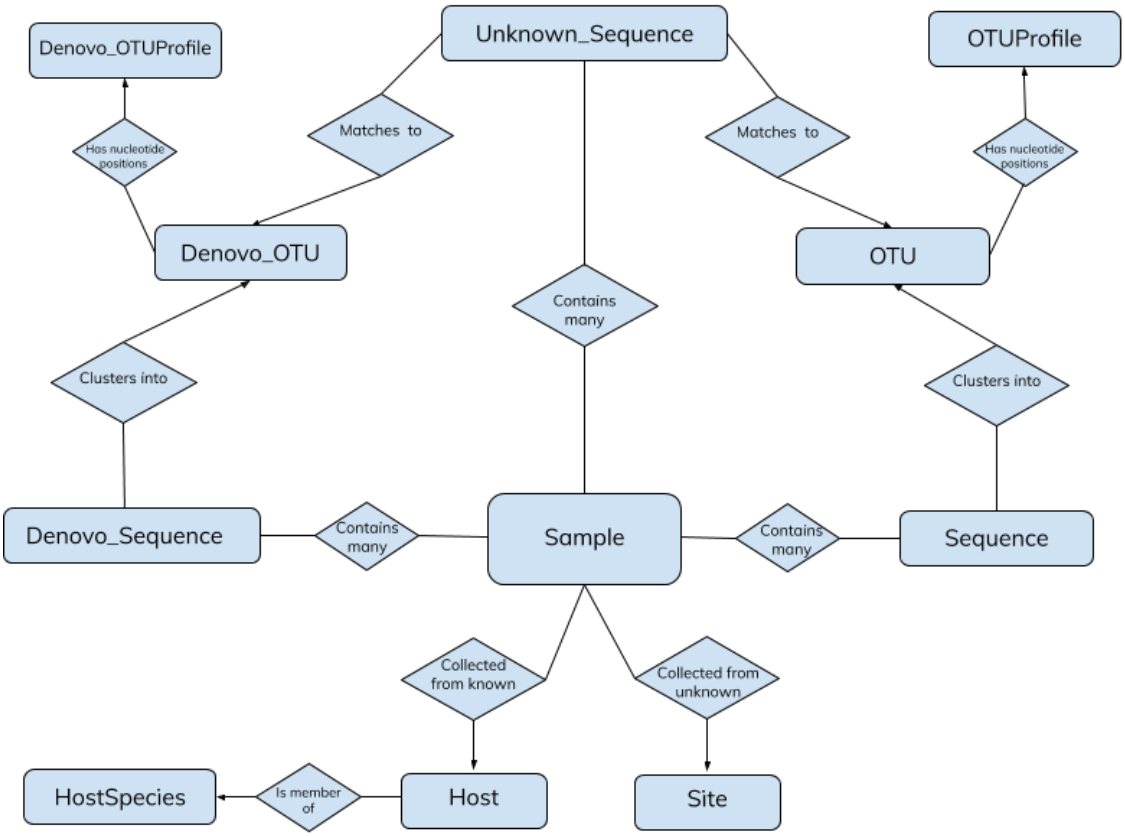


Figure 3.1: LOTUS Database ER Diagram

as the end result of the sequencing platform with further pipeline processing and is discussed further in the following section 3.3. An OTU is a related cluster of sequences and is represented by the centroid sequence used as the basis for the cluster. The LOTUS database entity-relationship (ER) diagram shown in Figure 3.1 summarizes the relationships between these data models. Figure 3.2 shows the translation of the ER model into a collection of relational tables. Note that Figure 3.2 includes the full list of attributes for each table and indicates all primary and foreign keys. A full annotated description of the database model is found in Appendix D.

Some clarification is necessary to explain the seemingly duplicate tables in the database. The LOTUS database has a dual functionality: maintaining a repository of OTUs to be used as the reference library, and storing unknown sequences for

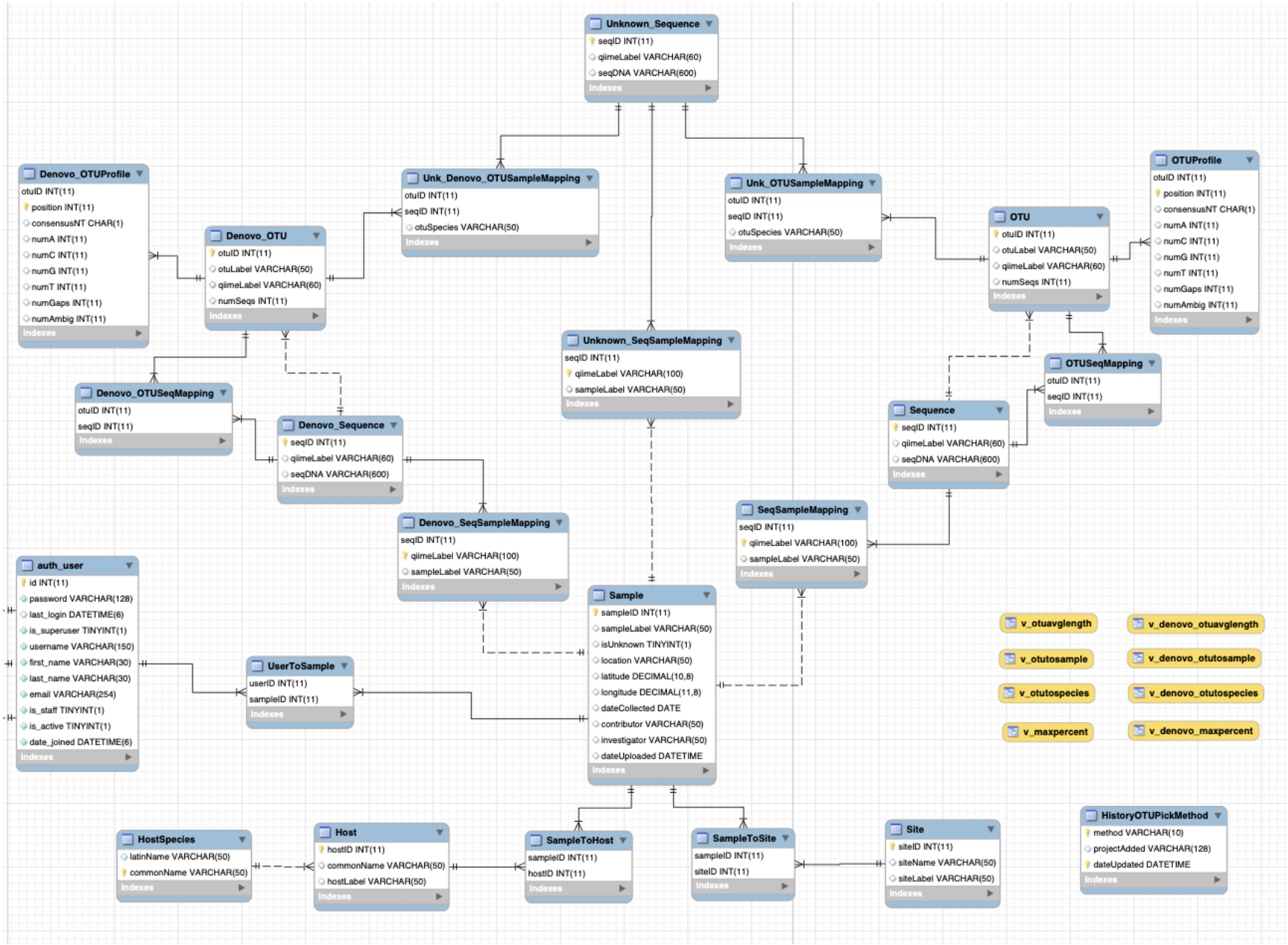


Figure 3.2: LOTUS MySQL Tables

comparison to the reference OTUs. This is further complicated by the fact that the reference OTUs can be created in different ways. As discussed in Chapter 2, OTUs can be clustered through *de novo*, closed, or open picking methods which means the output OTUs will be different depending upon the picking method used. For clarity, this document refers to OTUs produced through the open picking method as "open OTUs" and to OTUs produced through the *de novo* picking method as "*de novo* OTUs". The LOTUS database can therefore use either open OTUs or *de novo* OTUs as the reference library.

However, there is still the question of whether open or *de novo* OTUs will be more appropriate as the reference OTU library for the OBMM, and in fact, this determination is evaluated in Chapter 5. Rather than using a separate database for each type of OTU, the final unified database design allows for data to be constructed using either method. The result of these different OTU picking methods and the sequences produced in their creation is three "branches" in the database. The term "branch" is used here to convey the idea of mapping an OTU back to a Host as seen in Figure 3.3.

The branch concept is meant to give a mental picture of how the tables are related in the database. The regular or default branch is for open OTUs, meaning OTUs created by the default process are stored in the tables specific to the open "branch". The *de novo* branch maps Hosts to Samples to Sequences created during *de novo* processing and then clustered into *de novo* OTUs. In other words, *de novo* OTUs are

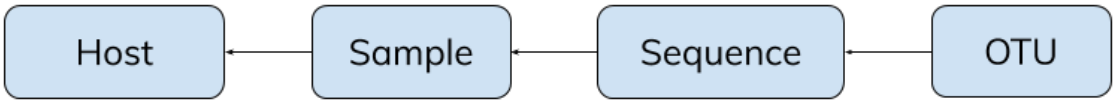


Figure 3.3: Ideological concept of a "branch" that maps from OTU to Host in the LOTUS database.

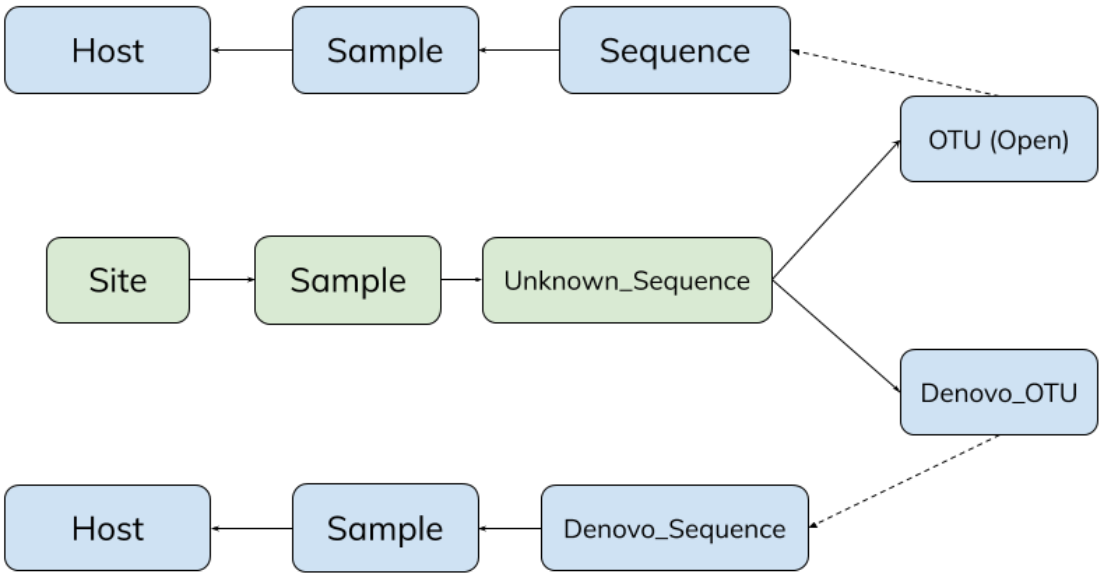


Figure 3.4: Overview of matching unknowns to reference library using the branch concept. There is only one Sample table which is used across all 3 branches. There is also only one Host table and one Site table. This diagram conveys the idea of OTU to Host mapping for the different branches.

created and stored in tables specific to the *de novo* "branch". The "unknown" branch is its own group because OTUs are not created from the unknown sequences, so there are no "unknown OTUs". The Unknown_Sequences are used to map to OTUs for source tracking. Figure 3.4 illustrates the mapping from Host to OTU of the different branches.

Other accessory database tables include OTU profile tables which give a detailed nucleotide breakdown for each OTU and a table to track the history of when either open OTUs or *de novo* OTUs were created. The database also includes tables necessary for the proper functioning of the web-based tool such as a users table for login and authentication. Information on ancillary database tables related to the Django web application is shown in Appendix E. Database views were created to assist with assigning a taxonomy to an OTU and are discussed in Chapter 4.

3.3 Cal Poly Pipeline for Picking OTUs (C3PO)

The relational database acts as the reference library to provide the associations between OTUs and Hosts which allow for source tracking. A fundamental aspect of the database is the inclusion of accurate and consistent data. OTUs must first be created in an external process before being loaded into the database. Therefore, the second component of LOTUS is a pipeline for creating OTUs from raw Illumina sequencing data, called the **Cal Poly Pipeline for Picking OTUs (C3PO)**. The term "pipeline" here refers to a series of sequential data processing steps wherein the input data for each step is dependent on the output data of the preceding step. As shown in Figure 2.10 and discussed in detail in Section 2.3 of the previous chapter, the output of the Illumina platform is a `fastq` file which must be processed before being clustered into OTUs.

As previously asserted, there is no one standard way of creating OTUs leaving it to individual researchers to create their own OTUs for each study [86, 28]. Bioinformatics pipelines such as those in Table 2.2 were developed to facilitate analysis for biologists while reducing the need for time intensive technical skills.

The decision to build a custom pipeline for the OBMM instead of using existing software was based on several reasons. First, the use of an outside lab to provide OTUs was expensive and lacked a method to compare or combine OTUs created during different sequencing runs. Second, OTU picking provided in QIIME tends to produce inflated estimates of OTUs [37]. As the most well-known and established software package, QIIME was used in preliminary feasibility testing as is discussed further in Section 5.2. For this evaluation, five samples were sent to the outside sequencing laboratory MR_DNA[1] which produced 379 OTUs for initial analysis. A comparable run on the same data using *de novo* picking in QIIME produced 4,610 OTUs for the

[1] `https://www.mrdnalab.com`, MR_DNA, Shallowater, TX, USA

same five samples, a 12 fold increase.[2] Third, the end goal of OTU creation is for use as a molecular signature for identifying sources of fecal contamination rather than the metagenomic diversity analysis output provided by standard tools.

A custom pipeline further meets the secondary objectives of providing consistent and accurate data, being time efficient, and allowing maintainability for future developers. By utilizing the same pipeline procedure for all samples, OTUs are created in a standardized manner that enables comparison across studies. All sequences in LOTUS are processed using the same pipeline. Unknown samples are also processed with the pipeline, but are not added to the library. This allows unknown sequences to be more accurately matched to OTUs for source tracking analysis. C3PO also allows developers and biologists to make changes as needed to improve performance and increase efficiency as well as to understand and troubleshoot any problems that may arise.

3.3.1 Pipeline Overview

As briefly mentioned, early initial experiments for the OBMM were run using the outside sequencing laboratory MR_DNA. Based on those results, the QIIME software package was explored further as a means of creating OTUs for the OBMM. However, the issues mentioned above necessitated further research which led to the conclusion that there was a need for creating a protocol specific to this project.

C3PO was constructed from extensive research based on several sources including MR_DNA documentation[3], Robert Edgar's recommended protocols for OTU analysis [33], "Microbiome/Metagenome Analysis Workshop: QIIME" from Brown University [32], and multiple QIIME 1 Google forums [16, 90, 17]. As the documentation from

[2]Note: A more correct comparison would have been to use closed reference picking in QIIME; however, at the time of testing, the simplest option was used with the available data.

[3]MR_DNA documentation provided by MR_DNA included with sequencing run.

MR_DNA is not publicly available, the relevant portion is quoted here:

> Sequencing was performed at MR_DNA (www.mrdnalab.com, Shallowater, TX, USA) on a MiSeq following the manufacturer's guidelines. Sequence data were processed using MR_DNA analysis pipeline (MR_DNA, Shallowater, TX, USA). In summary, sequences were joined, depleted of barcodes then sequences <150bp removed, sequences with ambiguous base calls removed. Sequences were denoised, OTUs generated and chimeras removed. Operational Taxonomic Units (OTUs) were defined by clustering at 3% divergence (97% similarity). Final OTUs were taxonomically classified using BLASTn against a curated database derived from Greengenes, RDPII and NCBI (www.ncbi.nlm.nih.gov, DeSantis et al 2006, http://rdp.cme.msu.edu).

Rather than recreating bioinformatics tools that were already well-tested and widely used by the scientific community, C3PO integrates existing tools to make a pipeline that creates OTUs specifically for the database reference library. The three most popular and established pipelines are QIIME, mothur, and USEARCH (now UPARSE) as discussed in Section 2.3.4. USEARCH is proprietary and commercial for anything other than individual use. The mothur software package is not suited as it is unknown how large the datasets will be for the OBMM and the complete-linkage clustering used in mothur does not handle large datasets well. Therefore, C3PO was built using QIIME scripts mostly following QIIME's workflow. VSEARCH was also integrated as an open source alternative to USEARCH. Recall from Section 2.3.4 that USEARCH and UCLUST are both offered as *de novo* OTU picking methods in QIIME, but that Robert Edgar recommends USEARCH over QIIME's default choice of UCLUST. As USEARCH is proprietary, VSEARCH offers comparable functionality.

C3PO serves two main purposes: 1) creating OTUs from known samples to add to

the reference library, and 2) processing sequences from unknown samples for proper comparison with the reference library. This means that the end product for known samples is OTUs that were constructed either in open or *de novo* fashion while the end product for unknown samples is processed sequences which can be individually compared against the OTUs in the reference library. The pipeline is external to the database and produces files external to the database. As there are multiple steps in the pipeline, each step produces output files whose relevance and function are discussed further in Chapter 4.

From a design perspective, C3PO is broadly divided into 2 parts: Pre-Processing and OTU Picking. The OTU Picking stage is further subdivided into an open-picking approach and a *de novo* approach. The general order of steps for C3PO is: paired-end assembly, conversion of `fastq` to `fasta`/`qual` file formats, quality filtering/demultiplexing, dereplication, chimera removal, and clustering of sequences into OTUs. An overview diagram of C3PO can be seen in Figure 3.5.

De novo vs Open Picking

As demonstrated in the database design and the pipeline overview, LOTUS can use OTUs created either from an open picking approach or from a *de novo* picking approach. An investigative question for this thesis is to determine whether open or *de novo* OTUs are more useful for the OBMM. As previously stated, *de novo* picking must be used for the initial library build. However, as open picking is the default recommended by QIIME, C3PO also uses open picking as its default when adding new samples with known sources to the library.

It is important to note how the *de novo* approach works in LOTUS and the flexibility required of the pipeline. The *de novo* steps for the initial library are as outlined in Figure 4.2, however, there is a slight modification when OTUs already exist

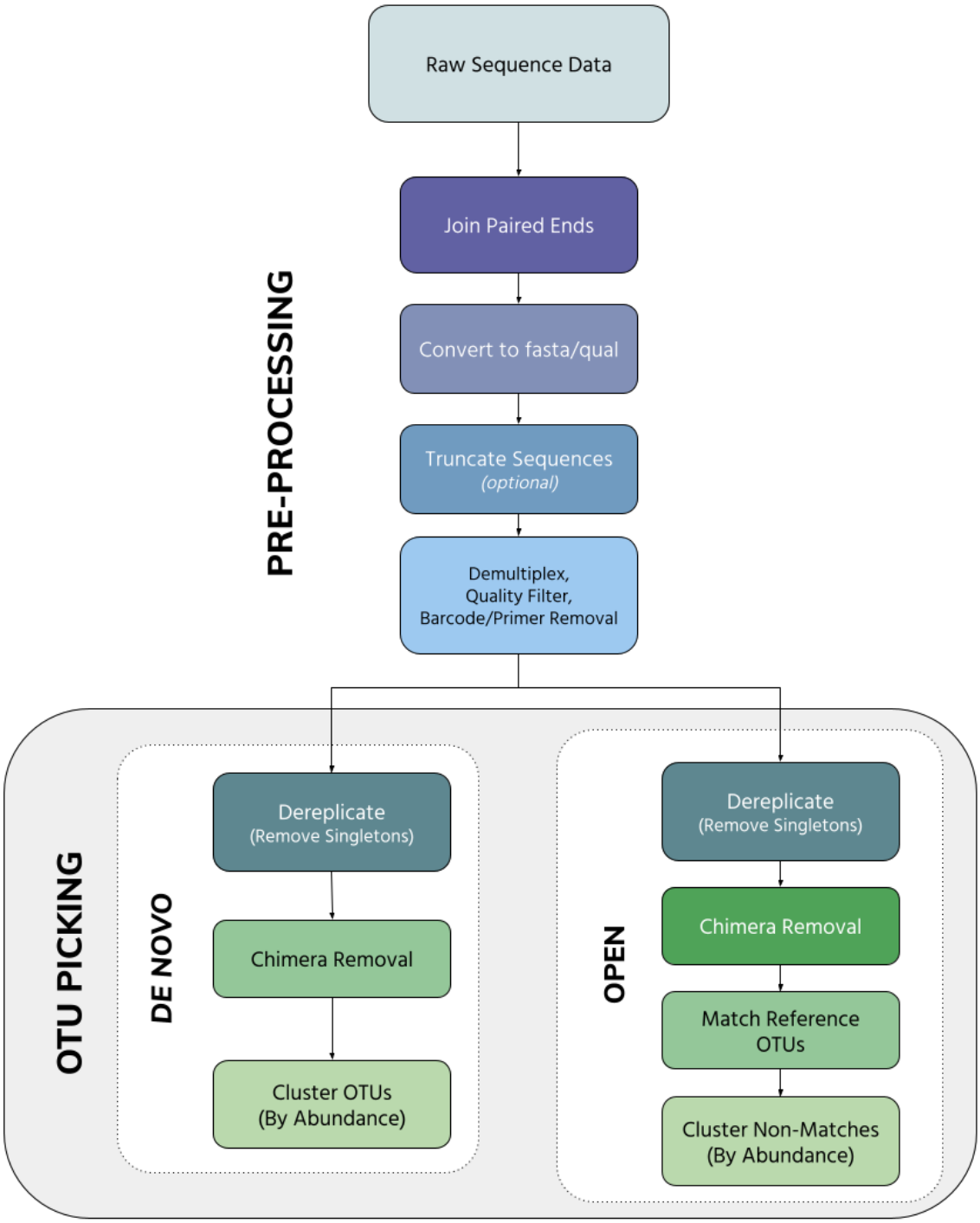


Figure 3.5: Overview of LOTUS C3PO.

in the library. To give researchers the ability to continue evaluating the differences in the approaches, C3PO is built to "recluster" the current library using a *de novo* method. As detailed in Chapter 4, files external to the database are produced on a per user per project basis. When the raw files are added, they are processed to produce open OTUs using the open picking pipeline pathway. The files are saved in a storage hierarchy as shown in Figure 3.10. During reclustering, the appropriate project files for all users are combined into a master file which is clustered *de novo*, meaning all the combined sequences are clustered against each other instead of a library. The external files must be used for reclustering since the database contains curated information and may have discarded sequences such as singletons which can now be used. Reclustering is the only way to provide *de novo* functionality to a library built with ongoing sample and sequencing runs.

Reclustering also solves another problem. The multiplexing aspect of NGS processing applies barcodes to distinguish between samples in a sequencing run. However, according to the biologists, there is a finite number of barcodes that can be used which means **biologists must reuse barcodes**. But it is required by the underlying software that the **barcodes are unique for a given run**. The solution is to use demultiplexed files for reclustering. By separating pre-processing into its own section, the demultiplexing step can remove the barcodes and produce demultiplexed `fasta` files which can then be combined with future samples that have undergone the same pre-processing procedure. Using demultiplexed files addresses the objective of increasing time efficiency by removing redundancy. The *de novo* reclustering also meets the objective of providing future experimentation by allowing biologists to alter parameters (e.g., percent identity or minimum sequence length) and analyze the effects of those alterations.

3.3.2 Pre-Processing

While processing open and *de novo* OTUs involves different steps, several initial steps remain the same. These are grouped into the Pre-processing stage. Pre-processing itself includes three basic steps: 1) joining paired ends, 2) converting `fastq` to `fasta`/`qual` files, and 3) quality filtering. An optional fourth step involves truncating sequences, which is used for specific instance of evaluation testing but is otherwise ignored.

Join Paired Ends

Paired-end (PE) sequencing is a feature of the Illumina NGS platform. C3PO must be able to handle either SE or PE sequencing runs. The input files determine if this step is needed. The user must upload both R1 and R2 `fastq` files for joining paired ends to occur. Figure 2.12 shows the process of joining paired ends. This step, if necessary, must occur before any others as the remainder of the pipeline works from a single file. The result of joining paired ends is a single `fastq` file.

Convert files

The input for the conversion step is a `fastq` file, either the one uploaded by the user in the case of SE sequencing or the output from joining paired ends for PE sequencing. Fastq files are a combination of two earlier file standards: `fasta` and `qual` [39]. Since `fastq` files contain quality information in encoded form, they can be converted into separate `fasta` files[4] which contain the nucleotide sequences and `qual` files which contain the decoded quality scores for each base.

Each `fasta` entry is composed of two lines [39]:

[4]FASTA formatted files can have different file extensions including `.fasta`, `.fna` and `.fa`. These extensions all have the same FASTA format.

Line 1: Illumina sequence identifier (Begins with ">")

Line 2: The actual nucleotide sequence

Similarly, each `qual` entry is composed of two lines [39]:

Line 1: Illumina sequence identifier (Begins with ">")

Line 2: The decoded quality scores for each nucleotide position

An example showing a `fastq` entry and its corresponding `fasta` and `qual` file entries is shown in Figure 3.6.

Truncate Sequences

This optional step was included in the pipeline for one special case for one evaluation test. This step simply trims the sequences at a given nucleotide position which allows testing sequences of different lengths.

Quality Filtering/Demultiplexing

Quality filtering is an essential step in sequence processing to ensure the validity of the reads and reduce errors in classification [15, 110, 75, 148]. The quality scores in the `qual` file are used to filter and retain the high quality reads. Quality scores and quality filtering are discussed in detail in Appendix A.

Demultiplexing assigns the barcodes to the correct sample labels provided in the metadata file submitted by the user. Barcodes and primers (adapters) are removed during this process. An example of demultiplexing is shown in Figure 3.7.

Two other quality control procedures are performed during this step. Sequences below the minimum sequence length are removed as errors [15]. Edgar states that

```
@M01522:151:000000000-B9DJG:1:2110:7582:6405 1:N:0:GTTTCG
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTTAG
GCACGTGTGCCTTTTTGTATGTACCGTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGATCCG
AGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACCGTAA
AATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTAGATATCAC
GAAGAACTCCGATTG
+
CCCCCGG>FGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGCFGJJJJJJJJJJJJJJJJJJJJJJJ@JJJJJJJJJJJJJJJJ
?>JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJ
JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJGGGFGGGGGGGGGGGGGGGGGFCCGGGGGGGGGGGGGGGCCCCC
```

(a)

```
>M01522:151:000000000-B9DJG:1:2110:7582:6405
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTTAG
GCACGTGTGCCTTTTTGTATGTACCGTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGATCCG
AGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACCGTAA
AATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTAGATATCAC
GAAGAACTCCGATTG
```

(b)

```
>M01522:151:000000000-B9DJG:1:2110:7582:6405
34 34 34 34 34 38 38 29 37 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38
38 38 38 38 38 38 38 38 38 38 38 38 34 37 38 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41
41 41 41 41 41 31 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 30 29 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41
41 41 41 41 41 38
38 38 37 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 38 37 34 34 38 38 38 38
38 38 38 38 38 38 38 38 38 38 38 34 34 34 34 34
```

(c)

Figure 3.6: Example of converting fastq to fasta/qual files for sequence id `@M01522:151:000000000-B9DJG:1:2110:7582:6405.` (a) Original `fastq` file entry. (b) `fasta` file entry showing just the sequence. (c) `qual` file entry showing the decoded quality scores.

```
>M01522:151:000000000-B9DJG:1:2110:7582:6405
CTCTCAGTCTGAACCAGCCAAGTAGCGTGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTTAG
GCACGTGTGCCTTTTTGTATGTACCGTATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGATCCG
AGCGTTATCCGGATTTATTGGGTTTAAAGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACCGTAA
AATTGCAGTTGATACTGGGTGTCTTGAGTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTAGATATCAC
GAAGAACTCCGATTG
```

(a)

```
>MB.Hu2_1 M01522:151:000000000-B9DJG:1:2110:7582:6405 orig_bc=CTCTCAGT
new_bc=CTCTCAGT bc_diffs=0
TGAAGGATGACTGCCCTATGGGTTGTAAACTTCTTTTATACGGGAATAAAGTTAGGCACGTGTGCCTTTTTGTATGTACCGT
ATGAATAAGGATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGATCCGAGCGTTATCCGGATTTATTGGGTTTAA
AGGGAGCGTAGGCGGATGCTTAAGTCAGTTGTGAAAGTTTGCGGCTCAACCGTAAAATTGCAGTTGATACTGGGTGTCTTGA
GTACAGTAGAGGCAGGCGGAATTCGTGGTGTAGCGGTGAAATGCTTAGATATCACGAAGAACTCCGATTG
```

(b)

Figure 3.7: Example showing demultiplexing output for sequence id `@M01522:151:000000000-B9DJG:1:2110:7582:6405`. (a) `fasta` entry after conversion from `fastq`. The barcode (`CTCTCAGT`) and primer (`CTGAACCAGCCAAGTAGCG`) are highlighted in red. (b) `fasta` entry after demultiplexing. The sample name highlighted in yellow is now added to the sequence identifier and the barcode and primer have been removed from the sequence.

to get good reads, "all sequences derived from the same biological template [should] start at the same position in the gene and have the same length" as these are more likely to be biologically correct [33]. Lastly, homopolymers of more than size 6 are also removed as a default setting in the software.

3.3.3 OTU Picking

The steps in C3PO's pre-processing stage essentially correct for sequencing errors and format the data for use in the OTU picking stage. There are four steps in the OTU picking stage: 1) dereplication, 2) chimera removal, 3) optional matching reference OTUs, and 4) OTU clustering.

Dereplication

The dereplication step reduces the size of the data to be processed by combining all identical sequences into a single representative sequence, essentially producing only

the unique sequences in a given sequencing run. A mapping file that traces each sequence to the representative sequence is necessary for maintaining source tracking. For C3PO, this mapping file is saved as `derep_out.uc` as shown in Figures 4.2 – 4.5 and this information is loaded into the database in the appropriate SeqSampleMapping table. The dereplication step is also used to remove singletons as discussed in Section 2.3.2.

Chimera Removal

Next, the unique sequences are checked for chimeras. Chimeras are hybrid DNA fragments made from more than one DNA template during PCR. As they are the artifacts of PCR sequencing errors, chimeras cause overestimates of microbial diversity and need to be removed as discussed in 2.3.2.

Match Reference OTUs

This step is both used in the open picking method and in matching unknowns. It is essentially closed-reference OTU picking where the high quality, non-singleton, non-chimeric sequences produced to this point are clustered against the OTUs in the reference library. Each sequence is individually aligned and matched to an OTU at 97% similarity as discussed in Appendices B and C. For known samples, sequences that cluster at 97% similarity or greater are added to the correct OTUs and non-matching sequences are then used in the next step to create new OTU clusters. For unknown samples, sequences that cluster at 97% similarity or greater are mapped to the correct OTUs for source tracking and non-matching sequences are reported as 'No Match'.

```
>_OTU_1;size=103690 MB.Do1_9;size=36343
TGAAGGATGAAGGTCCTACGGATTGTAAACTTCTTTTATAAGGGAATAAAACCTCCCACGTGTGGGAGCTTGTATGTACC
TTATGAATAAGCATCGGCTAACTCCGTGCCAGCAGCCGCGGTAATACGGAGGATGCGAGCGTTATCCGGATTTATTGGGT
TTAAAGGGAGCGCAGACGGGTCGTTAAGTCAGCTGTGAAAGTTTGGGGCTCAACCTTAAAATTGCAGTTGATACTGGCGT
CCTTGAGTGCGGTTGAGGTGTGCGGAATTCGTGGTGTAGCGGTGAAATGCTTAGATATCACGAAGAACTCCGATTG
```

Figure 3.8: A single entry in the fna output file of the OTU clustering step.

OTU Clustering

This step is only used for adding OTUs from known samples to the reference library. High quality, non-singleton, non-chimeric sequences are used for OTU clustering. All the input sequences are clustered against each other at 97% similarity using the VSEARCH style *de novo* method as described previously. The output of the pipeline is an `fna` file which can be used to load the database. The `fna` file is the exact same format as a `fasta` file, with an OTU label rather than a sample label. An example of an OTU entry in an `fna` file is shown in Figure 3.8. A human researcher cannot derive insights such as the percent of sequences that belong to a given sample or which sequences are related simply from glancing at the output files in the pipeline. For meaningful analysis, this file along with information in many others must be incorporated into the database.

3.4 Taxonomic Assignment of OTUs

LOTUS is a tool to aide biologists in determining sources of fecal contamination in environmental samples. The reference library is composed of relational database tables which map an OTU through its Sequences to the associated Hosts. In an ideal situation, all related sequences in an OTU would belong to the same phylogenetic group and thus map back to a single host species [102]. In reality, related sequences can come from multiple hosts, and even completely identical sequences can be found in different host species. This is one reason that improving OTU clustering is an

active area of research in the field of bioinformatics.

Cal Poly CAB biologists are interested in using OTUs as molecular signatures for source tracking. Simply put, biologists are asking "Are there OTUs that are specific to a given species and that species only?". The existence of such OTUs would answer the question of whether or not an OTU could be used to identify specific hosts as sources of contamination. As a concrete example, biologists want to know if there exists an OTU that is only ever found in dogs since that dog-specific OTU would act as a molecular signature indicating the presence of contamination due to dogs.

As there is no ideal OTU clustering method, the definition of a successful OTU cluster varies with each study. In line with the biologists' focus, this thesis defines a successful OTU as one that clusters to a single species, called the plurality species, i.e., the species that has the largest number of sourced sequences. The third component of LOTUS is a method of assigning taxonomy to OTUs based on this definition. More formally, LOTUS uses a measure called **purity** to find the plurality species by percentage of sequences present in the OTU. A 90% pure OTU has 90% of sequences from the plurality species. Section 5.1.1 explains clustering purity in more detail.

Using a purity threshold, LOTUS classifies OTUs as either single-source or multi-source for taxonomic assignment:

- **Single-source** OTUs have a purity greater than the given purity threshold.
- **Multi-source** OTUs have a purity less than or equal to the given purity threshold.

For example, if the given purity threshold is 95%, then a single-source cat OTU would need to be 95% pure, meaning more than 95% of cluster sequences need to be from cats. Conversely, a multi-source OTU would be any OTU in which the plurality species was 95% or less. Single-source OTUs can also be referred to by the plurality

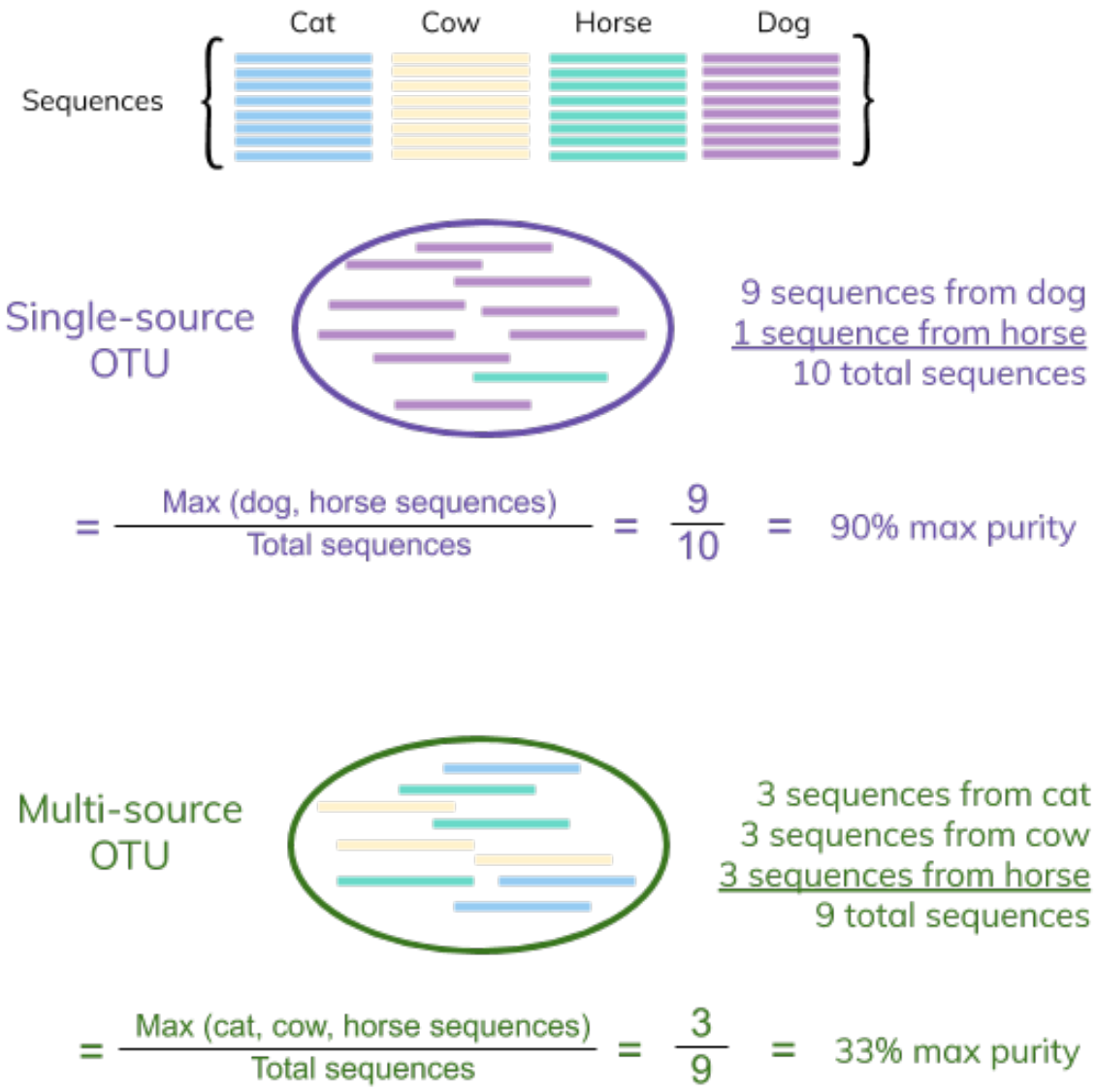


Figure 3.9: Single-source vs Multi-source OTUs. The species with the maximum number of sequences present in the OTU determines the purity of the OTU. In this example, the purity threshold for single-source OTUs is 75%.

species, i.e., a single-source cat OTU is simply termed a "cat OTU". Examples of single-source and multi-source OTUs are shown in Figure 3.9.

If there is no purity threshold, the default behavior assigns the OTU to the plurality species. A cat OTU would simply need more sequences to be from cats than from any other species, regardless of the percentage against the entire cluster. A practical scenario would be a case where an OTU consisting of 8 sequences has 2 sequences from a cat, and has 6 other sequences each from a different species. The OTU would be classified as cat even though the OTU purity was 25%. Determining the best purity threshold for the OBMM is evaluated in Section 5.6.

3.5 Web-Based User Interface

As with other tools seen in Table 2.2, LOTUS attempts to facilitate investigation of OTUs by providing easier access to biological study by simplifying the data processing for researchers. Therefore, the last component of LOTUS is the web-based user interface. A web application is generally more intuitive for users than command line programs and will also allow access by multiple users across different workstations. This component is essential for users to interact with the processed data. The goal for the web application is a simplified interface with minimal input requirements and straightforward results reporting. A detailed description of the web-based user interface is discussed in Section 4.5.

A few subjective concepts are defined here to understand LOTUS web application processing. A **sample** is the organic or environmental material to be DNA sequenced. A **sequencing run** is the process of running the experimental material through the Illumina NGS platform to obtain the sequencing results. A single sequencing run can contain multiple samples due to the multiplexing feature of Illumina platforms. This document refers to the final files produced by a single sequencing run along with all

the sample metadata as a **project**. Users upload "projects" to the web application. Users may submit multiple projects, with the limitation that they may only submit a single project at any one time. The configuration parameters selected by the user are stored with the project for future reference.

3.5.1 File Hierarchy

Because of all the external files used and produced by C3PO, the web application needs an appropriate file storage hierarchy. The files are organized as shown in Figure 3.10. The primary differentiation is between knowns and unknowns. User projects with known samples are used to build the library while those with unknown samples are used for matching analysis. Building the library simply means that OTUs are created by C3PO and added to the database. The file tree for known projects further branches into open OTUs and *de novo* OTUs. User folders are organized based on the OTU picking method used. The hierarchy allows multiple users to contribute to the library and each user can submit multiple projects.

3.5.2 User Types

There are four types of users for LOTUS in order of increasing permissions: public, guest, staff, and admin. Public users are unregistered users of the website. No account is required. It represents the access that is given to the general public and is viewable by anyone with an internet connection. Public access is limited to browsing and searching information about the current default OTU library. Guest users are users that register an account and login in. Guest users are able to browse the library and submit unknown samples for matching to the reference library, but do not have access to modify or update the library. The next level are staff users. These are registered and authorized users that can do everything a guest user can, but also have permission

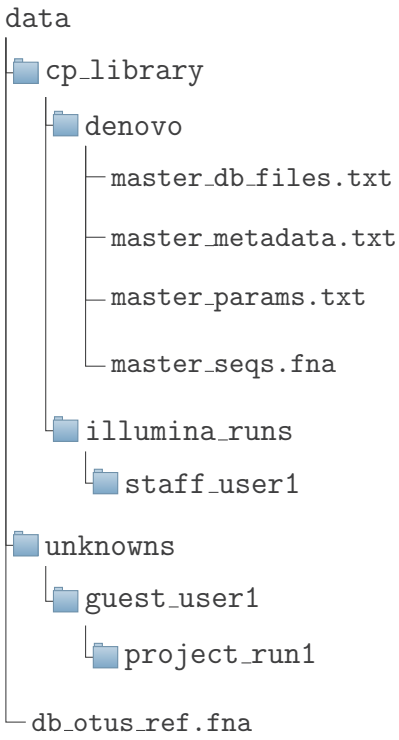


Figure 3.10: Raw and processed file hierarchy in LOTUS. Separated primarily into knowns (`cp_library`) and unknowns. Knowns are used to build the library. The library can be built using *de novo* or open methods. The default method is open and those files are stored in the `illumina_runs` folder. Files generated during *de novo* reclustering are stored in the `denovo` folder. Users have their own folders and each user can have multiple projects.

Table 3.1: The four user types for LOTUS

User Type	Account Required	Request Special Access	Allowed Actions
Public			Browse Public Informational Pages
Guest	✓		Public Actions + Submit Unknown Samples, View Matching Results
Staff	✓	✓	Guest Actions + Build Reference Library, Recluster Reference Library
Admin	✓	✓	Staff Actions + Manage Authorized Users

to modify and update the library. This applies to both the default library that uses open OTUs and the *de novo* library created by the reclustering command. Lastly, the admin user is a special case that has all the permissions of the staff user and has access to the admin page which can modify and update other users. The four user types are summarized in Table 3.1.

3.5.3 Requirements

The web application completes the function of LOTUS as a tool for MST research.

General Application Requirements

The web application shall implement the following general application requirements:

GR1. Sample: General Information. LOTUS shall display the provenance information of all Samples used to create the reference library in a searchable table format. The information displayed shall include the following:

- SampleID: the database unique identifier for the sample
- SampleLabel: the sample label as given by the user in the metadata_template
- HostLabel: the host label as given by the user in the metadata_template

- CommonName: the common name of the host species
- IsUnknown: indicates if the Sample is from an unknown environmental source and will be false for samples in the reference library
- Location: the location from which the sample was collected as given by the user in the metadata_template
- Latitude: the latitude of sample collection as given by the user in the metadata_template
- Longitude: the longitude of sample collection as given by the user in the metadata_template
- DateCollected: the date the sample was collected
- Contributor: the agency submitting the sample
- Investigator: the technician or student running the sample

GR2. Sample: Detailed Information. LOTUS shall display provenance information for the selected Sample. The information displayed shall include the following:

- SampleID: the database unique identifier for the sample
- SampleLabel: the sample label as given by the user in the metadata_template
- Date Collected: the date the sample was collected
- Location: the location from which the sample was collected as given by the user in the metadata_template
- Host: information about the host including the label and species
- Contributor: the agency submitting the sample
- Investigator: the technician or student running the sample

- Number of Sequences: the number of sequences (excluding chimeras and singletons) produced from this sample and used by the library
- List of OTUs: a linkable list of otuLabels that contain sequences from the selected Sample

GR3. OTU: General Information. LOTUS shall display general OTU information of all the default open-picked OTUs in the reference library in a sortable table format. The information displayed shall include the following:

- OtuID: the database unique identifier for the OTU
- OtuLabel: the label produced by C3PO during creation of OTUs in {project_name}_OTU_{otu_counter} format
- QiimeLabel: the name of the centroid consensus sequence produced by C3PO in {sampleLabel}_{seq_counter} format
- NumSeqs: the number of sequences (non-dereplicated, non-chimera, non-singleton) that make up the OTU
- OtuAvgLength: the average nt length of all the sequences in the OTU

GR4. OTU: Detailed Information. LOTUS shall display detailed information for a selected individual OTU. The information displayed shall include:

- OtuID: the database unique identifier for the OTU
- Centroid (Consensus) Sequence: the actual nucleotide sequence of the centroid of the OTU
- OtuLabel: the label produced by C3PO during creation of OTUs in {project_name}_OTU_{otu_counter} format
- Total number of sequences: the number of sequences (non-dereplicated, non-chimera, non-singleton) that make up the OTU

- Average Length: the average nt length of all the sequences in the selected OTU
- Species Breakdown: a table showing the breakdown of sequences in the OTU by species using number of sequences and percentage
- Species Classification: the taxonomic classification of the selected OTU

GR5. OTU Graphical Summary. LOTUS shall provide a data visualization of OTU purity for all OTUs in the library in graphical form. One graph will show all OTUs by species purity and the following graphs will show purity by individual species.

GR6. Library Summary. LOTUS shall provide a summary of the Samples, Sequences, and OTUs in the current reference library. The information displayed shall include:

- Total OTUs: the total number of OTUs in the library
- Sample and Host: the number of host species and samples used to create the current library
- Total Number of Sequences: the total number of sequences (non-dereplicated, non-chimera, non-singleton) used to create the library
- List of species: a list of the species that are currently represented by OTUs in the library
- Species Breakdown Graphs: a pie chart showing the species breakdown by the number of sequences and a pie chart showing the species breakdown by number of OTUs

GR7. Simplified Interface. LOTUS shall provide a clear, self-explanatory user interface to include navigation and user interaction for users with varying technical skills.

GR8. User Management. LOTUS shall provide a system of user permissions to assign users to one of the four types in Table 3.1 for authorization and authentication purposes.

GR9. Downloadable Templates. LOTUS shall provide downloadable templates for files submitted by users and needed by C3PO including the `lotus_metadata_template.xslx` file.

Requirements to Build Library

The web application shall implement the following requirements to provide functionality for initially building or adding OTUs to the reference library using the open OTU picking method:

BR1. Restricted Access. LOTUS shall allow only authorized users with staff permissions or higher as shown in Table 3.1 to modify the reference library.

BR2. Data Upload. LOTUS shall provide an authorized user a way to upload project data for samples of known provenance which shall include at least one `fastq` file and the metadata_template file.

BR3. Create OTUs. Upon user submission of files, LOTUS shall return a success page for the user to continue browsing and offload C3PO processing to run in the background. LOTUS will run the C3PO open picking pipeline to produce open-picked OTUs.

BR4. Load Database. Upon completion of the C3PO pipeline script with the associated output files, LOTUS shall parse output files to either initially load or update the database.

BR5. User Project File Storage. LOTUS shall store all project files submitted by

a given user and all associated output files produced by C3PO processing for that project. Users may submit multiple projects.

BR6. User Notification. LOTUS shall send an email notification to the user when the library has been successfully created or updated.

Requirements to Recluster Library

The web application shall implement the following requirements to provide functionality for *de novo* reclustering of the OTUs in the reference library:

RR1. Restricted Access. LOTUS shall allow only authorized users with staff permissions or higher as shown in Table 3.1 to modify the *de novo* library.

RR2. Optional Parameters. LOTUS shall provide users with the option to choose different parameters for creating *de novo* OTUs for further studies. The parameters shall include: barcode length, minimum quality score, minimum sequence length, maximum sequence length, and percent identity.

RR3. Recluster *De novo* Database. Upon user-initiated reclustering, LOTUS shall return a success page for the user to continue browsing and offload C3PO processing to run in the background. LOTUS will run the C3PO *de novo* picking pipeline using the `fastq` and metadata files of samples already in the library to produce *de novo* OTUs.

RR4. Store Configuration. LOTUS shall store the configuration of the *de novo* library in the `master_params.txt` file for future reference. This file includes the parameters and projects used to create the library.

RR5. User Notification. LOTUS shall send an email notification to the user when the *de novo* OTU library has been successfully reclustered.

Unknown Matching Requirements

The web application shall implement the following requirements to provide MST functionality:

MR1. Restricted Access. LOTUS shall allow only authorized users with guest permissions or higher as shown in Table 3.1 to use the reference library for unknown matching. A user must have a registered account to use this feature.

MR2. Data Upload. LOTUS shall provide an authorized user a way to upload project data for environmental samples of unknown provenance which shall include at least one `fastq` file and the metadata_template file.

MR3. User Project File Storage. LOTUS shall store all project files submitted by a given user and all associated output files produced by C3PO processing for that project. Users may submit multiple projects.

MR4. Match Unknowns. Upon user submission of files, LOTUS shall return a success page for the user to continue browsing and offload C3PO processing to run in the background. LOTUS will run the C3PO unknown matching pipeline to match sequences to the reference library and store the results.

MR5. User Notification. LOTUS shall send an email notification to the user when the unknown matching results are ready for viewing.

MR6. View Matching Results. LOTUS shall display the results for each unknown environmental sample submitted by user on a per project basis. The information displayed shall include:

- Project Name: the name of the project submitted by the user
- Date run: the date the files were uploaded and run through C3PO

- Files used: the files submitted by the user
- Parameters used: the parameters selected by the user
- Sample Matching Results: a container for each sample that includes sample metadata information and shows the number of sequences from the sample that matched to an OTU species in the library

MR7. Download Matching Results. LOTUS shall provide users a downloadable pdf report of the unknown matching results on a per project basis.

Chapter 4

IMPLEMENTATION

This chapter describes specifics of the implementation of LOTUS as documented in this thesis. The source code is available at `https://github.com/gdewitte06/lotus`.

4.1 Languages & Environment

The LOTUS software uses different programming languages including Python, SQL, and bash across all the component parts. The database uses SQL, C3PO is written in bash using Python QIIME scripts and executable C++ VSEARCH commands, and the web application is written in Python.

QIIME v1.9.1 requires the use of Python 2.7 and documentation recommends installation using Miniconda, a mini distribution of Anaconda which is a Python package manager that includes data science and bioinformatics focused packages[1]. For LOTUS, the full Anaconda v4.3 distribution was installed[2]. Python packages are essentially third party libraries. QIIME requisite packages are `qiime`, `matplotlib` v1.4.3, `mock`, and `nose`. Additional Python packages installed for LOTUS but not included in the Anaconda distribution include: `biopython`, `django`, `django-tables2`, `django-crispy-forms`, `django-nvd3`, `django-bower`, `celery`, `mysql-python`, `pymysql`, and `xlrd`. A conda virtual environment was implemented to keep track of these packages (dependencies) as part of installing QIIME. The complete list of Python packages is found in `requirements.txt` file of the web application.

Additional dependencies external to the Python virtual environment are:

[1] QIIME Installation: `http://qiime.org/install/install.html`
[2] Anaconda Installation: `https://docs.anaconda.com/anaconda/install/`

- VSEARCH[3] v2.10.4: bioinformatics tool detailed in Section 2.3.4
- RabbitMQ[4] v3.8.0: a message broker to assist with asynchronous `celery` tasks
- Node.js[5] v10.16.3: necessary for the package manager `bower`[6] which is in turn a dependency of the `django-nvd3` graphing package

The next four sections discuss the implementation of each of the components of LOTUS.

4.2 Reference Library

The reference OTU library is implemented through a relational database using MySQL version 5.7.20. The data models discussed in Section 3.2 were implemented as designed and shown in Figure 3.2 using the SQL CREATE TABLE statements found in Appendices D and E. For display on the web-based user interface, the database was connected to the web application's Object Relational Model (ORM). While an ORM can be used alone as the database, for this project the data is solely handled by C3PO. The ORM is only used to retrieve the data for display and is not given permission to modify the underlying database.

4.3 Cal Poly Pipeline for Picking OTUs (C3PO)

C3PO is implemented using five scripts written in bash. The scripts are called from the web application in specific order depending on the requested function as depicted in Figure 4.1. Each functional pathway utilizes two scripts: one for Pre-Processing and one for OTU Picking.

[3] `https://github.com/torognes/vsearch`
[4] `https://www.rabbitmq.com/`
[5] `https://nodejs.org/en/`
[6] `https://bower.io`

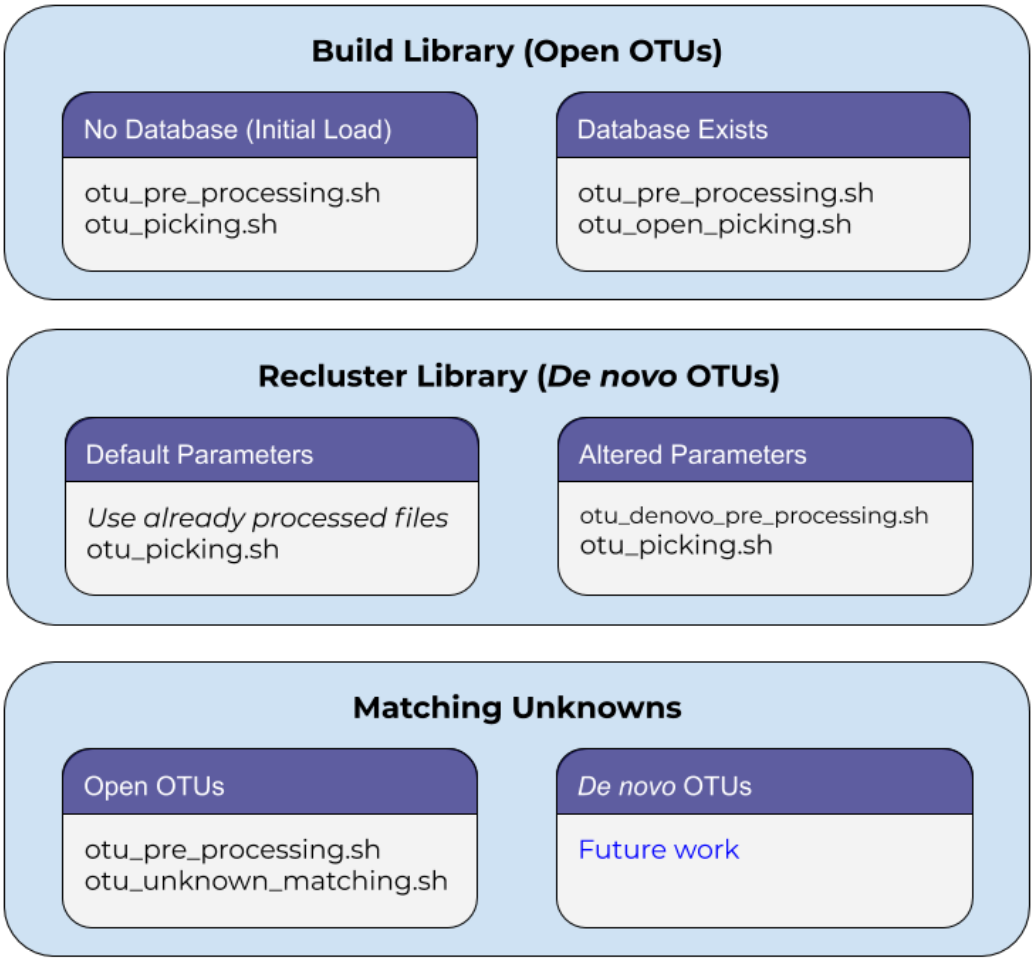


Figure 4.1: Overview of C3PO bash scripts for different functionality.

The Pre-Processing scripts use QIIME 1 commands to perform general NGS data processing and quality filtering. Basically, the pre-processing step is meant to convert NGS reads into more accurate biological sequences. Detailed information on all QIIME scripts is found at `http://qiime.org/scripts/index.html`.

The OTU Picking scripts use VSEARCH commands to organize the sequences, remove chimeras, and finally either cluster the sequences into OTUs or match sequences against a set of reference OTUs. All VSEARCH documentation can be downloaded from `https://github.com/torognes/vsearch/releases/download/v2.10.4/vsearch_manual.pdf`.

Several secondary objectives are met by these pipeline scripts. The reuse of scripts and the standardized procedure between scripts ensure consistent and accurate data. The modular nature of the scripts aide maintainability and future experimentation. The reuse of processed files reduces redundancy and improves time efficiency.

Two citations are needed for specific tools used in the pipeline. The QIIME script `join_paired_ends.py` uses the `fastq-join` tool from ea-utils[7] [8]. The VSEARCH chimera removal command `vsearch --uchime_denovo` utilizes the UCHIME algorithm developed by Robert Edgar [38].

Complete information produced by the pipeline is saved in the user project directory even if unused by LOTUS. As one example, the `join_paired_ends.py` QIIME script produces three files:

- `fastqjoin.join.fastq`: the successfully joined reads
- `fastqjoin.un1.fastq`: the unjoined forward reads from the R1 `fastq`
- `fastqjoin.un2.fastq`: the unjoined reverse reads from the R2 `fastq`

In this case, only the joined reads are used and the unjoined reads are ignored in LOTUS processing. Decisions regarding input and output file usage in C3PO were made in an effort to determine what best served OTU production as needed for the OBMM.

The different pipeline pathways are shown in Figures 4.2, 4.3, 4.4 and 4.5. Each figure shows the overall flow of information starting with the user and going through the files produced for use in the database. The pipeline steps as outlined in Figure 3.5 are shown with the corresponding QIIME script/VSEARCH command used and the output files produced.

[7] `https://expressionanalysis.github.io/ea-utils/`

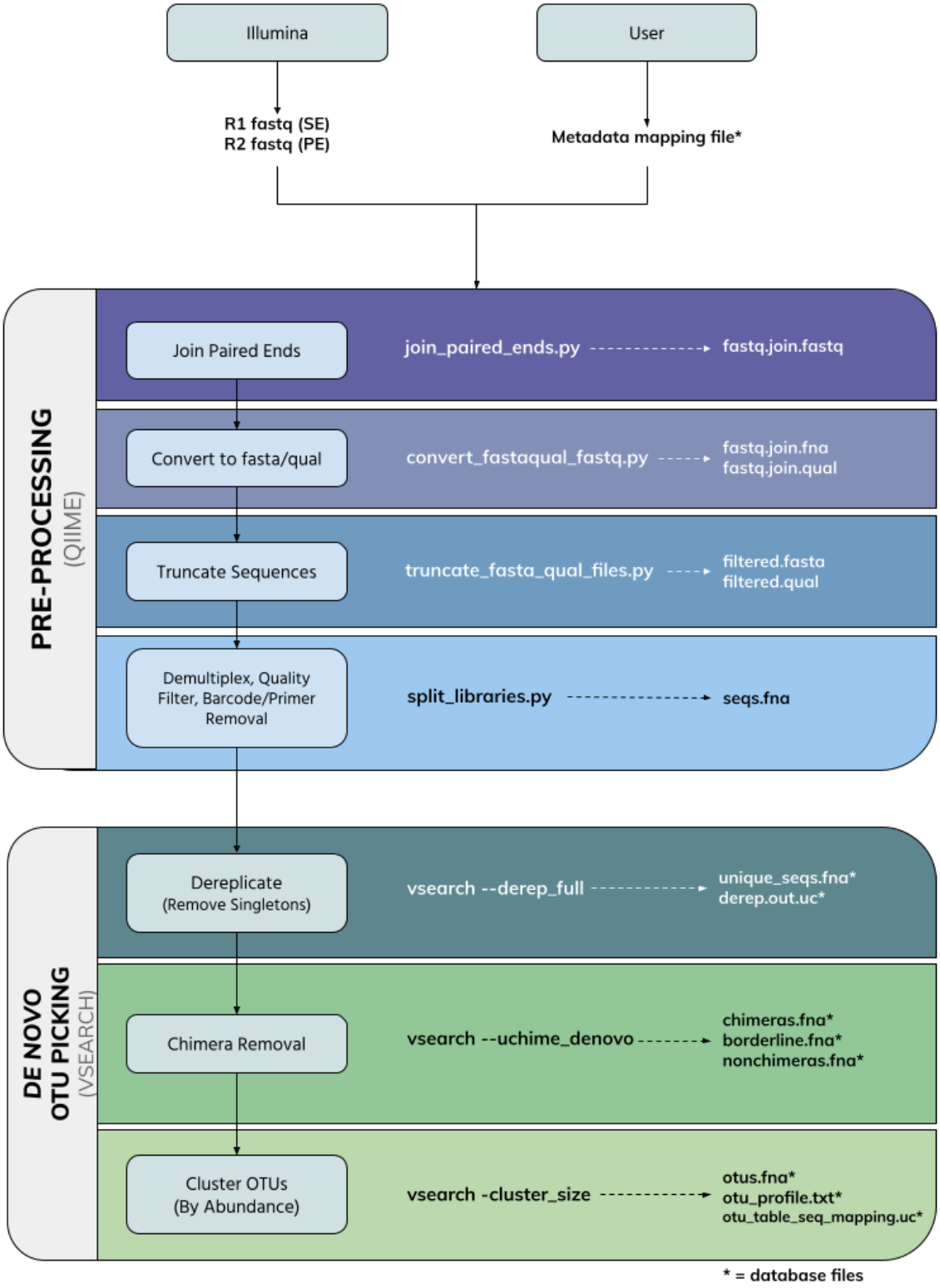


Figure 4.2: C3PO Initial *De novo* OTU Processing Pipeline.

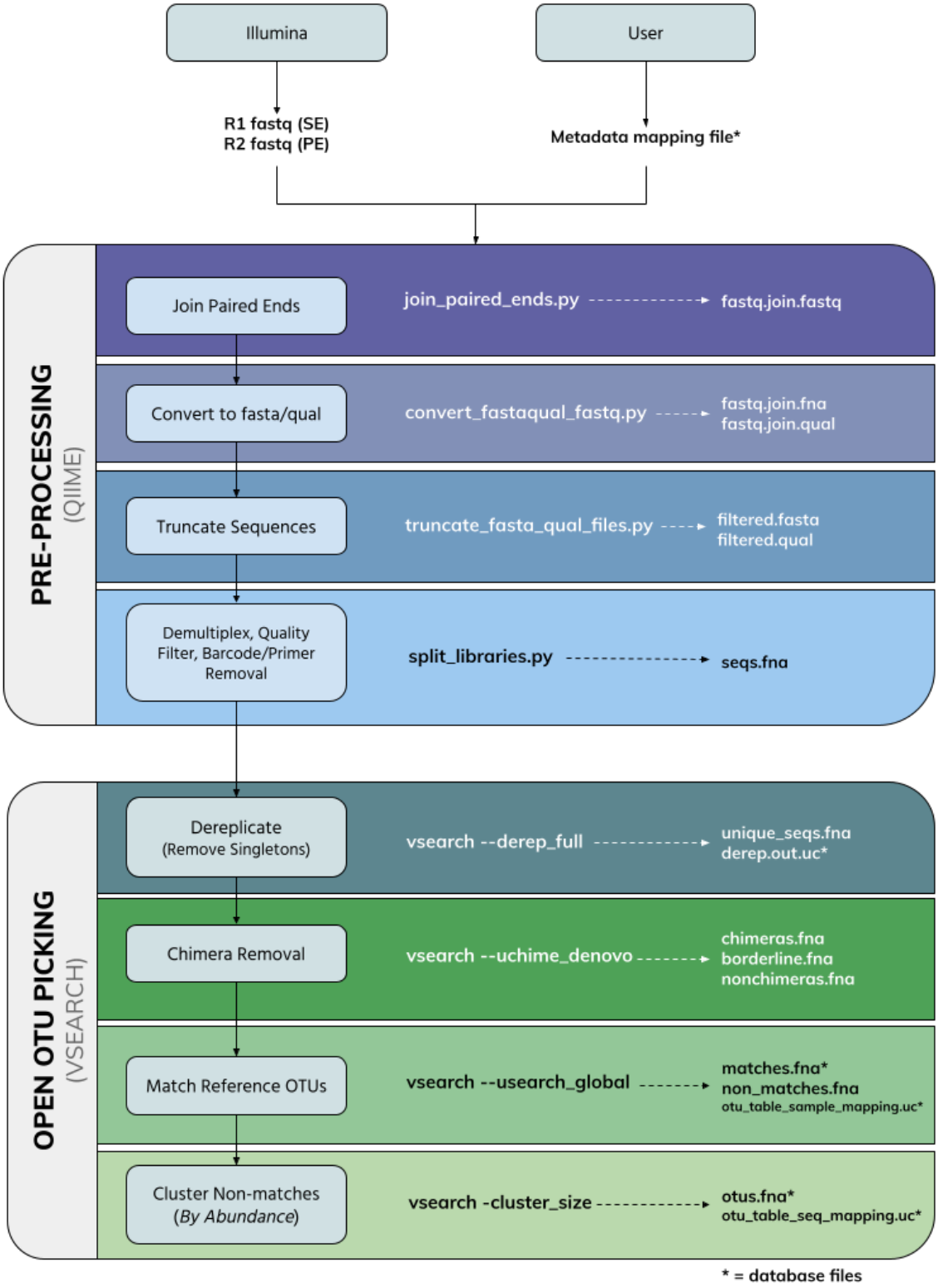


Figure 4.3: C3PO Default Open Picking OTU Processing Pipeline.

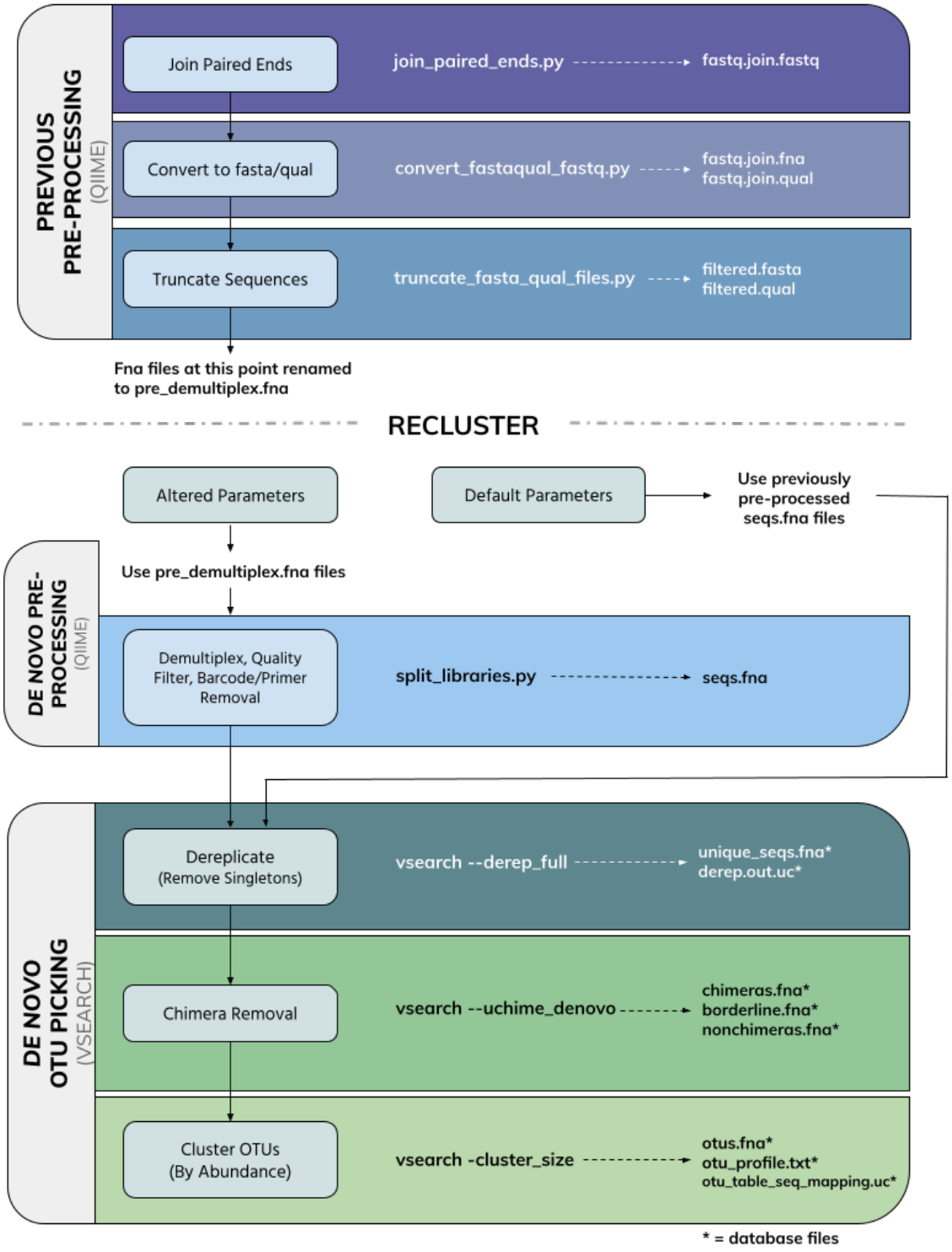


Figure 4.4: C3PO Recluster *De novo* OTU Processing Pipeline.

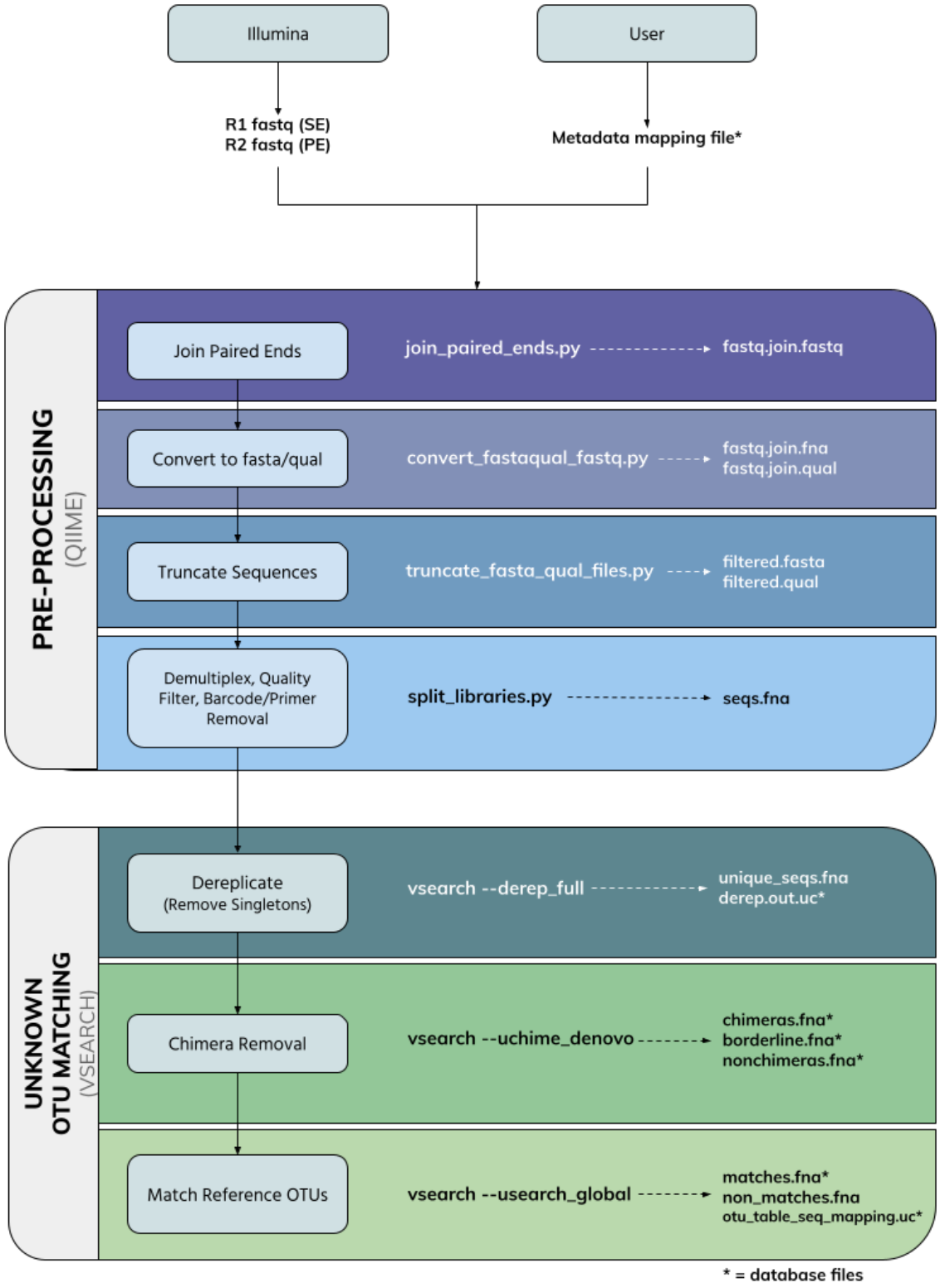


Figure 4.5: C3PO Processing Pipeline for Unknown Matching.

4.4 Taxonomic Assignment of OTUs

For the OBMM, successful OTUs are those whose sequences are clustered by phylogenetic grouping (indicating phylogenetic relatedness) as measured by the highest number of sequences per source. Such OTUs are referred to in this thesis as single-source (or pure) OTUs, previously shown in Figure 3.9. The calculations for such groupings are accomplished through the use of SQL View statements which are shown in Figure 4.8. SQL views are not stored in the database, but are executed as a query each time they are used. Views are a convenience during the early phase of building the library when samples, sequences and OTUs change with each project run. If the library grows to a sufficient size where no more samples need be added, the views can then be made into regular SQL tables.

Both *de novo* and open OTU branches in the database each have four views to enable and aide source tracking. The main View used for this purpose is `v_OTUToSpecies` which partitions the sequences in an OTU cluster by species. A similar view is `v_OTUToSample` which partitions the sequences in an OTU cluster by sample. The *de novo* equivalent to `v_OTUToSpecies` is `v_Denovo_OTUToSpecies` and the other *de novo* views follow the same convention. Example results data returned by these Views is shown in Figures 4.6 and 4.7.

otuID	numSeqs	numSamples	commonName	percent
1	103690	7975	Cat	7.69
1	103690	40489	Dog	39.05
1	103690	268	Horse	0.26
1	103690	54958	Human	53.00

Figure 4.6: Example results for a single OTU in OTU To Species View. OTU 1 contains 103,690 sequences from 4 species. The number of sequences and percent breakdown is calculated for each species. This example OTU would be classified as single-source Human since 53% of sequences (54,958) in the cluster are from Human sources. An equivalent terminology is to say the OTU is 53% pure human.

otuID	numSeqs	numSamples	sampleLabel	commonName	percent
1	103690	7975	MB.Ca1	Cat	7.69
1	103690	40489	MB.Do1	Dog	39.05
1	103690	268	MB.Ho1	Horse	0.26
1	103690	54708	MB.Hu1	Human	52.76
1	103690	250	MB.Hu2	Human	0.24

Figure 4.7: Example results for a single OTU in OTU To Sample View. OTU 1 contains 103,690 sequences from 5 samples (with 2 samples from the same species). The number of sequences and percent breakdown is calculated for each sample.

The Views are modeled in the web application's ORM in order to display the taxonomic purity graphs. The Views are used to determine whether an OTU is classified as single-source or multi-source. Single-source OTUs can then be assigned a single taxonomy based on the maximum percent purity as detailed in Section 5.1.1 and calculated by the `v_MaxPercent` View seen in Figure 4.8c.

4.5 Web-Based User Interface

Django[8] is an open source web framework used to create websites in Python. The LOTUS web application is implemented with Django v1.11 as it is the last stable Django version that supports Python 2.7 which as noted is required by QIIME 1. To simplify development, a single virtual environment using the same Python interpreter was created for LOTUS that incorporated both the backend QIIME scripts and the frontend web application. Django 1.11 documentation is found at `https://docs.djangoproject.com/en/1.11/`.

A very important objective of C3PO and the web functionality is time efficiency. Processing NGS data can take a long time, much longer than a typical 10 minute web session[9]. If this processing were to occur as a normal response of the web application, it would appear to the user that the screen had frozen for hours or even days. In

[8] `https://www.djangoproject.com`

[9] Anecdotally, processing over 565,000 sequences took around 45 minutes.

```
/* OTU Breakdown By Sample */
CREATE OR REPLACE VIEW v_OTUToSample AS
SELECT o.otuID, o.numSeqs, COUNT(*) AS numSamples, ssm.
    sampleLabel, h.commonName,
ROUND((COUNT(*)/o.numSeqs * 100), 2) AS percent
FROM OTU o
JOIN OTUSeqMapping osm ON o.otuID = osm.otuID
JOIN Sequence s ON osm.seqID = s.seqID
JOIN SeqSampleMapping ssm ON s.seqID = ssm.seqID
JOIN Sample sa ON ssm.sampleLabel = sa.sampleLabel
JOIN SampleToHost sth ON sa.sampleID = sth.sampleID
JOIN Host h ON sth.hostID = h.hostID
GROUP BY o.otuID, ssm.sampleLabel, h.commonName;
```

(a) OTU To Sample View Query

```
/* OTU Breakdown By Host Species */
CREATE OR REPLACE VIEW v_OTUToSpecies AS
SELECT o.otuID, o.numSeqs, COUNT(*) AS numSamples, h.
    commonName,
ROUND((COUNT(*)/o.numSeqs * 100), 2) AS percent
FROM OTU o
JOIN OTUSeqMapping osm ON o.otuID = osm.otuID
JOIN Sequence s ON osm.seqID = s.seqID
JOIN SeqSampleMapping ssm ON s.seqID = ssm.seqID
JOIN Sample sa ON ssm.sampleLabel = sa.sampleLabel
JOIN SampleToHost sth ON sa.sampleID = sth.sampleID
JOIN Host h ON sth.hostID = h.hostID
GROUP BY o.otuID, h.commonName;
```

(b) OTU To Host Species View Query

```
/* Dominant species by percent */
CREATE OR REPLACE VIEW v_MaxPercent AS
SELECT otuID, MAX(percent) AS purity
FROM v_OTUToSpecies
GROUP BY otuID;
```

(c) Max Percent OTU Species View Query

```
/* Average length of OTU in base pairs */
CREATE OR REPLACE VIEW v_OTUAvgLength AS
SELECT o.otuID,
ROUND(AVG(LENGTH(s.seqDNA))) AS otuAvgLength
FROM OTU o
JOIN OTUSeqMapping osm ON o.otuID = osm.otuID
JOIN Sequence s ON osm.seqID = s.seqID
JOIN SeqSampleMapping ssm ON s.seqID = ssm.seqID
GROUP BY o.otuID;
```

(d) Average Length of OTU View Query

Figure 4.8: LOTUS SQL View Query Statements. The four views shown are used for the default open OTU "branch". *De novo* views are created similarly using the Denovo "branch" tables.

order to keep the website functioning normally, these long-running tasks are run asynchronously in the background using the Python `celery` package. After the tasks are completed, an email notification is sent to the user indicating that the scripts have completed and the database is ready for review.

4.5.1 File Hierarchy

The file hierarchy necessary for C3PO file processing is implemented as designed in Figure 3.10. The web application stores uploaded files into the hierarchy as described. User files are organized per project and stored similarly whether for knowns or unknowns. A detailed illustration of the user project file tree is shown in Figure 4.9. In addition to the output files produced by C3PO, each user project contains a `run_params.txt` file which records the parameter configurations used to produce the output files and a `db_files.txt` file which is necessary to correctly load the processed data into the database.

4.5.2 User Management

The four user types outlined in Section 3.5.2 are implemented using restricted pages and navigation access. For guest level or higher access, users must register an account. Staff users must specifically request access when registering. An email is sent to the admin account requesting access which needs to be approved by the biologists. After approval, the admin can give staff permissions and notify the user by email.

The navigation bar enforces user permissions by displaying only the pages to which the user is permitted access as depicted in Figure 4.10. Further, access is restricted so that even if the page address is typed directly into the search bar, the user will be redirected to the login page if the user is not authorized.

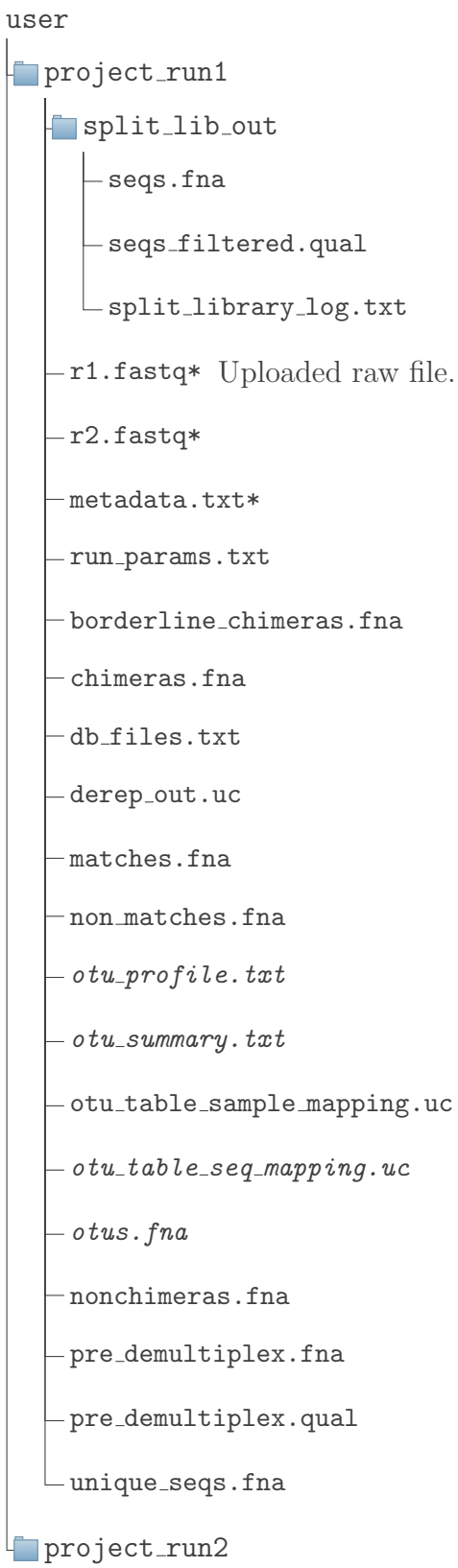


Figure 4.9: User files produced by C3PO. Starred files (*) are uploaded by user. *Italicized files* are only generated when used to build the library. The `run_params.txt` and `db_files.txt` files are generated separately by the web application.

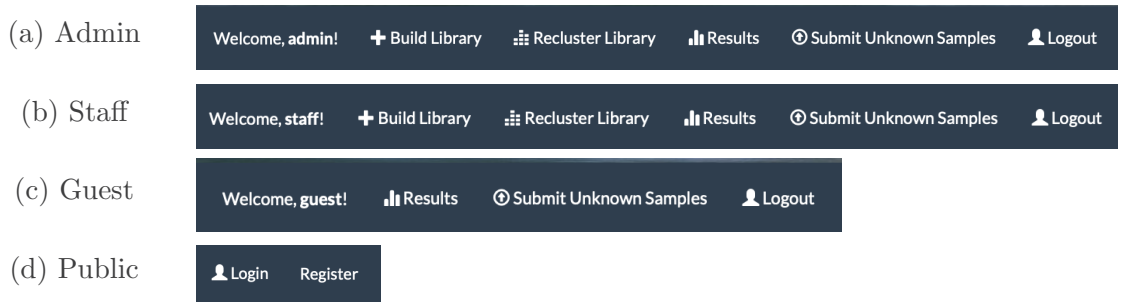


Figure 4.10: Navigation Bar Options for the four user types.

4.5.3 Requirements Implementation

This section gives an overview of the structure of the LOTUS web application, explaining common operations and the requirements fulfilled by the page design. The navigation bar headings are used as the names of the pages.

Home Page

The home or index page introduces users to LOTUS and allows browsing of the current library. This page has no user restrictions. Users can login or register an account.

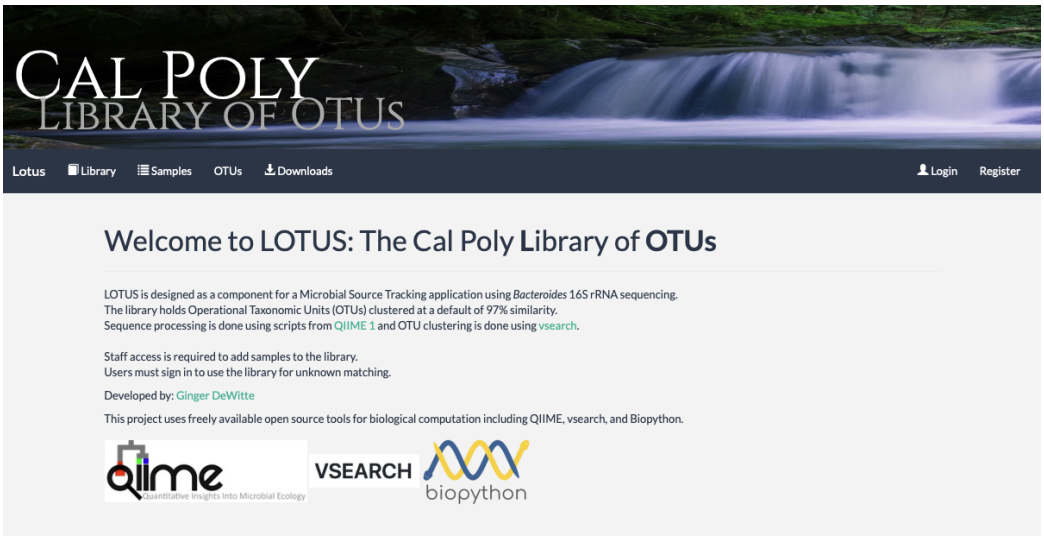


Figure 4.11: The Home Page

Library Page

The library page shown in Figure 4.12 is an informational page that displays a summary of the default library's current statistics. This page has no user restrictions and fulfills Requirement GR6. Users can click on the green button to see the statistics for the *De novo* picked OTUs shown in Figure 4.13.

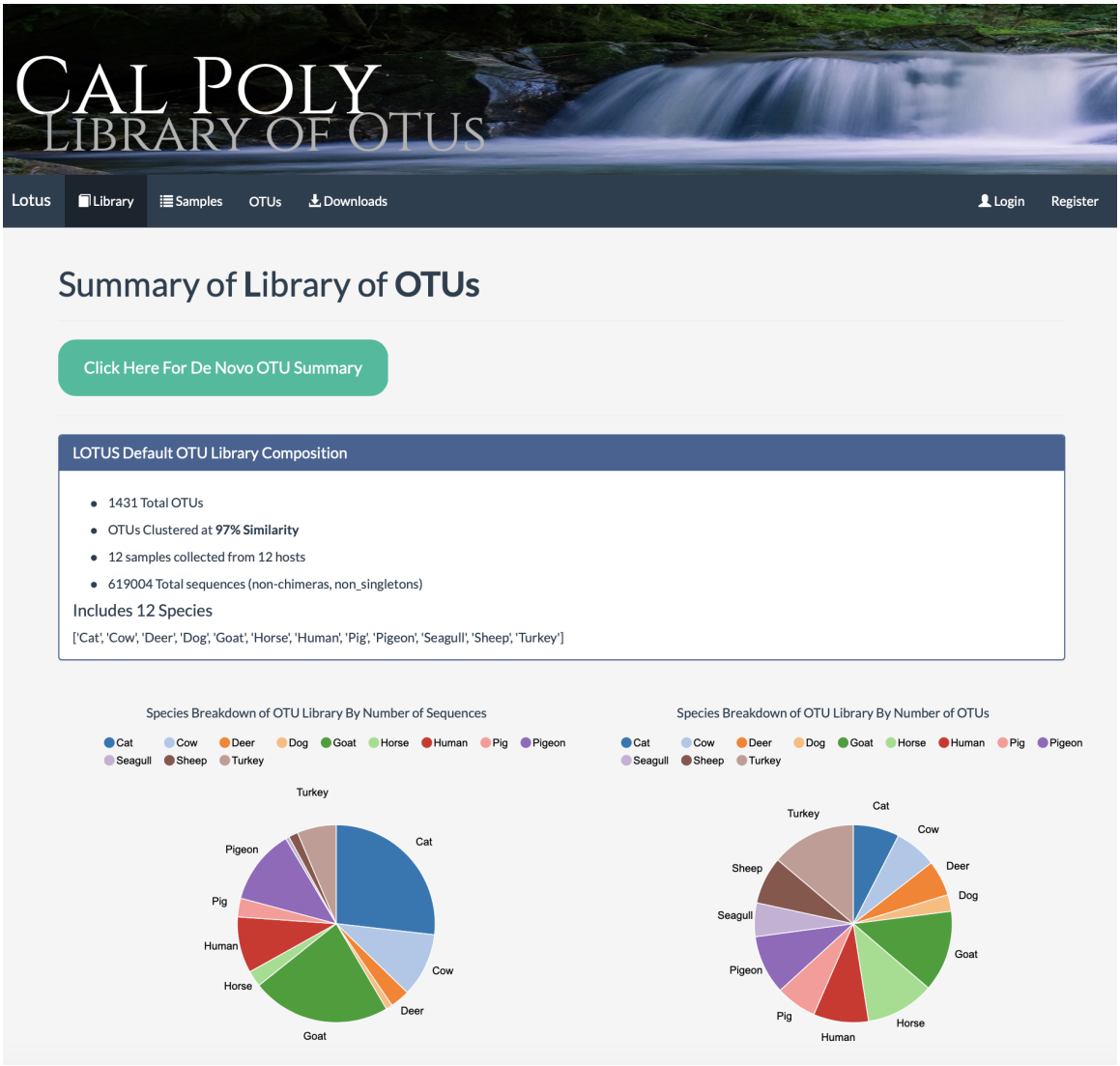


Figure 4.12: The Summary Page for Open OTUs in the Library

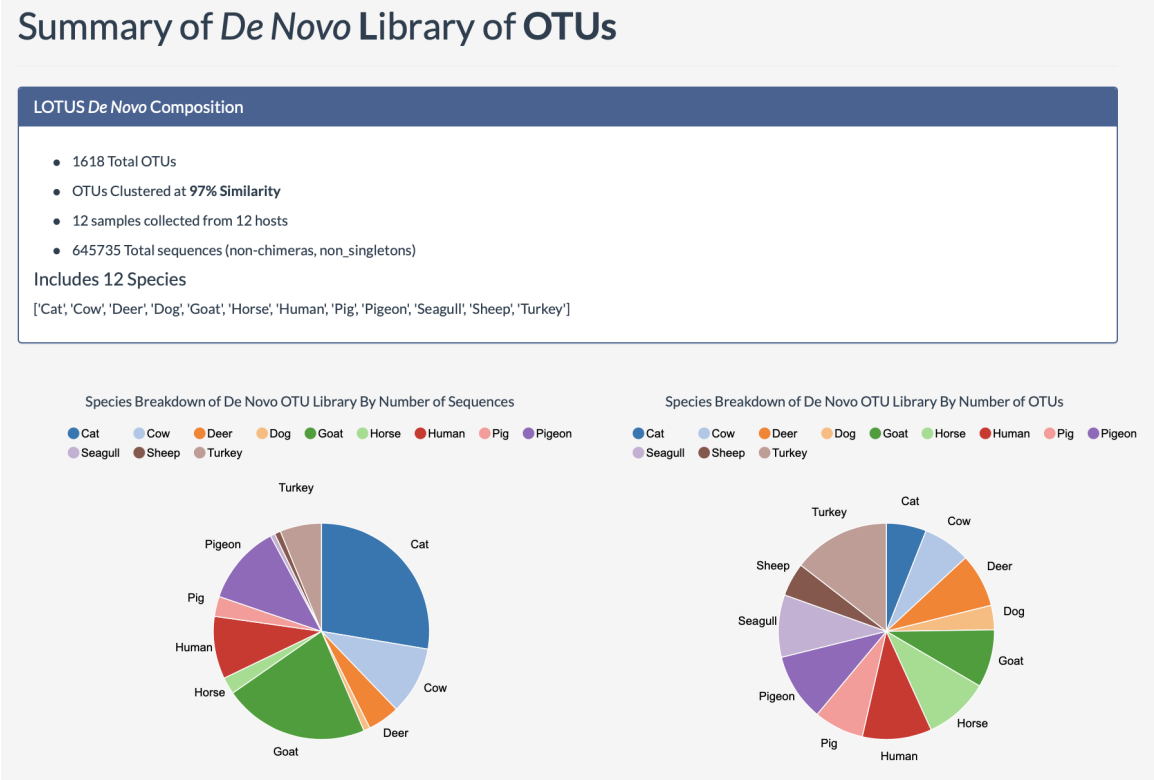


Figure 4.13: The Summary Page for *De novo* OTUs in the Library

Samples & Sample Detail Pages

The Samples page shown in Figure 4.14 fulfills Requirement GR1 and displays a list of the current samples used to create the database. Users can filter by different criteria or sort samples by clicking on the header name.

Clicking on the sampleID leads to the sample detail page which fulfills Requirement GR2. The sample detail page shown in Figure 4.15 gives detailed information about the sample and displays a list of the OTUs which include sequences from that sample. Clicking on the OTU label goes to the detail page for that OTU. Samples and Sample Details pages have no user restrictions.

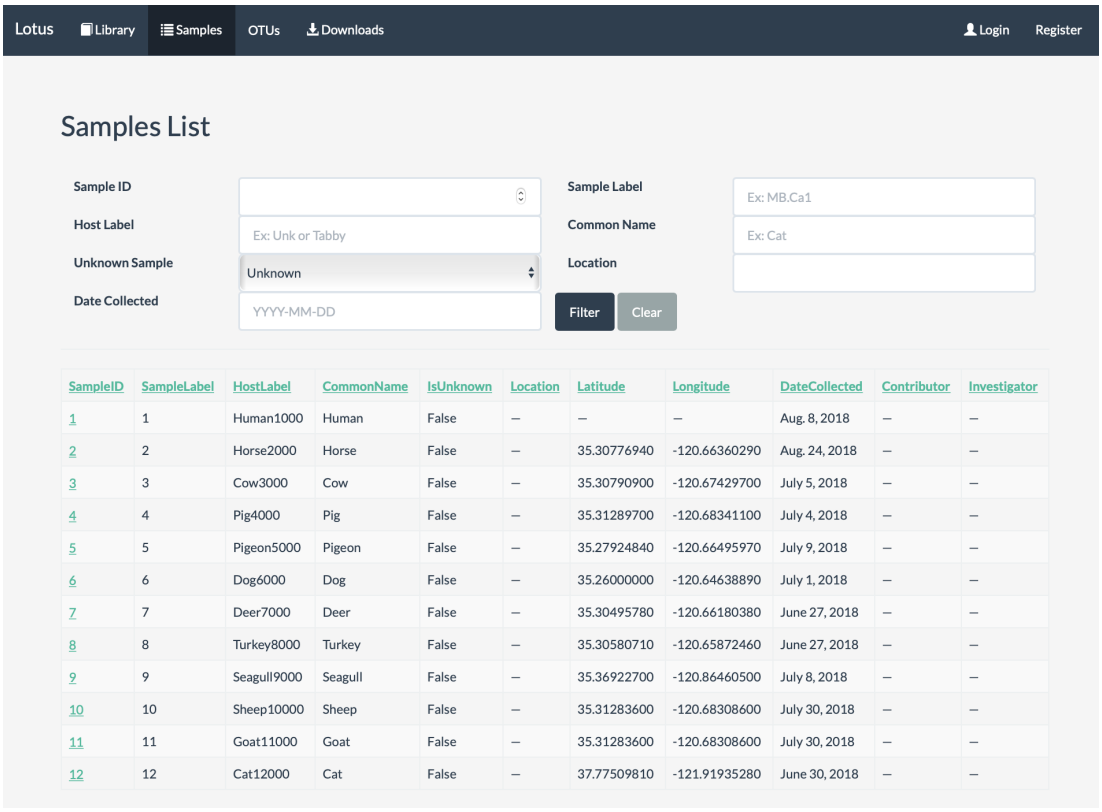


Figure 4.14: The Samples Page

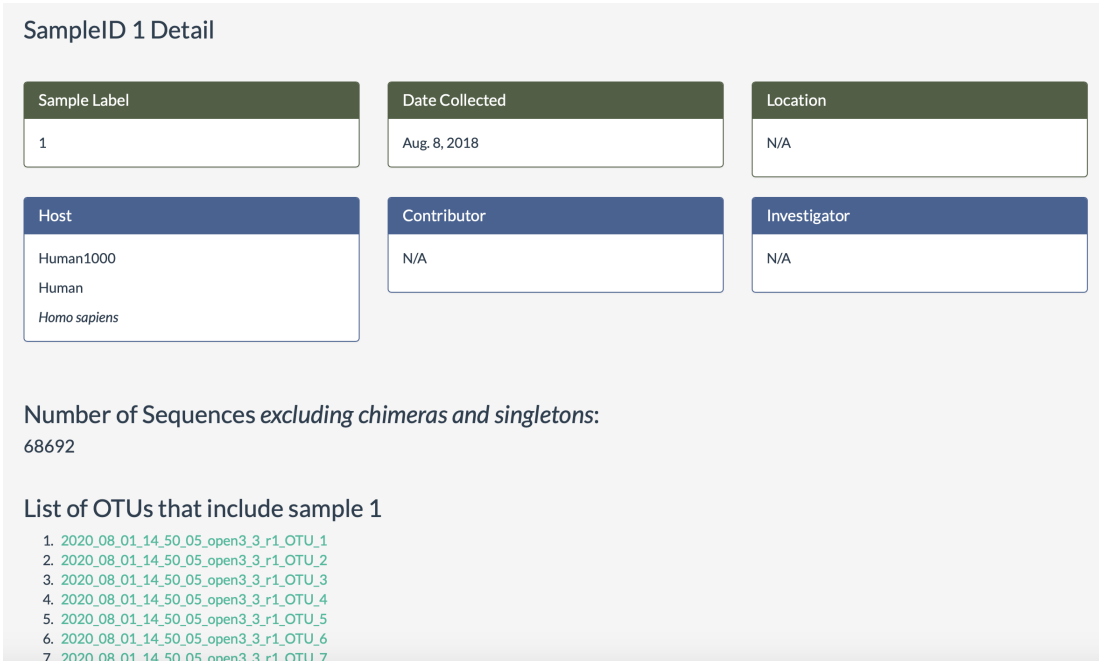
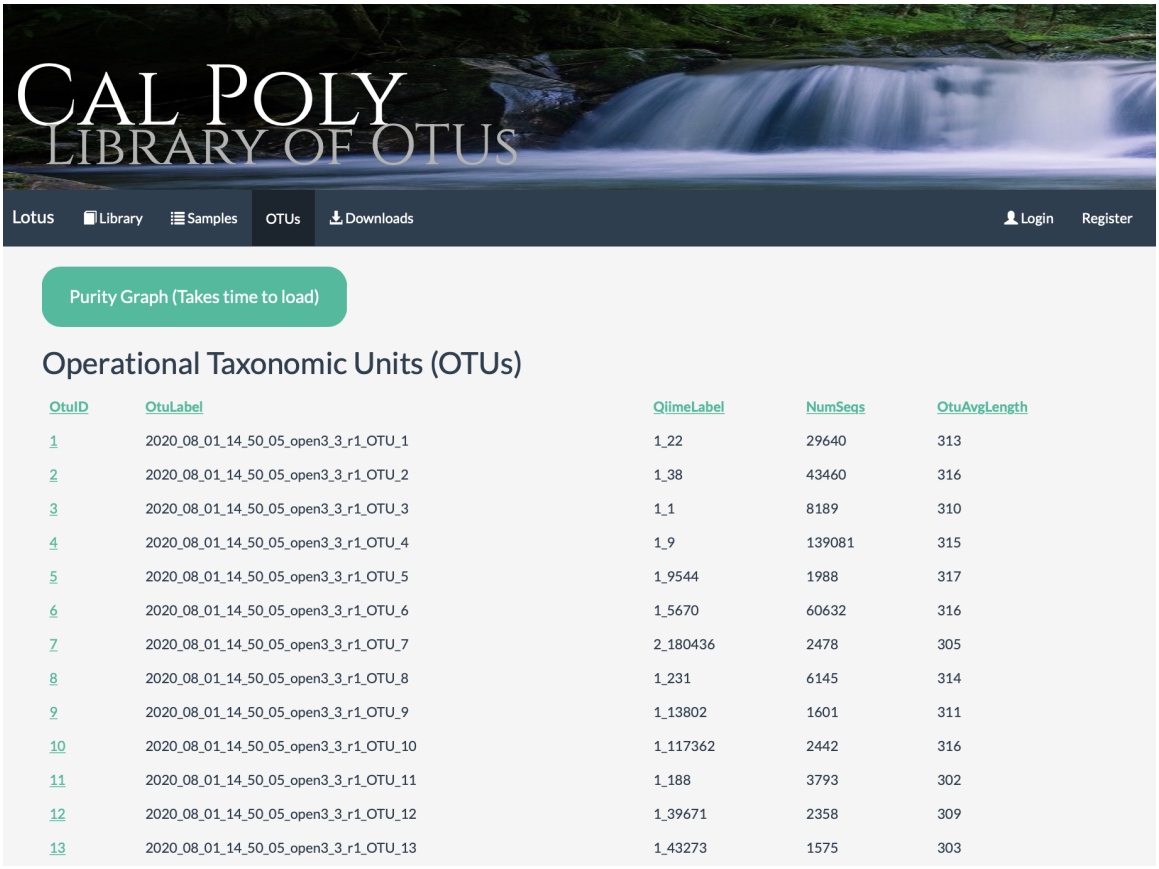


Figure 4.15: The Sample Detail Page

OTUs & OTU Detail Pages

The OTUs page shown in Figure 4.16 fulfills Requirement GR3 and shows a list of the current default (open picked) OTUs in the library. The list displays 50 OTUs per page. Users can sort OTUs by clicking on any of the header columns. Clicking on the green button leads to the OTU purity graphs shown in Figures 4.18 and 4.19.

Clicking on the OtuID goes to the OTU detail page which displays a summary of detailed information about that OTU including the centroid sequence the cluster is based on and the final taxonomic classification. The OTU detail page fulfills Requirement GR4 and is shown in Figure 4.17.



OtuID	OtuLabel	QiimeLabel	NumSeqs	OtuAvgLength
1	2020_08_01_14_50_05_open3_3_r1_OTU_1	1_22	29640	313
2	2020_08_01_14_50_05_open3_3_r1_OTU_2	1_38	43460	316
3	2020_08_01_14_50_05_open3_3_r1_OTU_3	1_1	8189	310
4	2020_08_01_14_50_05_open3_3_r1_OTU_4	1_9	139081	315
5	2020_08_01_14_50_05_open3_3_r1_OTU_5	1_9544	1988	317
6	2020_08_01_14_50_05_open3_3_r1_OTU_6	1_5670	60632	316
7	2020_08_01_14_50_05_open3_3_r1_OTU_7	2_180436	2478	305
8	2020_08_01_14_50_05_open3_3_r1_OTU_8	1_231	6145	314
9	2020_08_01_14_50_05_open3_3_r1_OTU_9	1_13802	1601	311
10	2020_08_01_14_50_05_open3_3_r1_OTU_10	1_117362	2442	316
11	2020_08_01_14_50_05_open3_3_r1_OTU_11	1_188	3793	302
12	2020_08_01_14_50_05_open3_3_r1_OTU_12	1_39671	2358	309
13	2020_08_01_14_50_05_open3_3_r1_OTU_13	1_43273	1575	303

Figure 4.16: The OTUs Page

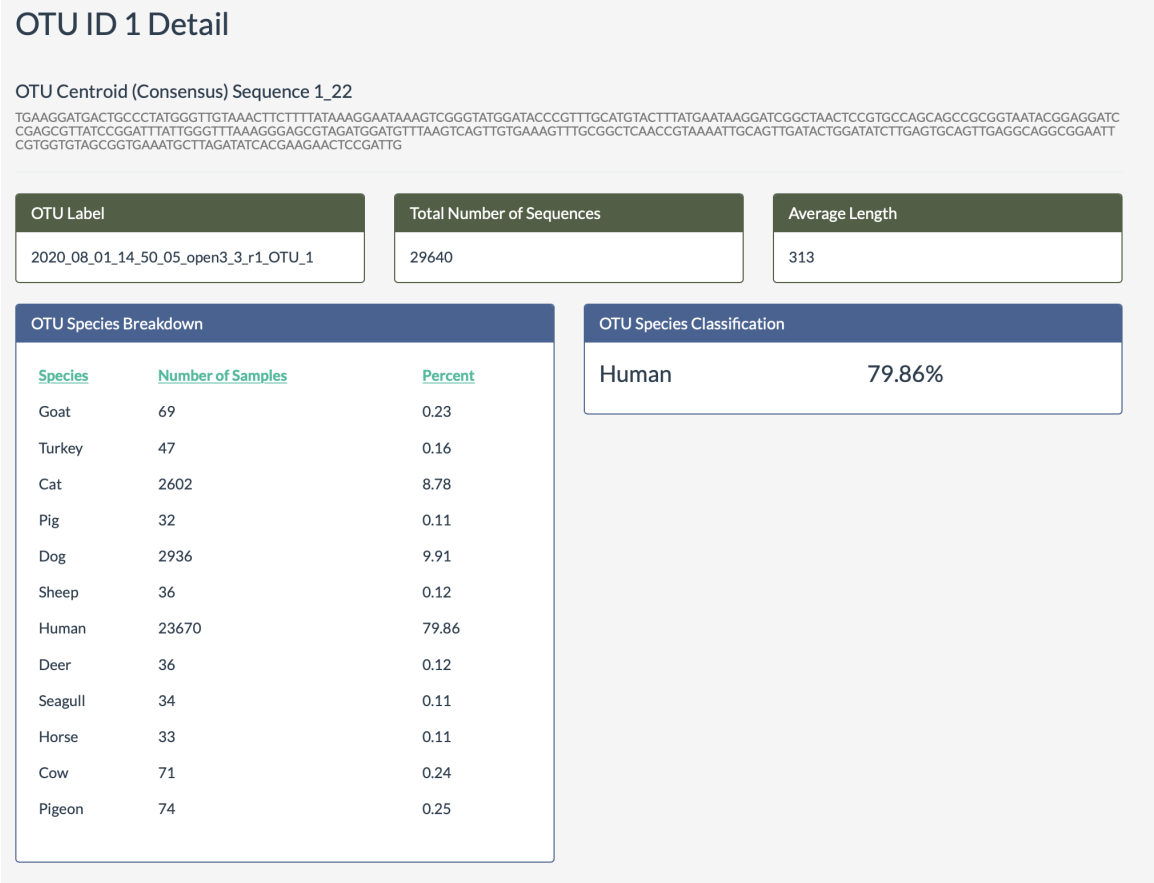


Figure 4.17: The OTU Detail Page

OTU Graphs Page

The OTU Purity Graph in Figure 4.18 displays a generalized view of the OTUs in the library and their taxonomic classification. This graph gives an overview of the species makeup of the OTUs in the library. Also included in the OTU Purity graph page are up to twelve individual species graphs which are depicted in Figure 4.19. The purity graphs fill Requirement GR5. None of the OTU display pages have user restrictions.

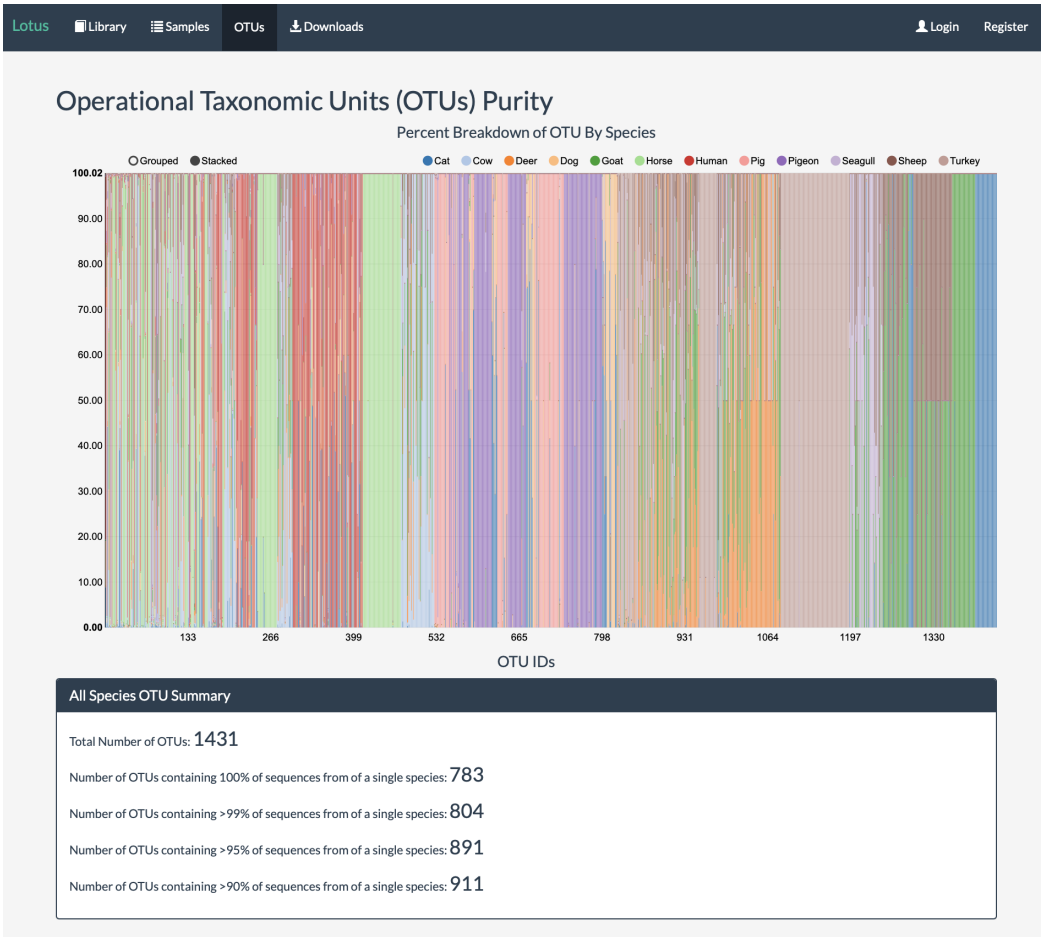


Figure 4.18: The OTU Purity Graph

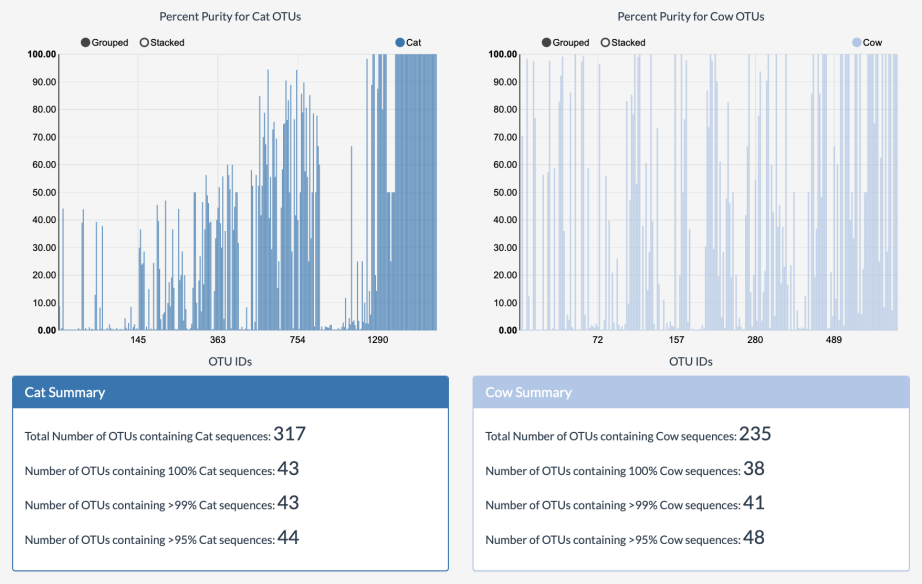


Figure 4.19: The OTU Purity Graphs By Species

Downloads Page

LOTUS provides a simplified interface seen in Figures 4.23 and 4.27 to allow users to upload project data. The only input files needed are: 1) a sample metadata file, and 2) the `fastq` files produced by the Illumina MiniSeq. The metadata file must be formatted in a specific manner to be parsed correctly by C3PO. The Downloads page shown in Figure 4.20 fulfills Requirement GR9 and provides a template for users to download and complete. The metadata file must be submitted as either a tab-separated value (.tsv or .txt) file or an Excel (.xlsx) file. There are three benefits to users submitting the raw sequencing data. First, it saves researchers time by allowing users to forgo their own data manipulation. Second, it standardizes the data processing across multiple users to create consistency. Third, the inclusion of the rawest data allows re-access for use in future iterations that can include any experimental changes to either C3PO or the database.

Figure 4.20: The Downloads Page

User Authentication (Register & Login Pages)

The Django framework provides pre-built user authentication pages including registration, login, and forgot password. The pages fill Requirement GR8. The registration page was modified to include a checkbox for requesting staff access to the library. The Register page is shown in Figure 4.21 and the Login page in Figure 4.22.

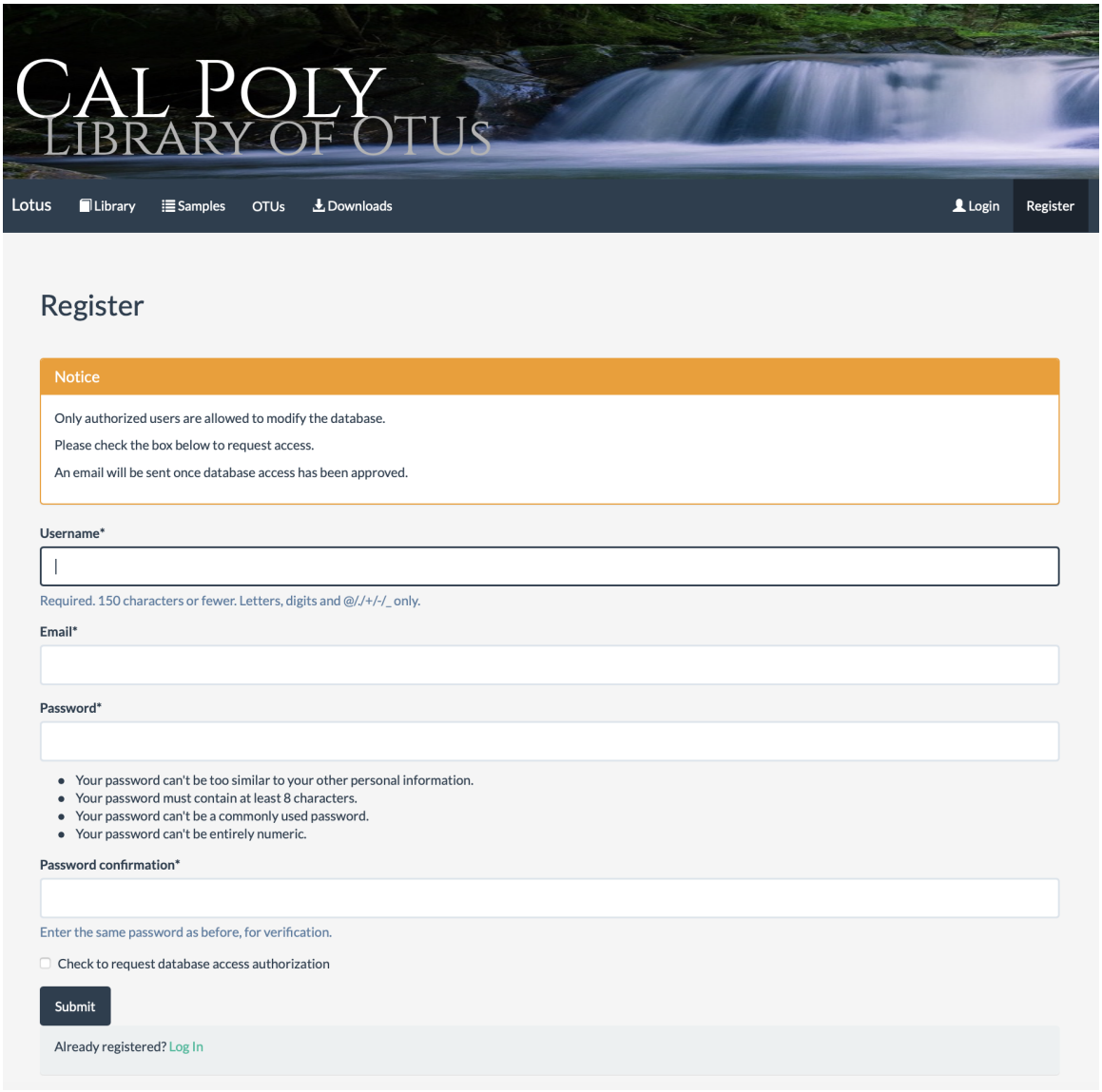


Figure 4.21: The Register Account Page

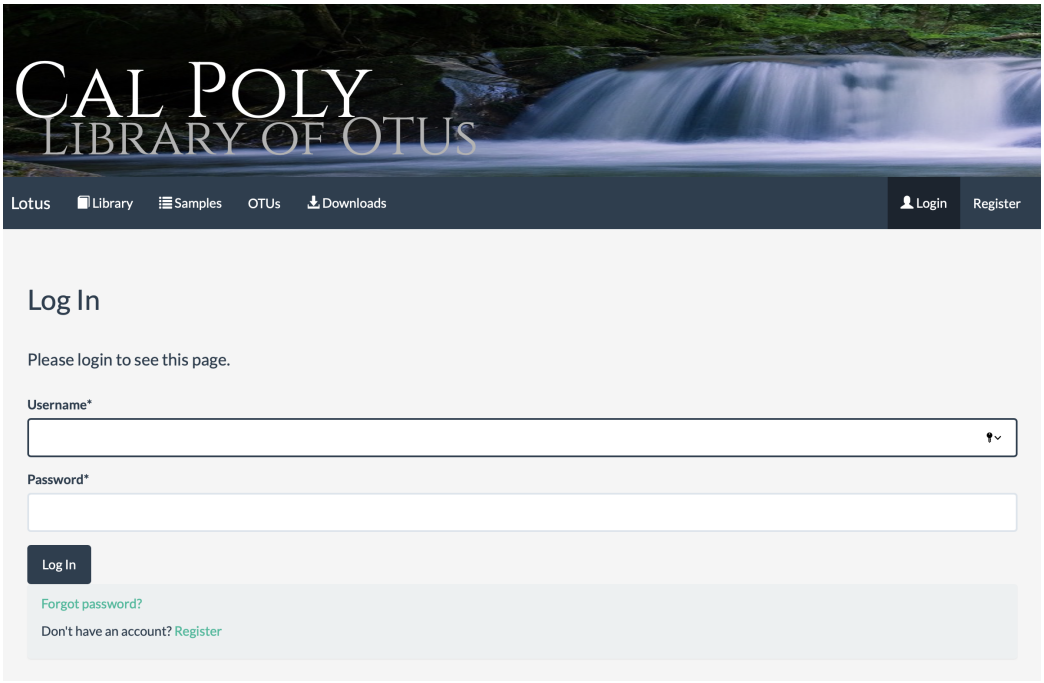


Figure 4.22: The Login Page

Build Library Page

The Build Library page fulfills Requirements BR1 - BR6. Only staff and admin users can access this page. Upon submission the files are loaded to the appropriate user folder as described in the file hierarchy and are processed in the correct C3PO pipeline. As these scripts take a significant amount of time to process, the web application offloads these to asynchronous tasks in order to allow the user to continue browsing through the website. An email notification is sent to the user upon completion of the data processing and loading into the database. The Build Library page is seen in Figure 4.23 and the submission confirmation in Figure 4.24.

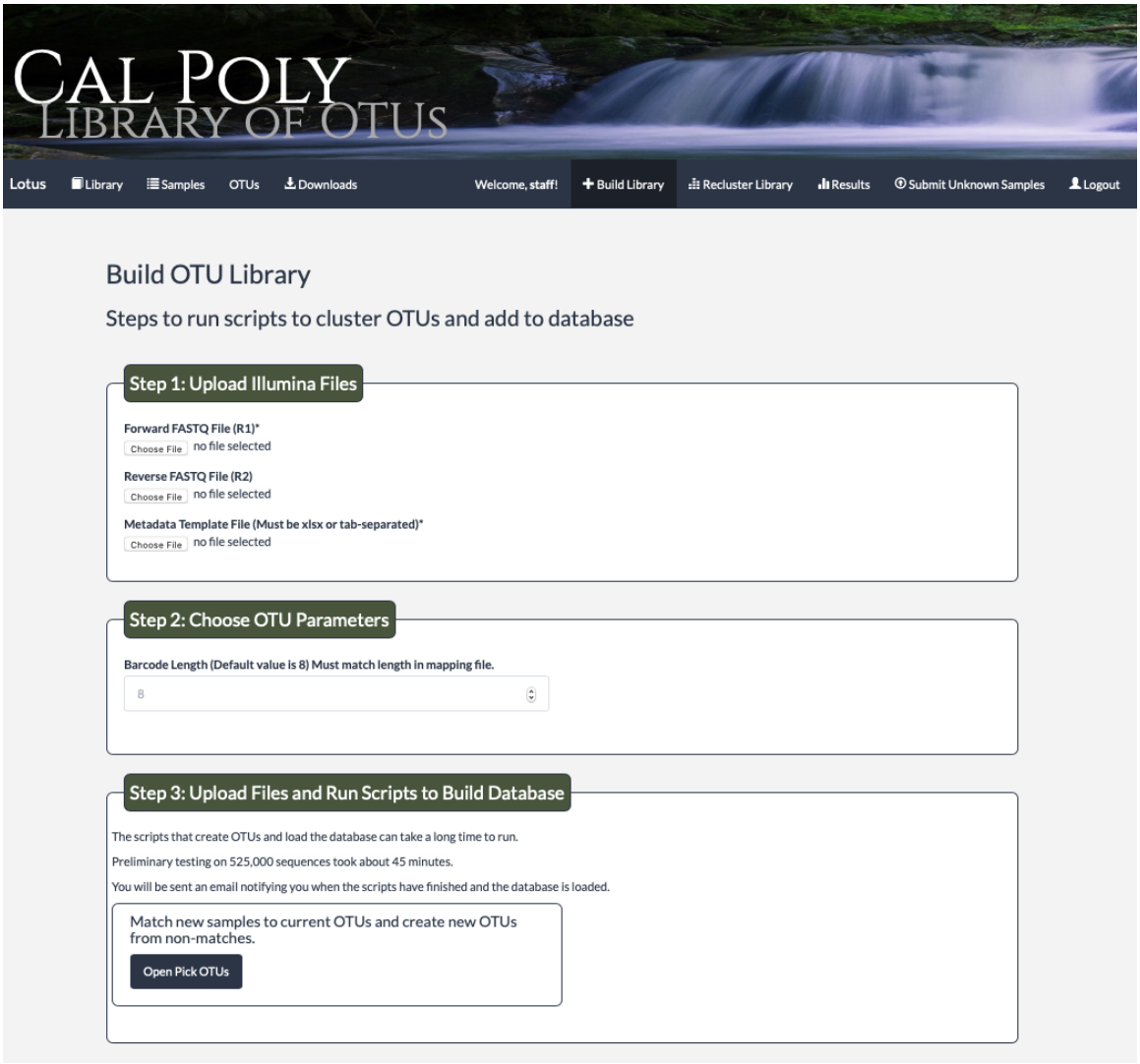


Figure 4.23: The Build Library Page

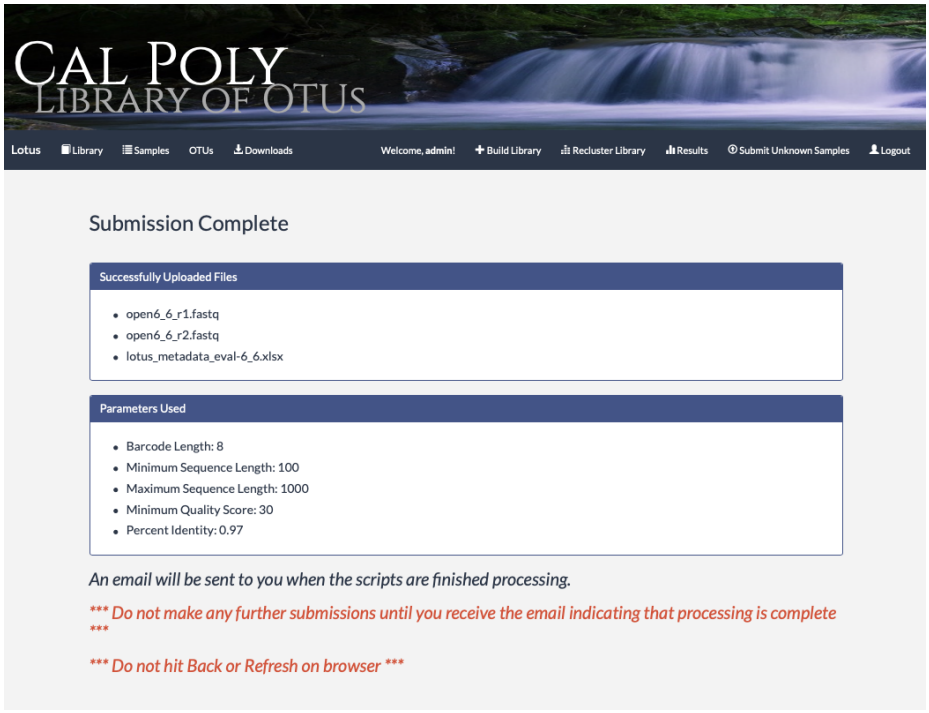


Figure 4.24: The Build Library Submission Confirmation Page

Recluster Library Page

The Recluster Library page fulfills Requirements RR1 - RR5. Only staff and admin users can access this page. Users are warned that reclustering the database loses all previous information for the *De novo* database. Users can alter the parameters such as minimum sequence length or percent identity to experiment with different types of OTUs produced by different parameters. There is no uploading of new files as the recluster command only works on the previously uploaded and processed files. As with regular submission, the reclustering scripts take a significant amount of time to process, so the web application offloads these to asynchronous tasks in order to allow the user to continue browsing through the website. An email notification is sent to the user upon completion of the data processing and loading into the database. The Recluster Library page is seen in Figure 4.25 and the confirmation in Figure 4.26.

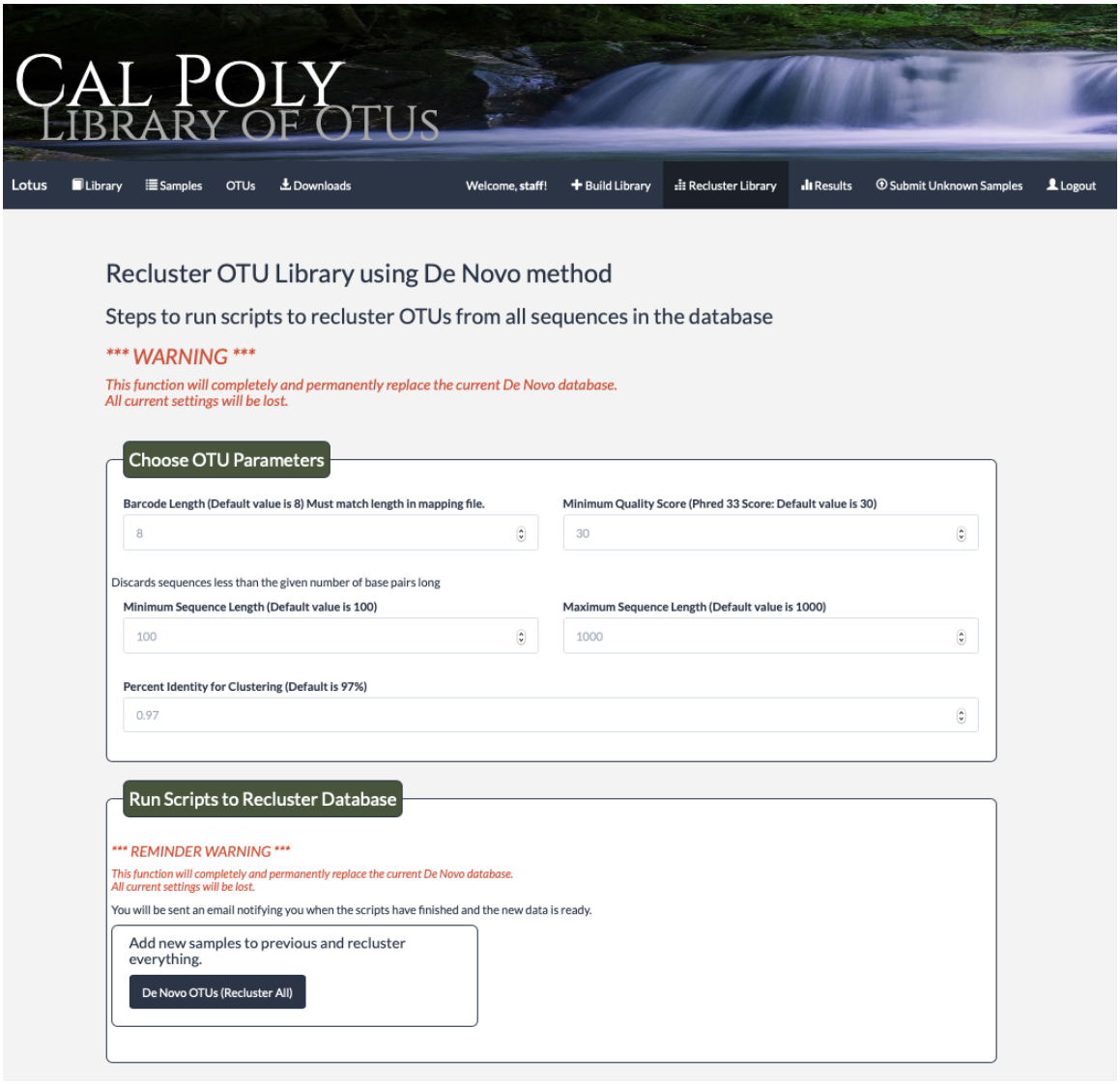


Figure 4.25: The Recluster Library Page

Figure 4.26: The Recluster Library Confirmation Page

Submit Unknowns Page

The Submit Unknowns page fulfills Requirements MR1 - MR5. Users must have an account to access this page. Upon submission the files are loaded to the appropriate user folder as described in the file hierarchy and are processed in the correct C3PO pipeline. As these scripts take a significant amount of time to process, the website offloads these to asynchronous tasks in order to allow the user to continue browsing through the website. An email notification is sent to the user upon completion of the data processing. The Submit Unknowns page is seen in Figure 4.27 and the confirmation page in Figure 4.28.

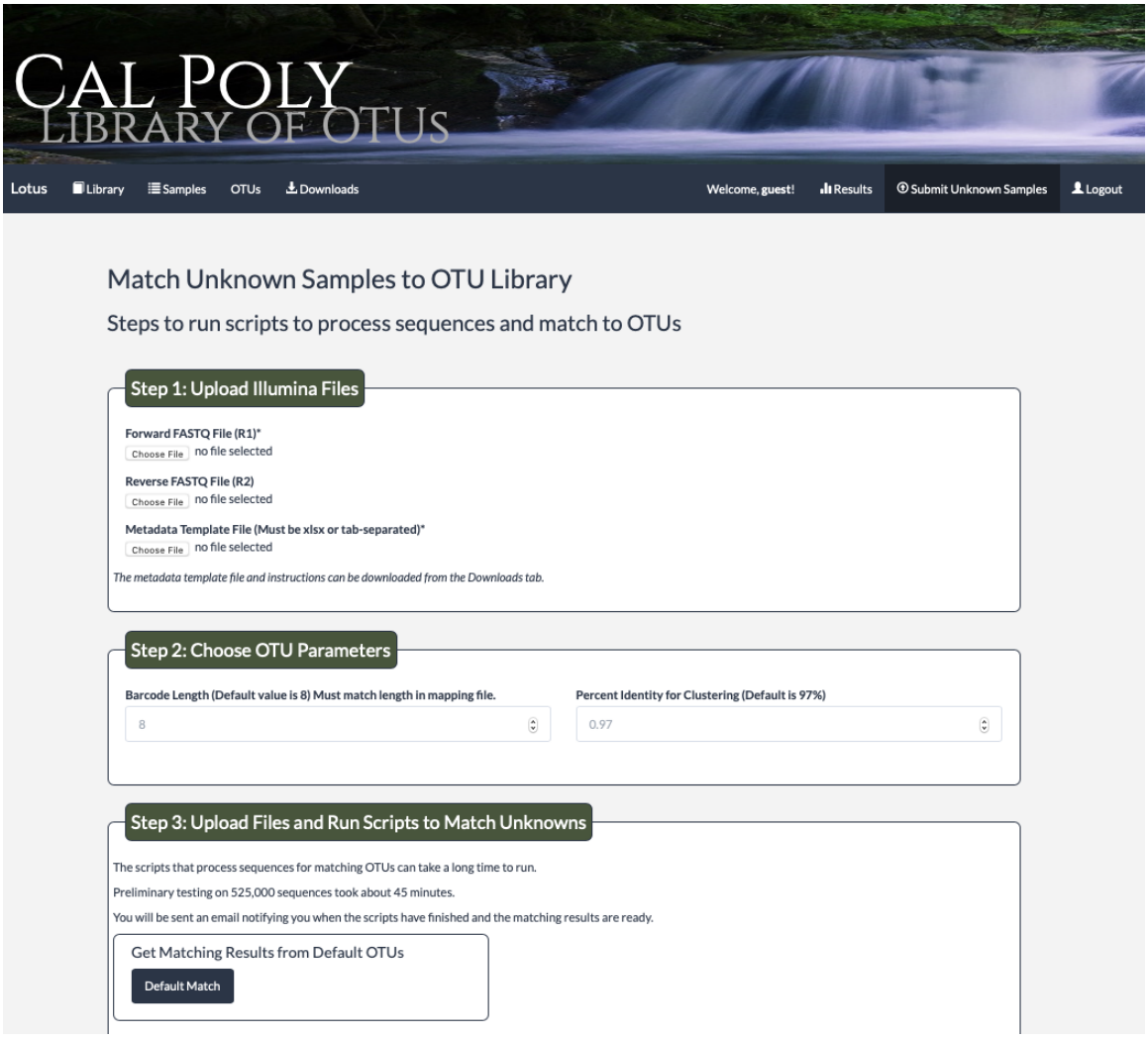


Figure 4.27: The Submit Unknowns Page

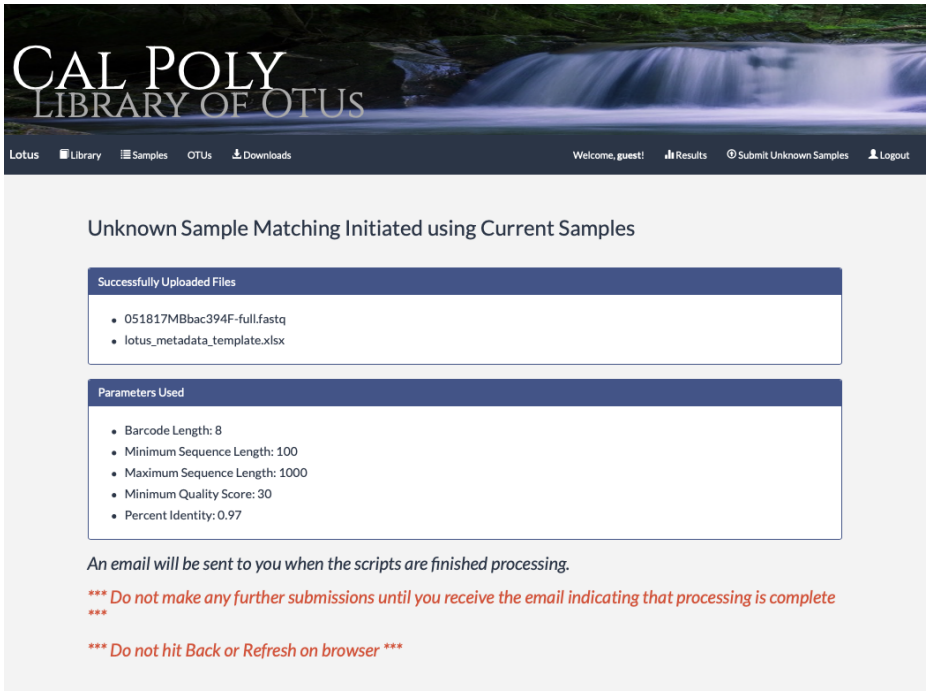


Figure 4.28: The Submit Unknowns Confirmation Page

Results Page

The Results page fulfills Requirement MR6. Logged in users can select from a list of projects they have submitted as in Figure 4.29. Once a project is selected, the matching results for user submitted environmental samples are displayed for review. The Results page is seen in Figure 4.30.

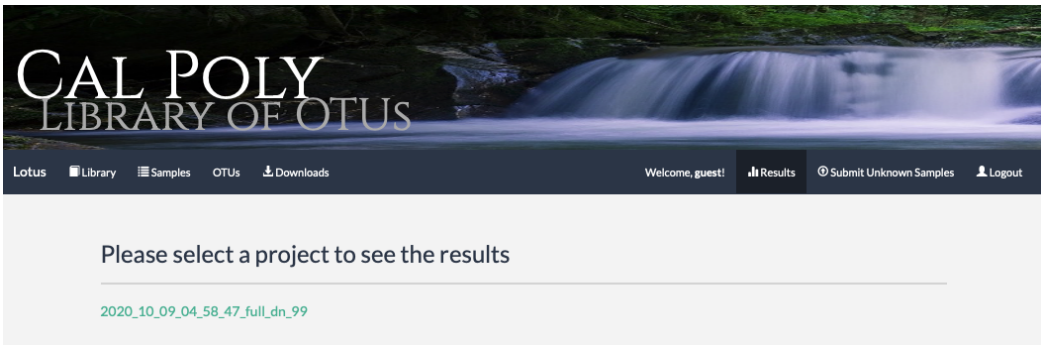


Figure 4.29: The Results Index Page which lists User Projects

Figure 4.30: The Results Page for a given project

Summary of Requirement Tracing

All the pages discussed above are the implementation of the web application component of the OBMM. A summary of the requirements are depicted in Table 4.1.

Table 4.1: Summary Table of LOTUS Requirements Tracing

Requirement ID	Requirement Title	Fulfilled By	Reference Figure
GR1	Sample: General Information	Samples Page	Figure 4.14
GR2	Sample: Detailed Information	Sample Detail Page	Figure 4.15
GR3	OTU: General Information	OTUs Page	Figure 4.16
GR4	OTU: Detailed Information	OTU Detail Page	Figure 4.17
GR5	OTU Graphical Summary	OTU Purity Graph Page	Figures 4.18 – 4.19
GR6	Library Summary	Library / Denovo Summary Page	Figures 4.12 – 4.13
GR7	Simplified Interface	All pages	Figures 4.11 – 4.30
GR8	User Management	Login Page/Nav Bar/Restricted Access	Figures 4.10, 4.21 – 4.22
GR9	Downloadable Templates	Downloads Page	Figure 4.20
BR1	Restricted Access	Build Library Page/Nav Bar	Figures 4.10, 4.23
BR2	Data Upload	Build Library Page	Figure 4.23
BR3	Create OTUs	C3PO (open picking)	Figures 4.3, 4.23 – 4.24
BR4	Load Database	Web app asynchronous task	Figure 4.23
BR5	User Project File Storage	File Hierarchy	Figures 4.9, 4.23
BR6	User Notification	Web app asynchronous task	Figure 4.23
RR1	Restricted Access	Recluster Library Page/Nav Bar	Figure 4.10, 4.25
RR2	Optional Parameters	Recluster Library Page	Figure 4.25
RR3	Recluster *De novo* Database	C3PO (*de novo* picking)	Figures 4.4, 4.25
RR4	Store Configuration	Web app/File Hierarchy	Figures 3.10, 4.25
RR5	User Notification	Web app asynchronous task	Figure 4.25
MR1	Restricted Access	Submit Unknown Page/Nav Bar	Figures 4.10, 4.27
MR2	Data Upload	Submit Unknown Page	Figure 4.27
MR3	User Project File Storage	File Hierarchy	Figures 4.9, 4.27
MR4	Match Unknowns	C3PO (unknown matching)	Figures 4.5, 4.27 – 4.28
MR5	User Notification	Web app asynchronous task	Figure 4.27
MR6	View Matching Results	Results Page	Figures 4.29 – 4.30
MR7	Download Matching Results	Unfulfilled	

Chapter 5

EVALUATION

The evaluation of LOTUS occurred at various stages of OBMM development. Overall, there are five questions to answer to lay the groundwork for the OBMM as a viable MST method and how LOTUS contributes to this effort:

1. Can OTUs be used as molecular signatures for specific hosts? In other words, can an OTU be used to identify a species as a source of contamination?
2. Are LOTUS-generated OTUs of comparable quality to industry standard OTUs created by an outside laboratory?
3. Are OTUs useful for source tracking at different sequence trim lengths considering the MiniSeq only produces reads up to 150 bp long?
4. Which method of LOTUS OTU picking (*De novo* vs Open OTUs) produces higher quality (more single-source) OTUs for source tracking?
5. How accurate is LOTUS unknown matching?

All evaluation tests were run on a 2017 MacBook Pro laptop running macOS Mojave 10.14.6 with 16GB RAM and 1TB storage using a 2.9 GHz Intel Core i7 Processor. The web application was tested using the Safari web browser.

5.1 Evaluation Metrics

The goal of LOTUS is to create successful OTUs as defined by the requirements of the OBMM for source tracking. In this thesis, a successful OTU is therefore a single-

source OTU that can be used to trace back to a single host species. The quantitative metrics used to evaluate this objective are discussed below.

5.1.1 Purity

Purity can be formally defined as a measure of the extent to which a cluster contains objects of a single class [147]. Put in terms of this thesis, purity is the measure of the extent to which an OTU contains sequences of a single species. The purity cutoff threshold is what defines single-source OTUs. For an individual OTU cluster c, purity is represented by the formula:

$$purity_c = \max_s(P_{cs}) \tag{5.1}$$

where P_{cs} is the class distribution of the data or the probability that a sequence of OTU cluster c belong to species s. This probability is computed by the formula:

$$P_{cs} = \frac{q_{cs}}{q_c} \tag{5.2}$$

where q_{cs} is the number of sequences of species s in OTU cluster c and q_c is the total number of sequences in OTU cluster c.

The total weighted purity for all OTUs in the library is:

$$purity_{total} = \sum_{c=1}^{K} \frac{q_c}{q} * purity_c \tag{5.3}$$

where K is the number of OTU clusters in the library, q_c is the total number of sequences in OTU cluster c, q is the total number of sequences in all the OTUs in the library, and $purity_c$ is the purity of cluster c. Figure 5.1 shows how to calculate purity.

OTU	Cat	Dog	Human	Total	Purity
1	0	14	0	14	1.0
2	4	3	5	12	0.417
3	26	0	1	27	0.963
Total	30	17	6	53	0.849

Figure 5.1: A simple example using three species showing how to calculate the purity of an individual OTU cluster and the total purity of all the clusters in the "library". The purity can be calculated as shown:

$purity_1 = \max(\frac{14}{14}) = \frac{14}{14} = 1$

$purity_2 = \max(\frac{4}{12}, \frac{3}{12}, \frac{5}{12}) = \frac{5}{12} = 0.417$

$purity_3 = \max(\frac{26}{27}, \frac{1}{27}) = \frac{26}{27} = 0.963$

$purity_{total} = 1 * \frac{14}{53} + 0.417 * \frac{12}{53} + 0.963 * \frac{27}{53} = 0.849$

5.1.2 Entropy

Entropy is formally defined as the degree to which each cluster consists of objects of a single class, and essentially represents the disorder of a cluster [147]. Again, in terms of this thesis, entropy is the degree to which each OTU consists of sequences of a single species. For a single OTU cluster c, entropy is represented by the equation:

$$entropy_c = -\sum_{s=1}^{k} P_{cs} * log_2 P_{cs} \tag{5.4}$$

where k is the number of species and P_{cs} is the probability that a sequence of OTU cluster c belong to species s as defined in equation 5.2.

The total weighted entropy for all OTUs in the reference library is calculated as the sum of the entropies of each individual OTU weighted by the size of each OTU

[147]:

$$entropy_{total} = \sum_{c=1}^{N} \frac{q_c}{q} * entropy_c \tag{5.5}$$

where N is the number of OTUs in the library, q_c is the total number of sequences in OTU cluster c, q is the total number of sequences in all the OTUs in the library, and $entropy_c$ is the entropy of cluster c.

Entropy measures the disorder in a cluster. The smallest possible value of entropy is 0 and indicates no disorder (e.g., an OTU made only of sequences from the same species). The highest possible entropy value results from an even distribution of all species (i.e., the most impure cluster contains an equal number of sequences from all 12 species). Entropy ranges from 0 to $log_2 k$ where k is the number of classes available. The reference library being evaluated contains 12 species ($k = 12$), thus entropy ranges from 0 to 3.58496. Figure 5.2 demonstrates how to calculate entropy.

OTU	Cat	Dog	Human	Total	Entropy
1	0	14	0	14	0.0
2	4	3	5	12	1.555
3	26	0	1	27	0.229
Total	30	17	6	53	0.468

Figure 5.2: A simple example using three species showing how to calculate the entropy of an individual OTU cluster and the total entropy of all the clusters in the "library". For three classes, the entropy ranges from 0 to 1.585. The entropy can be calculated as shown:

$entropy_1 = -\frac{14}{14} * log_2 \frac{14}{14} = 0$

$entropy_2 = -\frac{4}{12} * log_2 \frac{4}{12} - \frac{3}{12} * log_2 \frac{3}{12} - \frac{5}{12} * log_2 \frac{5}{12} = 1.555$

$entropy_3 = -\frac{26}{27} * log_2 \frac{26}{27} - \frac{1}{27} * log_2 \frac{1}{27} = 0.229$

$entropy_{total} = 1.555 * \frac{12}{53} + 0.229 * \frac{27}{53} = 0.468$

5.1.3 Accuracy

The effectiveness of matching sequences from unknown sources to OTUs is measured using two accuracy values: one for the overall total accuracy and one for the accuracy as it pertains to different purity levels. The total accuracy is calculated as follows:

$$accuracy_t = \frac{q_n}{q_u} \tag{5.6}$$

where q_n is the number of correctly classified sequences in an unknown sample and q_u is the total number of sequences in an unknown sample.

The more focalized purity accuracy reflects the matching accuracy of the available OTUs at a given purity cutoff, giving the accuracy using only single-source OTUs. The purity accuracy is calculated as:

$$accuracy_p = \frac{q_n}{q_p} \tag{5.7}$$

where q_n is the number of correctly classified sequences in an unknown sample and q_p is the total number of sequences in an unknown sample minus the number of non-matching sequences.

5.2 Feasibility Test of OTUs as Molecular Signatures

The initial question that must be answered for the OBMM is: Can OTUs be used as molecular signatures to identify specific hosts? Phrased another way, Are there species-specific OTUs? And the follow-up question is: Do species-specific OTUs occur in enough abundance to be useful for MST? The premise of OTU investigation is that OTUs are clusters of similar sequences that are phylogenetically related. This premise is evaluated using OTU purity.

Although the plan for the OBMM is to use the Illumina MiniSeq for sequencing, Cal Poly had not yet acquired the platform at the time of the initial investigation.

Table 5.1: Sample data sent to MR_DNA. The number of sequences produced is from MR_DNA's proprietary processing pipeline.

Sample Label	Species	Number of Sequences
MB.Ca1	Cat	114302
MB.Do1	Dog	122591
MB.Ho1	Horse	87961
MB.Hu1	Human	118512
MB.Hu2	Human	122074
Total	4	565440

Instead, researchers sent five samples to MR_DNA[1], an outside genetic laboratory, to perform sequencing and preliminary OTU analysis. MR_DNA uses the Illumina MiSeq system which, like the MiniSeq, is designed to support targeted sequencing studies. The MiSeq can output up to 15Gb of data and can produce longer reads (up to 300bp) than the MiniSeq [67].

The five samples sent to MR_DNA are shown in Table 5.1. Four different species were represented as two of the samples were human. In total, MR_DNA produced 565,440 sequences which were clustered into 379 OTUs. The laboratory-generated OTUs were created using the process discussed in Section 3.3.1. MR_DNA analysis files included mapping of sequences to OTUs which were loaded into a MySQL database in order to calculate purity as described in Section 5.1.1.

The purity results are shown in the graph in Figure 5.3. The graph plots the natural log of the size of the OTU against the purity of the OTU. The size of the OTU is determined by the number of sequences present in the OTU cluster ranging from 2 to over 100,000. The natural log scale helps with visualizing the large discrepancies

[1] `https://www.mrdnalab.com`, MR_DNA, Shallowater, TX, USA

in this data range. This graph shows the purity of OTU clusters of all sizes. The largest OTU clusters (made up of over 100,000 sequences) are around 50% purity. Most of the OTUs clusters are above 90% purity regardless of size. A graph focusing on OTUs with 90% purity or higher is seen in Figure 5.4. At a 90% purity cutoff, this graph shows that single-source OTUs are distributed across all cluster sizes.

In all, 309 out of the 379 OTUs were single-source given a 90% purity cutoff, and these included OTUs from all 4 species tested. This result is a positive indicator that OTUs can in fact be used as molecular signatures for source tracking.

5.3 Comparison of OTUs: MR_DNA vs LOTUS

One of the biologists' requests was an in-house pipeline for creating OTUs from raw sequencing data. The goal is to use the Illumina MiniSeq at Cal Poly rather than sending samples to an outside lab. Once it was established that OTUs are species-specific for MR_DNA-generated OTUs, the next step in evaluation was to determine if LOTUS could produce comparable quality OTUs to MR_DNA.

Using the same five samples from Table 5.1, C3PO produced 516,943 sequences and 117 *de novo* OTUs. LOTUS did not have a reference database and *de novo* OTU picking had to be used for the initial clustering. It is not stated what OTU picking method MR_DNA used. The differences in parameter choices most likely account for the difference in number of sequences and OTUs between LOTUS and MR_DNA (e.g., quality filtering in LOTUS is set to 30 and is unknown in MR_DNA). As seen in Table 5.2 and visualized in Figures 5.5 and 5.6, C3PO produces a slightly higher percentage of single-source OTUs at different purity thresholds, meaning that C3PO is an acceptable alternative to an outside laboratory for producing OTUs from raw sequencing data for the OBMM.

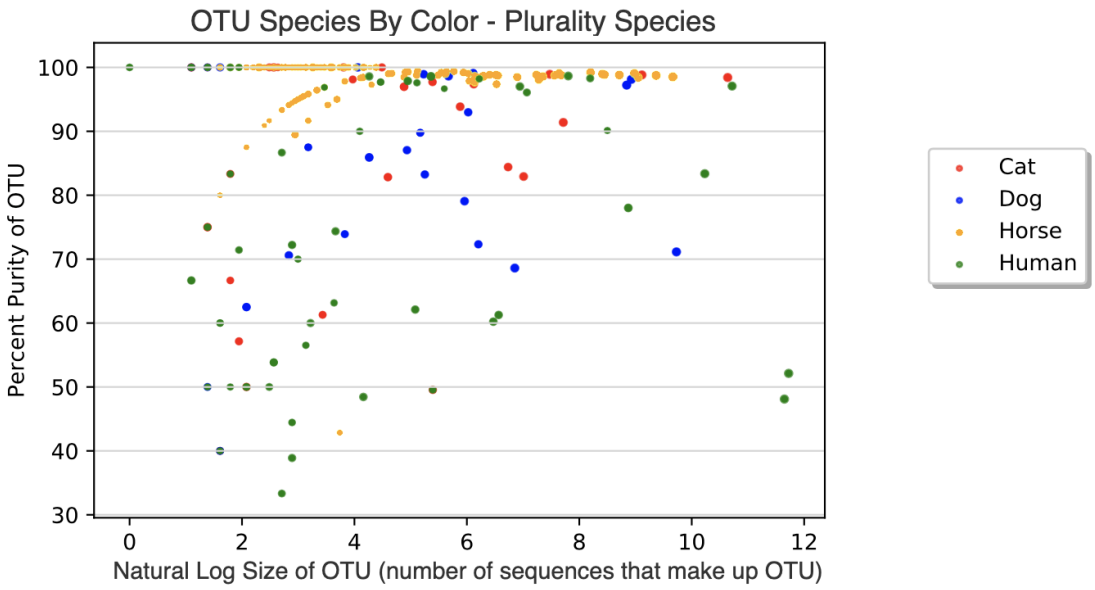


Figure 5.3: Plot of the natural log of the size of OTUs against the percent purity of the OTUs with no purity threshold. All OTUs were produced by MR_DNA. As there was no purity cutoff in this graph, there are no single-source OTUs and the species with the most frequent sequences in the OTU was the taxonomically assigned plurality species for the OTU. The taxonomic assignment of each OTU is shown by the color-coded species legend.

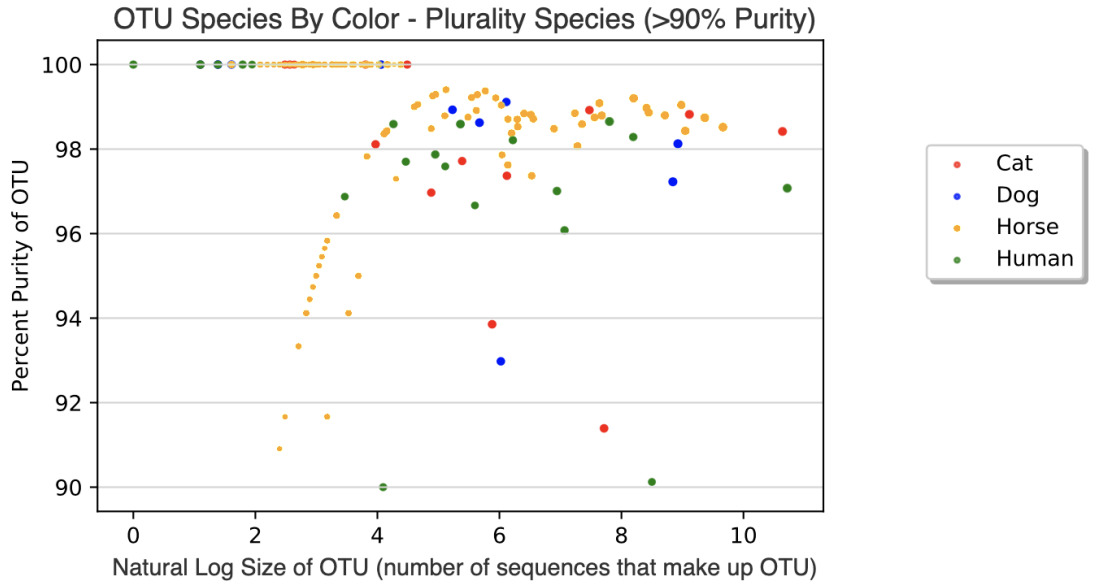


Figure 5.4: Plot of the natural log of the size of OTUs against the percent purity of the OTUs with a purity threshold of 90%. All OTUs were produced by MR_DNA. The 90% cutoff means that every OTU in this graph is a single-source OTU. The taxonomic assignment of each OTU is shown by the color-coded species legend.

Table 5.2: A table comparing OTUs created by MR_DNA and by C3PO. Three different purity cutoffs were used in this assessment. LOTUS has fewer numbers of OTUs, but similar percentages to MR_DNA. For example, 309 out of 379 (81.53%) of MR_DNA OTUS are >= 90% pure, and 99 out of 117 (84.62%) of LOTUS OTUs are >= 90% pure.

Source	Total Seqs	Total OTUs	90% purity		95% purity		99% purity	
			Number	%	Number	%	Number	%
MR_DNA	565440	379	309	81.53	294	77.57	229	60.42
C3PO	377241	117	99	84.62	95	81.2	75	64.10

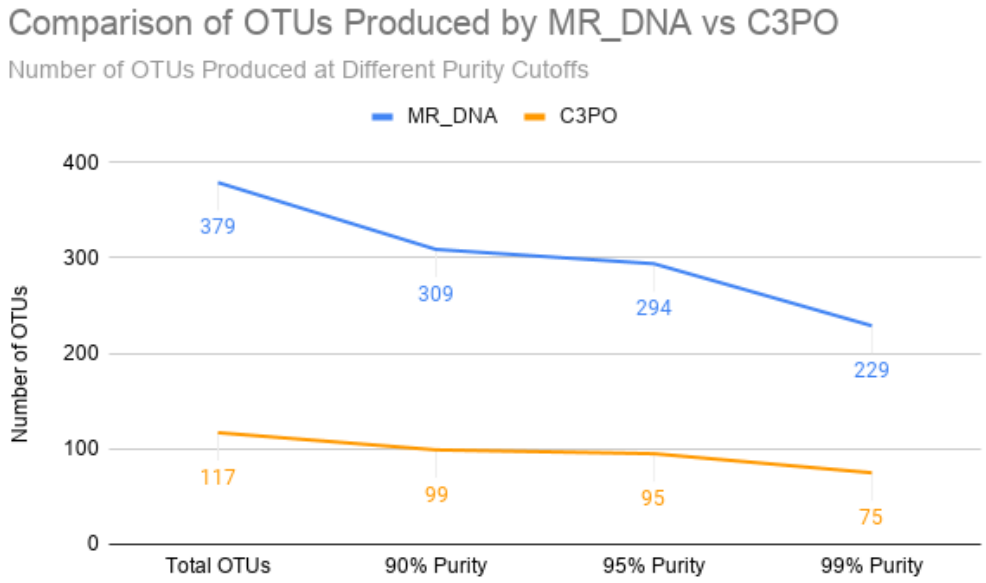


Figure 5.5: Line graph showing the *total number* of OTUs produced by MR_DNA versus C3PO at different purity cutoffs. Using the same five test samples, MR_DNA produces more overall OTUs than LOTUS at every purity cutoff.

5.4 Preliminary Analysis of OTU Sequence Length

The sequencing data from the samples sent to MR_DNA was used to develop C3PO. This meant that sequences had an average length of around 300 bp, and therefore

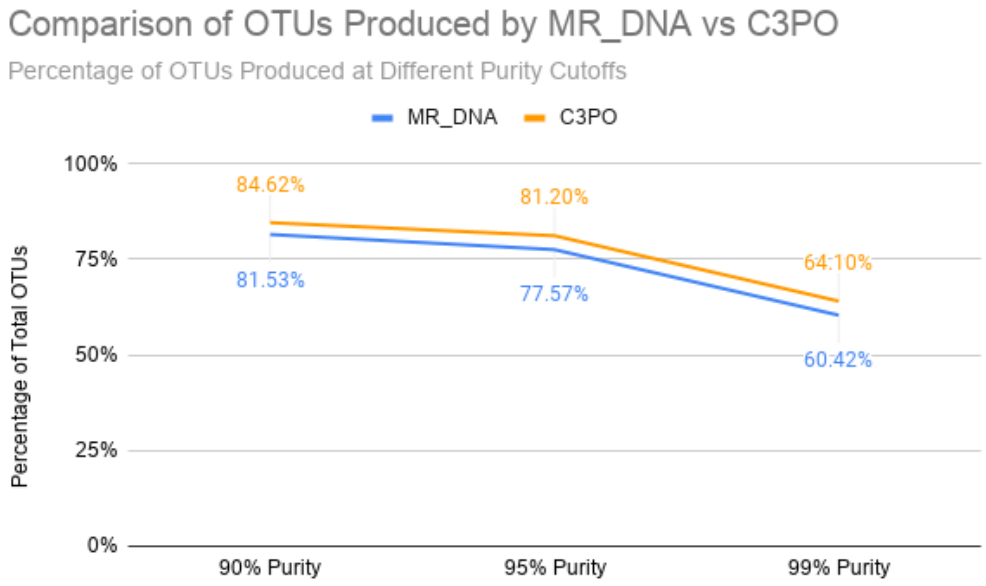


Figure 5.6: Line graph showing the *percentage* of OTUs produced by MR_DNA versus C3PO at three different purity cutoffs. Using the same five test samples, LOTUS produces a similar percentage of OTUs to MR_DNA at every purity cutoff.

the OTUs created by C3PO were also around 300 bp long. However, the MiniSeq produces 150bp sequences and biologists are interested in whether or not the shorter sequences produced by the MiniSeq can still produce single-source OTUs. Therefore, the next evaluation tested OTU creation using different sequence lengths.

Sequences of different lengths were produced using the optional Truncate Sequences step in C3PO as shown in Figure 3.5. Sequence truncation means the nucleotides beyond the given threshold were removed. For example, if the length is set at 200, only the first 200 nucleotides in the sequence are kept (including the barcode and primer) and the rest are discarded. Sequences were truncated at a given length and then run through the the rest of the pipeline to create OTUs. A special trim length of 123 is also included. This length is presumably what the MiniSeq would output for an 8 bp barcode and a 19 bp primer. Since $123 + 8 + 19 = 150$, the 123 length represents the MiniSeq read. The results of OTU processing showing the

differences in sequences and OTUs for different sequence trim lengths are shown in Table 5.3.

Table 5.3: LOTUS-produced OTU processing summary information for varying trim lengths from MR_DNA sequencing data.

Trim Length (bp)	Sequences	Uniques	Non-chimeras	Chimeras	Borderline	OTUs
100	516962	6740	6376	330	34	159
123[a]	516962	7771	7396	336	39	153
150	516962	9589	9040	448	101	121
200	516952	13537	10593	2890	54	133
250	516947	16128	12285	3787	56	140
300	516944	18023	13852	4141	30	120
316[b]	516943	19065	14775	4265	25	117

[a] MiniSeq length.

[b] Average untrimmed length from MR_DNA.

Table 5.4 shows that single-source OTUs are produced in consistent ratios even at different trim lengths. Figures 5.7 and 5.8 provide visualizations of the data in this table. In particular, 82.3% of the MiniSeq read length OTUs are 90% pure or higher. These results indicate that MiniSeq length reads can be used for the OBMM although further testing is warranted as discussed in Chapter 6.

5.5 Evaluate Open vs. *De novo* OTUs

As discussed earlier, the question of open vs. *de novo* OTUs is unsettled in academia and varies depending on the measure of success. For the purposes of the OBMM, success is defined as single-source OTUs. This work evaluates which method is more useful for source tracking in three ways: 1) by comparing total processing time to

Table 5.4: A table showing the number of single-source OTUs at various purity thresholds and at different truncations of base pairs. Three different purity thresholds were used for assessment. The number of OTUs generated differed with different trim lengths. Figures 5.7 and 5.8 represent the data visually.

Length (bp)	**Total OTUs**	**90% purity**		**95% purity**		**99% purity**	
		Number	%	Number	%	Number	%
100	159	129	81.1	127	79.9	104	65.4
123[a]	153	126	82.3	122	79.7	100	65.3
150	121	100	82.6	98	81.0	80	66.1
200	133	102	76.7	99	74.4	78	58.6
250	140	119	85.0	114	81.4	90	64.3
300	120	102	85.0	99	82.5	78	65.0
316[b]	117	99	84.6	95	81.2	75	64.1

[a] Length of MiniSeq sequences with barcode and primer removed.

[b] Average length of untrimmed sequences.

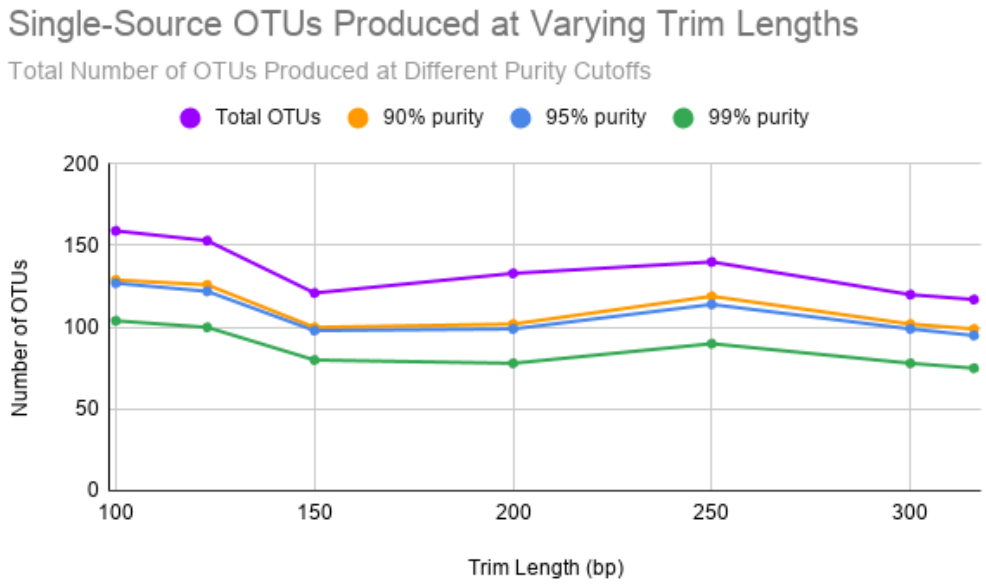


Figure 5.7: Line graph showing the *number* of OTUs produced at different sequence trim lengths. The 123 bp trim length represents the MiniSeq length and is comparable to the numbers produced at the actual length of 316 bp.

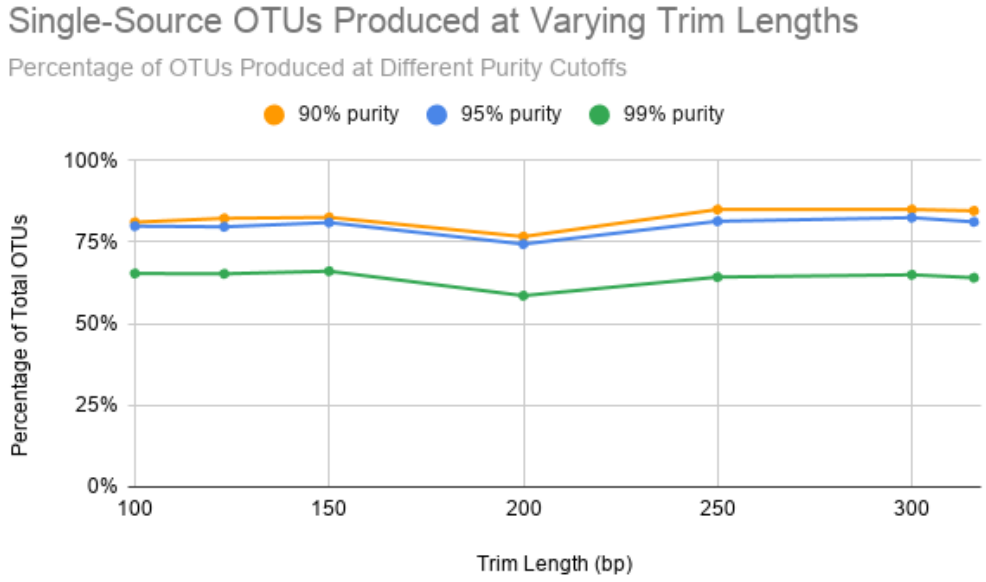


Figure 5.8: Line graph showing the *percentage* of OTUs produced at different sequence trim lengths. The 123 bp trim length represents the MiniSeq length and is comparable to the percentages produced at the actual length of 316 bp.

produce OTUs, 2) by measuring the entropy and purity of OTU clusters produced by both methods, and 3) by comparing ratios of single-source to multi-source OTUs.

A new set of test data was used to evaluate open vs. *de novo* OTUs produced by C3PO. Because the MiniSeq was in use for another research project, the samples were again sent to MR_DNA for sequencing rather than using the MiniSeq. However, the raw `fastq` provided by Illumina was used rather than using MR_DNA files, meaning all evaluation was done using LOTUS-generated files. The sample dataset shown in Table 5.5 was used to create both open and *de novo* OTUs for evaluation. The new dataset contains 12 samples from 12 different species with a total of 1,563,411 raw sequence reads. As discussed, the reads must go through processing in C3PO before being clustered into OTUs. To test open picking, the samples must be partitioned into **batches** to be added to the database at separate times. For this reason, the larger dataset of 12 samples rather than the original 5 sample dataset was used to create the library using the different strategies. For evaluation testing, each library created by a different strategy was loaded into its own database. Comparing OTUs between databases is not a feature of LOTUS at present.

The test setup needs further explanation, specifically in relation to batches. The term batch is used here specifically for evaluation of open picking. Three separate batches were created for testing: batch 3, batch 4, and batch 6. Batch 3 refers to partitioning the sample data into four sets of three samples each. The first set contains samples 1 - 3, the second samples 4 - 6, and so on. Batch 4 partitions the sample data into three sets of four samples in a similar manner. Batch 6 partitions the data into two sets of six samples each. The partitioning of samples used for each batch is shown in Table 5.6.

Batch notation is as follows for examples op3_6 and dn_6. The first two letters denote the OTU picking method used: "op" for open and "dn" for *de novo*. For

Table 5.5: Sample data used to construct OTU reference library. Each sample is a mixture of fecal material from 12 individuals to provide greater coverage of *Bacteroides* strains. DB Sequences (database sequences) are non-singleton, non-chimeric, quality filtered sequences suitable for use in the database. The number of sequences and uniques is from default open picking pipeline using batch 3 strategy op3_12.

Known Sample Label	Species	Number of DB Sequences	Number of Unique Sequences
1	Human	68692	6012
2	Horse	16721	1951
3	Cow	57604	4468
4	Pig	22488	2393
5	Pigeon	79641	5003
6	Dog	65639	4361
7	Deer	40690	4922
8	Turkey	39741	3168
9	Seagull	37665	5008
10	Sheep	40741	6075
11	Goat	69472	8021
12	Cat	79910	5266

Table 5.6: Sample partitioning for the three batches showing the sample number and species.

Batch Sample Partitioning							
Batch 3							
op3_3		**op3_6**		**op3_9**		**op3_12**	
1	Human	4	Pig	7	Deer	10	Sheep
2	Horse	5	Pigeon	8	Turkey	11	Goat
3	Cow	6	Dog	9	Seagull	12	Cat
Batch 4							
op4_4		**op4_8**		**op4_12**			
1	Human	5	Pigeon	9	Seagull		
2	Horse	6	Dog	10	Sheep		
3	Cow	7	Deer	11	Goat		
4	Pig	8	Turkey	12	Cat		
Batch 6							
op6_6		**op6_12**					
1	Human	7	Deer				
2	Horse	8	Turkey				
3	Cow	9	Seagull				
4	Pig	10	Sheep				
5	Pigeon	11	Goat				
6	Dog	12	Cat				

open methods, the first number is the batch number and the final number is the total samples in the batch. For *de novo* methods, the number is the total samples. A special case with "_rc" is for recluster. Comparisons between methods were made based on the number of samples. Open picking methods all initially used *de novo* clustering, and then used closed reference picking before clustering new OTUs. In batch 3 open picking, the procedure is as follows:

1. op3_3 is run first to initially cluster the first set of 3 samples into OTUs.

2. Next, op3_6 is run to open pick samples 4 - 6 against the OTUs created by samples 1 - 3 from the first set op3_3. Any sequences that do not match OTUs already present are clustered into new OTUs.

3. Then, op3_9 is run with samples 7 - 9 to open pick against the OTUs previously created by samples 1 - 6 from op3_3 and op3_6. Again, any sequences that do not match any OTUs are clustered into new OTUs.

4. Finally, op3_12 is run with samples 10 - 12 to open pick against the OTUs previously created by samples 1 - 9 from op3_3, op3_6, and op3_9. Any leftover sequences are clustered into new OTUs.

For each strategy, the library created by the strategy is saved to the database to allow purity and entropy calculations. The same procedure is used for batch 4 starting with op4_4 and batch 6 starting with op6_6. A special case is the three recluster strategies (e.g., op3_12_rc). In LOTUS, *de novo* OTUs are created through reclustering the database to save time by using already processed files. For evaluation, the dn batch *de novo* OTUs were instead created from the raw data directly to obtain accurate time comparisons. The recluster step uses the output files from samples that have already been through pre-processing. These pre-processed files are combined and used

for *de novo* OTU picking as illustrated in Figure 4.4. The recluster batches were only run on open strategies with 12 samples.

5.5.1 Timing Tests

As time efficiency is a secondary requirement of LOTUS, part of the evaluation of open vs *de novo* OTUs is to measure how long processing sequences takes from initial input to inclusion in the reference library database. One advantage of open picking methods is the ability to parallelize the closed reference OTU picking step. As *de novo* clustering is dependent on all the experimental sequences, it cannot be parallelized and therefore takes longer. Figure 5.9 shows the total processing time from raw `fastq` data to database OTUs for Batch 3 strategies.

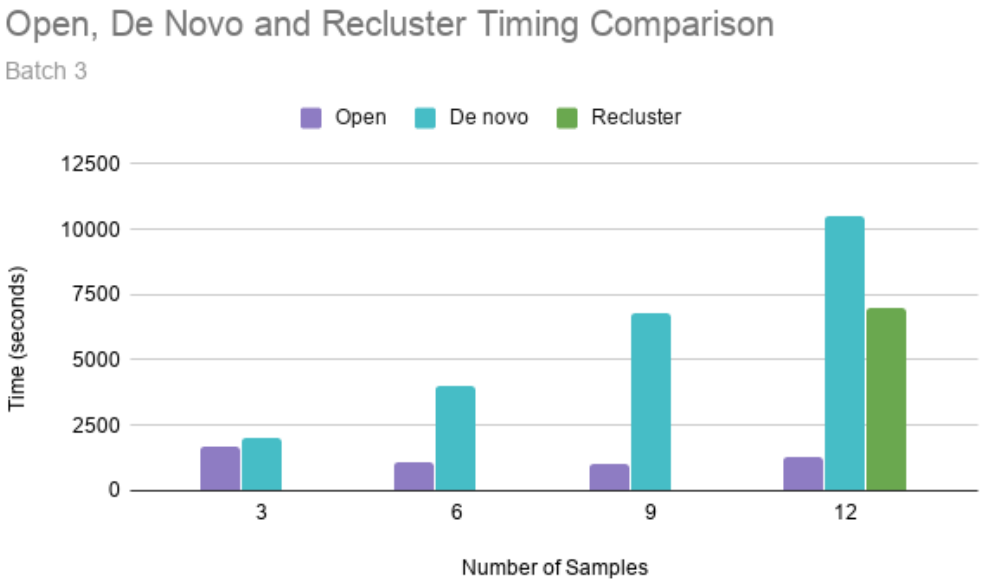


Figure 5.9: Bar graph showing total processing times for different batch 3 strategies based on number of samples. The recluster strategy op3_12_rc is only done for 12 samples. The comparison is made by number of samples. For example, dn_6 is the denovo strategy for 6 samples. It is compared to op3_6, the open strategy for 6 samples.

Similar timing comparison graphs for Batch 4 and Batch 6 are shown in Appendix

F. All batches show similar results. In comparing the methods based on the number of samples, open strategies are much faster in all cases as expected. This is a result of open strategies processing less data at one time. Looking at the Batch 3 comparisons, as each open strategy contains a set of 3 samples, the processing time is similar for each strategy. As each *de novo* strategy increases by 3 samples, the processing time has a corresponding increase. The time comparisons are further detailed in Table 5.7 which breaks the processing steps into the C3PO stages of pre-processing and OTU picking.

It is expected that the initial open strategy for all batches should be the same as the *de novo* because the initial open strategy uses *de novo* picking. For example, op3_3 and dn_3 are expected to be very similar and the table shows this to be the case. The advantage of the open strategies is seen in the ensuing steps. For example, op3_9 takes 17 minutes to process while dn_9 takes almost 2 hours. Technically, the processing time for op3_9 should include the processing time for op3_3 and op3_6, but from a user perspective uploading samples, the time a user perceives involves only the current samples. Therefore, timing comparisons were done from the point of view of the user.

The table also shows that OTU picking is the fastest process in the pipeline. This makes sense because OTU picking is done using pre-compiled C++ VSEARCH commands and includes parallelization. The pre-processing commands are Python QIIME commands which are not parallelized and Python scripts are also used for loading the database.

Compared to dn_12, the reclustering strategies do save time both in pre-processing and a small amount of time in the database loading step as the sample metadata does not need to be inserted again. The regular *de novo* processing dn_12 took 2 hours 54 minutes while the three reclustering strategies took 1 hour 56 minutes on average, a

Table 5.7: Timing comparison between batch strategies. All times are recorded in seconds. The three "rc" reclustering strategies do not have pre-processing times since the pre-processing occurred during the original run. For example, op3_12_rc was reclustered from op3_12 data which had already been through pre-processing. Since the pre-processing occurred during the op3_12 run, it does not apply to the time for reclustering from a user perspective.

Batch strategy	Pre-Processing	OTU Picking	Load Database	Total Time (seconds)	Total Time (HH:MM:SS)
op3_3	983.449	41.097	669.051	1693.597	00:28:14
op3_6	872.501	30.91	162.409	1065.820	00:17:46
op3_9	838.224	35.603	164.038	1037.865	00:17:18
op3_12	1069.407	42.635	181.71	1293.752	00:21:34
op4_4	1219.314	53.859	940.952	2214.125	00:36:54
op4_8	1172.592	46.973	261.937	1481.502	00:24:42
op4_12	1328.479	56.986	259.421	1644.886	00:27:25
op6_6	1850.595	85.061	1879.237	3814.893	01:03:35
op6_12	1884.802	87.486	430.303	2402.591	00:40:03
dn_3	1114.322	52.888	827.631	1994.841	00:33:15
dn_4	1205.405	52.289	1024.121	2281.815	00:38:02
dn_6	1832.625	81.743	2069.074	3983.442	01:06:23
dn_8	2339.795	117.157	3321.201	5778.153	01:36:18
dn_9	2647.244	137.544	4000.699	6785.423	01:53:05
dn_12	3685.895	214.214	6570.821	10470.93	02:54:31
op3_12_rc	N/A	216.866	6728.127	6972.525	01:56:13
op4_12_rc	N/A	229.708	6728.127	6957.835	01:55:58
op6_12_rc	N/A	226.358	6676.926	6903.284	01:55:03

savings of 1 hour for 12 samples.

From a user perspective, open picking is clearly faster as the data is broken up into smaller sections and parallelization also improves performance. C3PO-implemented open picking thus performs as described in the literature. Open picking is also a likely practical scenario as researchers will load data in smaller sections as they obtain it rather than uploading all the data at once. The reclustering option is shown to save time compared to regular *de novo* processing in case *de novo* OTUs are desired.

5.5.2 Purity & Entropy

Although timing is an important consideration in utilizing OTUs for source tracking, the most crucial factor is the sequence makeup of the OTU clusters. Entropy and purity are used as metrics to evaluate OTU clusters produced by both open and *de novo* methods. The goal is to minimize entropy and maximize purity in a given OTU cluster. For reference, the ideal OTU has an entropy of 0 and a purity of 1. As stated earlier, the worst possible entropy value for 12 species is 3.58496. The total weighted entropy and total weighted purity are shown in Table 5.8 for Batch 3, Table 5.9 for Batch 4, and Table 5.10 for Batch 6. A comparison graph of weighted entropy and weighted purity for only batch strategies of sample size 12 across all three tables is shown in Figure 5.10.

For all tables, a general observation can be made about entropy and sample size. The entropy increases as more samples are added. This makes sense as the number of OTU clusters also increases with more samples and entropy, as the summation of all clusters, increases with the number of clusters. A second general observation involves purity and sample size. As more samples are added, the purity decreases. This suggests that the species being added have similar or overlapping sequences to other species. Meaning, there are sequences that are not specific to a single species.

Table 5.8: Weighted Entropy and Weighted Purity of OTU clusters in Batch 3 Comparison of Open vs *De novo* OTU Picking Methods. Relevant summary information from different stages of C3PO sequence processing during OTU creation is also included.

	op3_3	dn_3	op3_6	dn_6	op3_9	dn_9	op3_12	dn_12	op3_12_rc
Total Number of OTUs	528	528	821	853	1247	1401	1431	1613	1618
Number of Samples Added Per Batch	3	3	3	6	3	9	3	12	12
Raw Input Sequences	413078	413078	360904	773982	347161	1121143	442268	1563411	1563411
Quality Filtered Sequences	386438	386438	337358	723796	324990	1048786	413089	1461875	1461875
Dereplicated Sequences	184990	184990	130597	309349	160523	462006	173938	607912	607912
Singletons Removed	161793	161793	112527	268213	140548	400995	149831	525027	525027
Unique Sequences	23197	23197	18070	41136	19975	61011	24107	82885	82885
Chimeras Removed	11410	11410	7544	18575	9878	27771	9978	35656	35648
Borderline Removed	169	169	83	272	165	466	210	724	714
Non-Chimeras (used for OTUs)	11618	11618	10443	22289	9932	32774	13919	46505	46523
Matches	N/A	N/A	9918	N/A	8135	N/A	13592	N/A	N/A
Non-Matches	N/A	N/A	1325	N/A	1797	N/A	327	N/A	N/A
Number Matches Added to DB	N/A	N/A	8099	N/A	7077	N/A	8313	N/A	N/A
Number Non-Matches Added to DB	N/A	N/A	1325	N/A	1797	N/A	327	N/A	N/A
New OTUs (from Non-Matches)	N/A	N/A	293	N/A	426	N/A	184	N/A	N/A
Number Unique Sequences in DB	11618	11618	21402	22289	29916	32774	38556	46505	46523
Total Sequences in DB	143017	143017	310785	317730	428881	438657	619004	645677	645735
Total Samples in DB	3	3	6	6	9	9	12	12	12
Weighted Entropy	0.092	0.092	0.484	0.525	0.812	0.886	1.364	1.467	1.467
Weighted Purity	0.986	0.986	0.896	0.886	0.804	0.793	0.634	0.608	0.608

Table 5.9: Weighted Entropy and Weighted Purity of OTU clusters in Batch 4 Comparison of Open vs *De novo* Methods. Relevant summary information from different stages of C3PO sequence processing during OTU creation is also included.

	op4_4	dn_4	op4_8	dn_8	op4_12	dn_12	op4_12_rc
Total Number of OTUs	625	625	1182	1264	1475	1613	1611
Number of Samples Added Per Batch	4	4	4	8	4	12	12
Raw Input Sequences	511276	511276	493394	1004670	558741	1563411	1563411
Quality Filtered Sequences	478447	478447	461490	939937	521938	1461875	1461875
Dereplicated Sequences	237475	237475	181017	409958	223289	607912	607912
Singletons Removed	208450	208450	155715	355764	192519	525027	525027
Unique Sequences	29025	29025	25302	54194	30770	82885	82885
Chimeras Removed	15239	15239	10218	24779	12887	35656	35645
Borderline Removed	199	199	181	387	290	724	724
Non-Chimeras (used for OTUs)	13587	13587	14903	29028	17593	46505	46516
Matches	N/A	N/A	12922	N/A	16602	N/A	N/A
Non-Matches	N/A	N/A	1981	N/A	991	N/A	N/A
Number Matches Added to DB	N/A	N/A	11611	N/A	11931	N/A	N/A
Number Non-Matches Added to DB	N/A	N/A	1981	N/A	991	N/A	N/A
New OTUs (from Non-Matches)	N/A	N/A	557	N/A	293	N/A	N/A
Number Unique Sequences in DB	13587	13587	27179	29028	40101	46505	46516
Total Sequences in DB	166037	166037	391239	397751	624725	645677	645737
Total Samples in DB	4	4	8	8	12	12	12
Weighted Entropy	0.188	0.188	0.601	0.642	1.387	1.467	1.467
Weighted Purity	0.965	0.965	0.870	0.865	0.630	0.608	0.608

Table 5.10: Weighted Entropy and Weighted Purity of OTU clusters in Batch 6 Comparison of Open vs *De novo* OTU Picking Methods. Relevant summary information from different stages of C3PO sequence processing during OTU creation is also included.

	op6_6	dn_6	op6_12	dn_12	op6_12_rc
Total Number of OTUs	853	853	1497	1613	1617
Number of Samples Added Per Batch	6	6	6	12	12
Raw Input Sequences	773982	773982	789429	1563411	1563411
Quality Filtered Sequences	723796	723796	738079	1461875	1461875
Dereplicated Sequences	309349	309349	318518	607912	607912
Singletons Removed	268213	268213	275307	525027	525027
Unique Sequences	41136	41136	43211	82885	82885
Chimeras Removed	18575	18575	18633	35656	35659
Borderline Removed	272	272	374	724	720
Non-Chimeras (used for OTUs)	22289	22289	24204	46505	46506
Matches	N/A	N/A	20033	N/A	N/A
Non-Matches	N/A	N/A	4171	N/A	N/A
Number Matches Added to DB	N/A	N/A	16353	N/A	N/A
Number Non-Matches Added to DB	N/A	N/A	4171	N/A	N/A
New OTUs (from Non-Matches)	N/A	N/A	644	N/A	N/A
Number Unique Sequences in DB	22289	22289	42813	46505	46506
Total Sequences in DB	317730	317730	638884	645677	645649
Total Samples in DB	6	6	12	12	12
Weighted Entropy	0.525	0.525	1.428	1.467	1.467
Weighted Purity	0.886	0.886	0.615	0.608	0.608

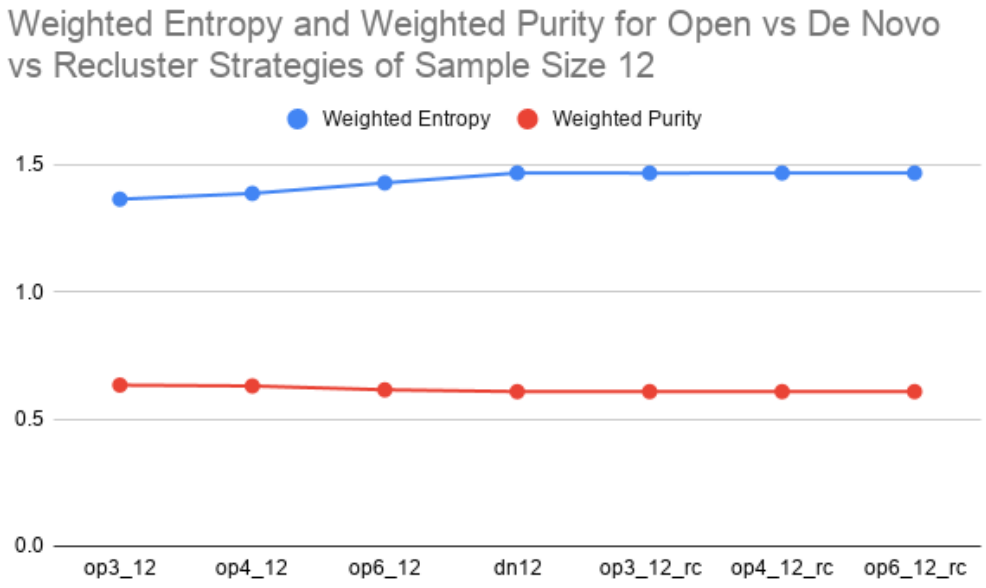


Figure 5.10: Line graph showing weighted entropy and weigthed purity of different batch strategies os sample size 12. The op3_12 strategy had the highest purity and lowest entropy, indicating that op3_12 has more overall pure OTUs than the other strategies.

It should be noted that the total weighted entropy and total weighted purity include all OTUs in the database for a given strategy. This gives a reference point for comparison of the OTU clusters for each batch strategy. However, purity and entropy would be different for source tracking as only single-source OTUs are used. As applied to source tracking, rather than using purity to evaluate single-source OTUs, purity is used to create single-source OTUs.

In comparing purity and entropy between open and *de novo*, the open method appears to perform slightly better although they are very similar. For example, in Table 5.8, comparing op3_9 to dn_9, the entropy is 0.812 for the open method and a slightly higher 0.886 for the *de novo* method. Similarly, the purity is 0.804 for the open method and a slightly lower 0.793 for the *de novo* method. Comparisons of sample size 12 batches as seen in Figure 5.10 are used in determining which strategy to use for the reference library in unknown matching. This pattern holds for open

vs *de novo* for the different strategies in all three batches with open methods having slightly lower entropy and slightly higher purity.

5.5.3 Single-Source vs. Multi-Source OTUs

As single-source OTUs are the goal for the OBMM, the question for open and *de novo* methods is which method produces higher quality OTUs? In other words, does the open picking method produce more single-source OTUs or does the *de novo* method?

The different strategies in the three batches were evaluated for the number of single-source OTUs to multi-source OTUs produced. Single-source OTUs are defined by the purity cutoff. Evaluations were made for each of the different batch strategies at five purity cutoffs: 50, 75, 90, 95, and 99. Table 5.11 contains the results. An example graph for this evaluation using total numbers of OTUs and single-source to multi-source ratios is seen in Figure 5.11. Similar graphs for the other batches and purity cutoffs are found in Appendix G. A different visualization comparing open vs *de novo* strategies of sample size 12 using percentages of single-source OTUs produced is shown in Figure 5.12. Corresponding graphs showing the open vs *de novo* strategy comparison of single-source OTUs produced for batches of size 3, 4, 6, 8, and 9 are located in Appendix H.

The results seen in Figure 5.11 are repeated in the other batches. From the graphs and Table 5.11, certain trends emerge. First, as purity increases, the number of single-source OTUs drops and the number of multi-source OTUs rises. Second, as the number of samples increases, the ratio of single-source to multi-source decreases, meaning there are fewer species-specific OTUs as the number of samples/species increases. This is due to the likelihood that with more species comes more sequence overlap. In other words, more species will potentially have more sequences that are similar to each other and therefore are clustered together into the same OTUs.

Table 5.11: Single-source to multi-source comparison for different batch strategies at five different purity thresholds. S = Single-source OTUs, M = Multi-source OTUs, R = Ratio of Single-source : Multi-source OTUs. Ratios greater than 1 mean than there are more single-source OTUs than multi-source OTUs.

Batch strategy	**50% purity**			**75% purity**			**90% purity**			**95% purity**			**99% purity**		
	S	**M**	**R**	**S**	**M**	**R**	**S**	**M**	**R**	**S**	**M**	**R**	**S**	**M**	**R**
op3_3	510	18	**28.3**	488	40	**12.2**	471	57	**8.3**	459	69	**6.7**	426	102	**4.2**
op3_6	786	35	**22.5**	730	91	**8.0**	682	139	**4.9**	667	154	**4.3**	615	206	**3.0**
op3_9	1173	74	**15.6**	1040	207	**5.0**	959	288	**3.3**	928	319	**2.9**	842	405	**2.1**
op3_12	1197	234	**5.1**	994	437	**2.3**	911	520	**1.8**	891	540	**1.7**	804	627	**1.3**
op4_4	594	31	**19.2**	576	49	**11.8**	549	76	**7.2**	532	93	**5.7**	481	144	**3.3**
op4_8	1144	38	**30.1**	1052	130	**8.1**	985	197	**5.0**	952	230	**4.1**	845	337	**2.5**
op4_12	1198	277	**4.3**	973	502	**1.9**	891	584	**1.5**	866	609	**1.4**	770	705	**1.1**
op6_6	786	67	**11.7**	708	145	**4.9**	630	223	**2.8**	597	256	**2.3**	503	350	**1.4**
op6_12	1141	356	**3.2**	953	544	**1.8**	853	644	**1.3**	814	683	**1.2**	696	801	**0.9**
dn_3	510	18	**28.3**	488	40	**12.2**	471	57	**8.3**	459	69	**6.7**	426	102	**4.2**
dn_4	594	31	**19.2**	576	49	**11.8**	549	76	**7.2**	532	93	**5.7**	481	144	**3.3**
dn_6	786	67	**11.7**	708	145	**4.9**	630	223	**2.8**	597	256	**2.3**	503	350	**1.4**
dn_8	1192	72	**16.6**	1055	209	**5.0**	947	317	**3.0**	896	368	**2.4**	743	521	**1.4**
dn_9	1266	135	**9.4**	1067	334	**3.2**	947	454	**2.1**	889	512	**1.7**	741	660	**1.1**
dn_12	1174	439	**2.7**	941	672	**1.4**	832	781	**1.1**	781	832	**0.9**	646	967	**0.7**
op3_12_rc	1189	429	**2.8**	948	670	**1.4**	847	771	**1.1**	799	819	**1.0**	664	954	**0.7**
op4_12_rc	1176	435	**2.7**	922	689	**1.3**	816	795	**1.0**	764	847	**0.9**	631	980	**0.6**
op6_12_rc	1175	442	**2.7**	933	684	**1.4**	831	786	**1.1**	782	835	**0.9**	647	970	**0.7**

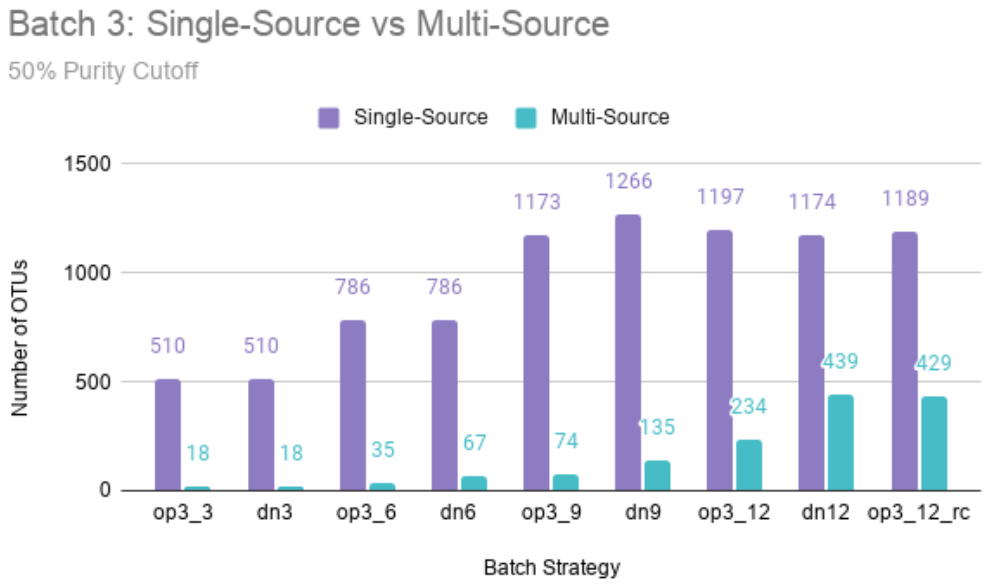


Figure 5.11: Graph comparing single-source vs multi-source OTUs for different batch 3 strategies using 50% purity cutoff to define single-source OTUs.

Lastly, in an unexpected result, open methods performed better overall, producing more single-source OTUs than the comparable *de novo* methods. Focusing on the strategies with 12 samples, op3_12 consistently maintained higher ratios than dn_12 and op3_12_rc at all purity levels. Figure 5.12 also reflects these findings with op3_12 outperforming all other strategies for sample size 12 in the production of the highest percentage of single-source OTUs. Additionally, op3_12 maintained ratios over one, meaning that there were more single-source OTUs than multi-source OTUs produced. The ratios for dn_12 and op3_12 fell below one at higher purities, meaning there were more multi-source OTUs. For source tracking, multi-source OTUs do not provide as much information since they can belong to multiple species.

The last result indicating that open methods outperform *de novo* is unexpected since it is assumed *de novo* methods would provide more accurate clusters. With centroid based clustering, one of the known issues is the choice of centroid sequence for a cluster. Using the most frequent sequence is seen as the best option for choosing a

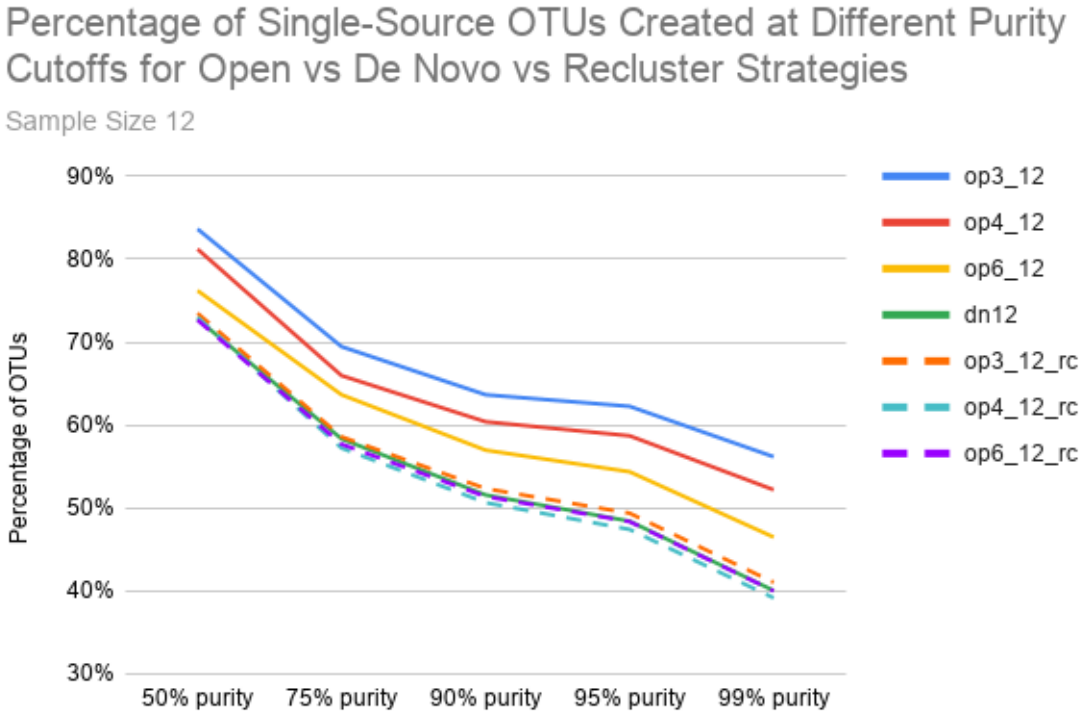


Figure 5.12: Graph comparing open vs *de novo* vs recluster strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 12. The recluster strategies (which end with "_rc") use *de novo* picking and are shown in dashed lines. The green line representing the *de novo* strategy dn_12 can be seen underlying the dashed lines for the recluster strategies. All three open strategies produced a higher percentage of single-source OTUs than any of the *de novo* strategies, with op3_12 producing the highest percentages.

"good" centroid. *De novo* clusters start with more sequences and may sort sequences by abundance differently than would be done in open clusters, which have a smaller pool of sequences at the start. However, the results seen in Table 5.11, Figure 5.12, and Appendix H indicate that open clusters have enough sequences to choose good centroids for OTU clusters. Another factor that may contribute to open methods performance is OTU stability. With each new batch of samples, sequences are added to established OTUs. Perhaps clustering from a smaller number of species to start creates purer species clusters. Further experimentation with different numbers of samples and different ordering of samples is recommended.

In addition to the results above, a unique batch was created as a default baseline of single-source to multi-source OTUs using the dn_12 sample data. The OTUs evaluated so far have all been clustered at the default 97% similarity, meaning sequences that were 97% similar are grouped into the same OTU. For a baseline, the question is do multi-source OTUs exist if OTUs are clustered at 100% similarity? In effect, this would mean that all the sequences in an OTU cluster are exactly the same. The results of this special clustering are seen in Table 5.12.

Table 5.12: Single-source vs. Multi-source OTUs created when OTUs are clustered at 100% identity. One-time batch using dn_12 sample data. All OTUs are created from identical sequences, therefore the presence of multi-source OTUs means that the exact same sequence is found in multiple species.

Purity Cutoff	**Number of OTUs**		
	Single-Source	**Multi-Source**	**Total**
Original dn_12			1613
50%	27551	11624	39175
75%	21411	17764	39175
90%	20121	19054	39175
95%	19735	19440	39175
99%	19291	19884	39175
100%	19272	19903	39175

The first observation is that the number of OTUs vastly increases from 1,613 to 39,175 which is logical considering that clustering is used to reduce the size of data. The second observation is that there are in fact multi-source OTUs at all purity cutoff levels. Since these OTUs contain exactly the same sequences, this means that more than one species has the exact same sequence which is the reason that no OTU method can produce only single-source OTUs and that impure clusters will always exist. However, the focus of this thesis is whether or not single-source OTUs can be found that can act as molecular signatures for source tracking, and the data reveals

that such single-source OTUs are possible.

5.6 Evaluate Matching Unknown Samples to OTUs

5.6.1 OTU Clustering at Default 97% Similarity

The final step in evaluation of LOTUS is testing the accuracy of identifying unknown samples. Given the results of the open vs *de novo* evaluation, the unknown matching was performed using the op3_12 open OTUs as the reference library. The original 5 samples sent to MR_DNA were used as the unknown samples as seen in Table 5.13. Single-source OTUs were matched at seven different purity cutoffs: None, 50, 75, 90, 95, 99, and 100. The results are displayed in Tables 5.14 - 5.20.

Table 5.13: Sample data used as unknowns for matching. Each sample is fecal material from a single individual animal. DB Sequences (database sequences) are non-singleton, non-chimeric, quality filtered sequences suitable for use in the database. These are the exact same samples as seen in Table 5.1.

Sample Label	Species	Number of DB Sequences	Number of Unique Sequences
Unk.Ca1	Cat	82834	4551
Unk.Do1	Dog	84336	1951
Unk.Ho1	Horse	59211	4509
Unk.Hu1	Human	80129	4205
Unk.Hu2	Human	70731	4055

Table 5.14 shows the results of matching the five "unknown" samples against the reference library built from 12 samples. The sequences in the unknown are matched to an OTU which has an assigned taxonomy. In this scenario with no purity cutoff, the taxonomy is assigned to the species with the most frequent sequences in the OTU cluster. This means that sequences are matched against all the OTUs in the library. Although the majority of sequences were matched, the successful matches

were poor, with only 2 samples out of 5 being correctly identified by a majority of matched sequences. By species, horse had the best accuracy, followed by human, then dog, then cat. The purity accuracy is almost the same as the overall accuracy with the difference being a few sequences that did not match to any OTUs in the reference database. These rare non-matching sequences suggest that there is still greater *Bacteroides* strain variation within a host species than is represented in the reference OTU library.

Table 5.15 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU clusters was 50% or higher. In this case, many sequences were unable to be matched to OTUs, and the successful matches improved slightly with 3 of the 5 samples being correctly identified by a majority of matched sequences. The non-matching sequences in this scenario essentially represent matches to multi-source OTUs. To judge the accuracy that reflects the single-source OTU matches, purity accuracy is used. Looking at purity accuracy by species, horse again performed best with 98.83% accuracy, followed by human, then dog, then cat. In general purity accuracy is higher for 50% pure OTUs than for no-purity OTUs. Using the highest number of matched sequences per sample as a classification, the horse sample and both human samples were correctly identified. The dog sample matched more with humans, but did match to a significant portion of single-source dog OTUs. The cat sample was completely misidentified as turkey with the next highest portion of sequences incorrectly matching to human.

Table 5.16 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU clusters was 75% or higher. In this case, the percentage of no match sequences increased from the 50% cutoff. Again, the non-matching sequences in this scenario essentially represent matches to multi-source OTUs. The successful matches also

improved with 4 of the 5 samples being correctly identified by a majority of matched sequences. Using purity accuracy, horse is still the most accurate species match holding at 98.85%, followed by human, dog, and cat. Again, purity accuracy is higher for 75% pure OTUs than for 50% pure OTUs. Increasing the purity cutoff from 50% to 75% did not change the cat accuracy as it was still misidentified as turkey.

Table 5.17 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU clusters was 90% or higher. Again, the percentage of no match sequences increased from the 75% cutoff, representing matches to multi-source OTUs. The successful matches also remained the same with 4 of the 5 samples being correctly identified by a majority of matched sequences. Looking at overall total accuracy, by species, horse is still the most accurate species match although total accuracy has fallen from 97.38% to 76.45% from the 75% purity cutoff to the 90% purity cutoff. This makes sense because accuracy falls as the number of no match sequences (multi-source OTU matches) increases. However, purity accuracy improves from 98.85% to 98.90% from the 75% purity cutoff to the 90% purity cutoff. This is the hypothesized outcome as horse sequences should match better to more pure horse OTUs. Interestingly, the human purity accuracy *decreased* from 88.60% to 75.86% from the 75% purity cutoff to the 90% purity cutoff. This means there are less human OTUs at the 90% purity cutoff for the human sequences to match. The cat sample is still identified as turkey even at 90% purity, meaning the OTUs are 90% pure turkey and still being matched to the unknown cat sequences. Oddly, the total accuracy for cat bounced slightly from 0.17% to 1.37%, likely because some of the sequences that previously matched to 75% pure human OTUs now matched to cat OTUs as the 75% pure OTUs were excluded.

Table 5.18 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU

clusters was 95% or higher. There is very little change from the 90% level. The accuracies per species are mostly unchanged. The cat sample is still misidentified as turkey.

Table 5.19 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU clusters was 99% or higher. The successful matches decreased with only 3 out of 5 samples being correctly identified by a majority of matched sequences. The cat sample is now misidentified as a human followed closely by turkey. The per species total accuracies are all below 50%. The drop in the accuracies is because there are far fewer OTUs to match against at the 99% purity cutoff. The purity accuracy increased for cat, dog, and horse from the 95% to 99% purity cutoff; however, the human purity accuracy decreased from 75.87% at the 95% purity cutoff to 65.59% at the 99% purity cutoff. This is due to a decrease in matched sequences from the Human1 sample. The Human2 sample sequences matches remained the same between purity cutoffs. Because the number of no match sequences increased for Human1 between 95% and 99% purity cutoff, it can be inferred that sequences that formerly matched to the 95% pure human OTUs no longer matched at the higher 99% purity level.

Finally, Table 5.20 shows the results of matching the five "unknown" samples against the subset of single-source OTUs in the reference library where the purity of the OTU clusters was 100%. There is little change from the 99% level. The successful matches remained the same with 3 out of 5 samples being correctly identified by a majority of matched sequences. The cat sample is back to being misidentified as a turkey. The per species total accuracies are all below 50%. Again, the drop in the overall accuracies is because there are far fewer OTUs to match against at the 100% purity cutoff. The purity accuracy for cat, dog, and horse increased from the 99% purity cutoff while the purity accuracy for humans slightly dropped from 65.59% to 64.17% due to a greater number of Human1 sequences being non-matching.

There are several possibilities for the continued misidentification of the cat sample as turkey or human across multiple purity thresholds. One reason could be because the cat sample was added as the last sample to the reference library in the open OTU picking strategy. By the time the cat sample was added to the library, many other species OTUs were already present, so many sequences could have been clustered into other species OTUs. However, this suggests the possibility that the issue may be more biological than computational. Even though there are cat-specific OTUs in the library, it is possible that this particular "unknown" cat did not have the same *Bacteroides* strain as the cat population used to create the library. A second possibility is that the *Bacteroides* strains in certain host species have more potential crossover sequences with the strains in other host species. Although *Bacteroides* is host-specific, it could be either that various *Bacteroides* strains between host species are similar enough that OTUs clusters cannot distinguish between them or that the short read length from NGS processing removes the variable region that distinguishes between species. Recall that the 16S gene is 1500 bp long, but the OTUs in the reference library are 300 bp long. The hypervariable region targeted by the primer in this project may correctly distinguish between certain host species, but not others (i.e., maybe hypervariable region V3 distinguishes between horses, but maybe another region such as V2 distinguishes between cats). Appendix I shows an analysis of single-source OTUs using the graphs of OTU Purity in the reference library by individual host species. An overview of the graphs shows that host species turkey, horse, pigeon and pig have a higher percentage of single-source OTUs, suggesting that the OBMM may perform better for some host species than others. Put another way, cat sequences are frequently found in OTUs that are not cat-specific, but turkey sequences are usually found only in turkey-specific OTUs, suggesting the OBMM may work well for turkeys but not for cat sources. However, further testing is needed before such a definitive conclusion can be made. A general recommendation is the addition of more

samples to the library, including several representatives of the same species to cover a wider array of sequence variations.

Interestingly, at all purity cutoffs, the Human2 sample matches to the human OTUs far better than the Human1 sample. This is likely due to an underlying biological explanation regarding the difference in gene expression in humans, or more technically the human *Bacteroides*. As with the cat misidentification issue, the recommendation is to add more data points to the OTU library to provide more sequence/strain coverage.

Table 5.14: Unknown Matching Accuracy for All Open OTUs (no purity cutoff). Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Both single- and multi-source OTUs are used since no purity cutoff for single-source was defined. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 12 sequences that did not match to any OTUs in the library and there were 55,058 sequences that matched to OTUs classified as Cat species based on purity cutoff. The purity cutoff in this case was not defined, meaning classification is defaulted to the species with the most frequent sequences in the OTU. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	12	12	13	60	703
Cat	55058	238	7961	44419	273
Cow	19			53	
Deer		17			
Dog	154	145	3499	10374	58
Horse	96	133	147	151	57658
Human	24646	69984	30035	29108	435
Pigeon					
Seagull	27			46	
Sheep	11	1			
Turkey	106	201	41179	125	84

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	62.73	9.61	12.3	97.38
Accuracy_p (%)	62.74	9.61	12.31	98.55

Table 5.15: Unknown Matching Accuracy for Open OTUs with 50% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 50% pure or higher single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 43,511 sequences that did not match to any OTUs in the library and there were 385 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	43511	141	10989	14718	870
Cat	385	63	127	11294	51
Cow					
Deer		17			
Dog	11156	144	278	28653	112
Horse	96	133	147	151	57658
Human	24861	70031	30110	29363	436
Pigeon				3	
Seagull				28	
Sheep	14	1		1	
Turkey	106	201	41183	125	84

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	62.9	0.15	33.97	97.38
Accuracy_p (%)	88.51	0.18	41.16	98.83

Table 5.16: Unknown Matching Accuracy for Open OTUs with 75% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 75% pure or higher single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 43,686 sequences that did not match to any OTUs in the library and there were 258 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	43686	155	11016	20138	884
Cat	258	61	111	11260	43
Cow					
Deer		17			
Dog	11173	148	281	28663	115
Horse	96	133	147	151	57658
Human	24810	70016	30096	23968	427
Pigeon				3	
Seagull				28	
Sheep					
Turkey	106	201	41183	125	84

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	62.86	0.13	33.99	97.38
Accuracy_p (%)	88.60	0.15	44.65	98.85

Table 5.17: Unknown Matching Accuracy for Open OTUs with 90% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 90% pure or higher single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 45,885 sequences that did not match to any OTUs in the library and there were 3,641 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	45885	14173	33082	38673	13443
Cat	3641	6461	1135	9311	90
Cow					
Deer		17			
Dog	11173	148	281	28663	115
Horse	72	101	104	118	45265
Human	19252	49630	7049	7415	214
Pigeon				3	
Seagull				28	
Sheep					
Turkey	106	201	41183	125	84

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	45.66	1.37	33.99	76.45
Accuracy_p (%)	75.86	2.28	62.77	98.90

Table 5.18: Unknown Matching Accuracy for Open OTUs with 95% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 95% pure or higher single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 45,902 sequences that did not match to any OTUs in the library and there were 3,625 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	Human1	Human2	Cat1	Dog1	Horse1
Total Sequences	80129	70731	82834	84336	59211
No Match	45902	14175	33094	38777	13443
Cat	3625	6461	1132	9259	90
Cow					
Deer		17			
Dog	11173	148	281	28663	115
Horse	72	101	104	118	45265
Human	19251	49628	7040	7363	214
Pigeon				3	
Seagull				28	
Sheep					
Turkey	106	201	41183	125	84

	Human	Cat	Dog	Horse
Accuracy_t (%)	45.66	1.37	33.99	76.45
Accuracy_p (%)	75.87	2.28	62.91	98.90

Table 5.19: Unknown Matching Accuracy for Open OTUs with 99% Purity Cutoff. Accuracy$_t$ is overall total accuracy and Accuracy$_p$ is purity accuracy. Only 99% pure or higher single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 57,714 sequences that did not match to any OTUs in the library and there were 3,625 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	57714	6461	67881	39016	35584
Cat	3625		1132	9259	90
Cow					
Deer	3	17			
Dog	11173	148	281	28663	115
Horse	37	49	55	71	23213
Human	7557	49304	6921	7287	186
Pigeon			2	3	
Seagull				28	
Sheep					
Turkey	20	14	6562	9	23

	Human	**Cat**	**Dog**	**Horse**
Accuracy$_t$ (%)	37.69	1.37	33.99	39.2
Accuracy$_p$ (%)	65.59	7.57	63.25	98.25

Table 5.20: Unknown Matching Accuracy for Open OTUs with 100% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 100% pure single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 59,042 sequences that did not match to any OTUs in the library and there were 3,629 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	59042	31536	70735	45889	37433
Cat	3629	6518	1133	9260	90
Cow					
Deer	3	17			
Dog	11173	148	281	28663	115
Horse	34	42	48	70	21458
Human	6228	32456	4073	414	92
Pigeon			2	3	
Seagull				28	
Sheep					
Turkey	20	14	6562	9	23

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	25.64	1.37	33.99	36.24
Accuracy_p (%)	64.17	9.36	74.55	98.53

5.6.2 *De novo* OTU Clustering at Restrictive 100% Similarity

At the biologists' request, a special case was evaluated using more restrictive constraints in an effort to judge the usefulness of OTUs for MST. The reference OTU library was constructed using the *de novo* dn_12 strategy, but with OTUs clustered at 100% similarity rather than the default 97% similarity. This means that all the sequences in a given OTU were basically identical while allowing for alignment gaps. Unknown matching was further restricted to 100% purity and unknown sequences were also matched at 100% similarity. The results of this special restricted matching are seen in Table 5.21.

In this case, there are many non-matching sequences in each test sample. The successful matches are disappointing with only 3 out of 5 samples being correctly identified by a majority of matched sequences. Although the purity accuracy of human and horse is high, both cat and dog are low due to misidentification. The cat sample is still misidentified as turkey and the dog sample is misidentified as human. The *de novo* clustering method helps rule out the possible computational cause of misidentification due to the order of input in open picking. The 100% similarity OTU picking also helps narrow down computational causes of misidentification. This leaves either a biological explanation, lab error or contamination, or some other OTU filtering threshold yet to be determined.

As this restrictive criteria still results in misidentification of the cat and the dog, it is possible there is another filtering threshold that needs to be determined. One idea is using the abundance (size) of OTU clusters to determine a "good" OTU cluster. One final test of unknown matching was run using the same criteria as above with the additional restriction of only using OTU clusters made of 100 sequences or more. The results are seen in Table 5.22. The only OTUs matched for all sequences were Human OTUs. Further investigation found this is because there are only 6 OTUs (3

Human, 2 Pigeon, and 1 Cow) in the database that meet this restrictive criteria and have the potential to be used for unknown matching. Hence, there should ideally not be any matches to cat, dog, or horse species at this level. Abundance criteria could potentially help narrow down source-specific OTUs and is discussed further in future work Section 6.2.1 under the heading Quality Control.

Table 5.21: Special Restrictive Case: Unknown Matching Accuracy for *De novo* OTUs clustered at 100% Similarity with 100% Purity Cutoff. Accuracy_t is overall total accuracy and Accuracy_p is purity accuracy. Only 100% pure single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 73,353 sequences that did not match to any OTUs in the library and there were 495 sequences that matched to OTUs classified as Cat species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	73353	61721	75865	81619	45799
Cat	495	91	206	99	1
Cow					
Deer					
Dog	1089	16	250	228	
Horse	9	19	29	32	13357
Human	5143	8876	2436	2273	43
Pig	19		1	35	1
Pigeon					
Seagull	15			47	
Sheep					
Turkey	6	8	4047	3	10

	Human	**Cat**	**Dog**	**Horse**
Accuracy_t (%)	9.29	0.25	0.27	22.56
Accuracy_p (%)	88.81	2.96	8.39	99.59

Table 5.22: Special Restrictive Case with Abundance: Unknown Matching Accuracy for *De novo* OTUs clustered at 100% Similarity with 100% Purity Cutoff and OTU cluster size >= 100. Accuracy$_t$ is overall total accuracy and Accuracy$_p$ is purity accuracy. Only 100% pure single-source OTUs are used for matching. Each column gives a breakdown of the number of sequences from the unknown sample that were matched to an OTU which was taxonomically assigned to the species. For example, for sample Human1, there were 80,091 sequences that did not match to any OTUs in the library and there were 38 sequences that matched to OTUs classified as Human species based on purity cutoff. The highest number of sequences per sample is highlighted in green if correct species match or red if incorrect species match.

Samples used as Unknowns	**Human1**	**Human2**	**Cat1**	**Dog1**	**Horse1**
Total Sequences	80129	70731	82834	84336	59211
No Match	80091	70424	82832	84333	59211
Cat					
Cow					
Deer					
Dog					
Horse					
Human	38	307	2	3	
Pig					
Pigeon					
Seagull					
Sheep					
Turkey					

	Human	**Cat**	**Dog**	**Horse**
Accuracy$_t$ (%)	0.23	0.00	0.00	
Accuracy$_p$ (%)	100	0.00	0.00	

Chapter 6

CONCLUSION

Dr. Michael Black and Dr. Chris Kitts of the Cal Poly Center for Applications in Biotechnology (CAB), in conjunction with the Computer Science Department, are investigating a new library-dependent MST method using OTUs clustered from 16S rRNA *Bacteroides* sequences to identify fecal contamination in aquatic environmental samples. The work in this thesis describes LOTUS, the Cal Poly **L**ibrary of **OTUs**. The components of LOTUS form a software solution built to assist with the computational portion of this new OTU-Based MST Method (OBMM). The requirements for LOTUS are outlined in Chapter 3 – Sections 3.1 and 3.5.3. Chapter 4 summarizes how those requirements were met in the software product as seen in Table 4.1. The main contribution of LOTUS is C3PO, the **C**al **P**oly **P**ipeline for **P**icking **O**TUs, a pipeline that creates OTUs from raw sequence reads. Other contributions include:

- A reference library utilizing a database design that models OTUs and the underlying data relationships that are necessary for source tracking
- A method of assigning taxonomy to OTUs and therefore enabling identification of unknown sources
- A simplified web-based user interface that can be used by researchers with varying levels of technical expertise

6.1 Conclusion

There were many questions at different stages of the development of LOTUS to answer regarding whether or not OTUs could be used as an MST methodology. Preliminary

feasibility tests confirmed that OTUs clustered from 97% similar sequences are related phylogenetically, meaning that OTUs can potentially be used as molecular signatures for source tracking. Further testing confirmed that LOTUS could be used to create OTUs with a standardized process for creating consistent OTUs without the need for sending samples to an outside genetic laboratory. The last test in regards to the use of OTUs themselves looked at sequence length and showed that related sequences can still be clustered down to the MiniSeq platform's read length output of 150 base pairs.

Once preliminary testing confirmed the feasibility of using OTUs as molecular signatures for MST, evaluations focused on two ultimate questions:

1. Can LOTUS create high quality single-source OTUs that can serve as a reference library? If so, which OTU picking strategy should be used to create such OTUs?

2. Can these reference OTUs be used for microbial source tracking of unknown environmental samples?

6.1.1 Reference Library OTUs

A primary function of LOTUS is creating high quality OTUs from raw sequence reads. In order to be useful as a reference library for MST, the OTUs must be created in a standardized manner that results in consistent OTUs across samples while efficiently using time and computer resources. High quality OTUs in this thesis are defined as single-source OTUs in which the percentage of sequences above a given purity threshold come from a single host species. Such OTUs can be used to represent that species during source tracking.

C3PO was used to create reference OTUs using different batch strategies to determine which OTU picking method created more single-source OTUs. Three different

aspects of batch strategy performance were measured: timing comparison, purity and entropy, and ratios of single-source to multi-source OTUs. Across all three metrics using 12 samples, the open picking method outperformed the *de novo* picking.

The timing tests confirmed that open picking is faster than *de novo* picking. These results reflect the advantages of open picking: parallelization during the closed reference picking step and a smaller number of sequences to process per project run.

Final LOTUS OTU clusters were evaluated using purity and entropy as discussed in Chapter 5. Ideal clusters would have a purity of 1.0 (or 100%) and entropy of 0, indicating no impurity or disorder in the cluster. The worst possible entropy for the 12 species in the reference library ($k = 12$) is $log_2 k = 3.58496$. Practically speaking, there are no OTU picking strategies that produce all ideal clusters. Even with OTU clusters created with 100% similar sequences (i.e., identical sequences), there remain impure OTUs, meaning the sequences in those clusters were found in multiple species and are not phylogenetically related. Therefore the goal was to find the best performing strategy that provided the highest number of single-source OTUs that could be used.

For each batch strategy, the total weighted purity and total weighted entropy was used to evaluate all the OTU clusters produced by that strategy. Among the seven strategies[1] using all 12 samples to create the reference OTUs, op3_12 performed the best, with a total weighted purity of 0.634 and a total weighted entropy of 1.364 for 1,431 OTUs. The total weighted purity value means that the average OTU cluster in op3_12 had a purity of 63.4%, or that over half the sequences in each OTU belonged to a single host species. This result suggests that op3_12 has more higher percentage purity OTUs overall than the other strategies measured.

The last metric looked at single-source vs multi-source OTUs produced by each

[1] op3_12, op3_12_rc, op4_12, op4_12_rc, op6_12, op6_12_rc, and dn12

strategy at various purity thresholds. The total weighted purity looks at the **average** purity of all the OTUs produced, where this evaluation focuses specifically on single-source OTUs produced. The theory being that single-source OTUs would be source-specific OTUs and hence act as molecular signatures. The highest ratio was reached by strategy op3_12 at 75% purity which produced 994 single-source OTUs to 437 multi-source OTUs.

Based on these results, the reference library created by batch strategy op3_12 was used for evaluation testing. These findings also recommend open OTU picking over *de novo* picking, and the default settings in LOTUS are set to open picking. It should be noted that this recommendation is based on a limited set of sample data and may change with the addition of new samples.

6.1.2 Unknown Matching For MST

The reference library OTUs from batch strategy op3_12 were used for evaluation of LOTUS unknown matching. The five samples from the original data sent to MR_DNA were used as "unknown" samples. The sequences from the unknown samples were clustered to an OTU in the reference library which was assigned to a host species taxonomy based on purity.

At 95% purity for the reference OTUs, LOTUS correctly matched four of the five unknown samples according to host species (horse at 98.90% purity accuracy, both human samples at 75.87% purity accuracy, and dog at 62.91% purity accuracy). The cat unknown sample was completely misidentified as turkey with 2.28% purity accuracy. The per species overall total accuracy dropped as purity increased. At 99% purity, only 3 of 5 unknown samples successfully matched. The unknown samples that came from the cat were misidentified at every purity level tested. The misidentification issue may have several possible causes including: sample order dur-

ing open picking, differences in individual host *Bacteroides* strains, close similarity of *Bacteroides* strains between host species, or even possible contamination of one host strain by another host (e.g., if the cat ate a bird). Many of these causes are biological, not computational, and can be explored further by biologists using LOTUS. The single-source analysis in Appendix I shows that the OBMM may work better for certain host species.

Overall, these results indicate that OTUs can be used successfully for MST, but this method needs refining.

6.2 Future Work

LOTUS is designed to be part of the OBMM, but it does not yet function in that capacity. The current functionality of LOTUS is designed to investigate whether or not OTUs can be used as molecular signatures for source tracking. Much of the work done in this thesis laid the groundwork for the back-end processing that would be needed to ensure data integrity. There is further work to be done to create a fully functioning MST method.

6.2.1 Methodology

MiniSeq

The evaluations in this thesis used a total of 17 samples from 12 different host species with sequences produced by the genetic laboratory MR_DNA; however, the goal of the OBMM is to use Cal Poly's in-house Illumina MiniSeq platform. The MiSeq used by MR_DNA and the MiniSeq are similar platforms, but differ in read lengths produced. Although evaluation testing in Section 5.4 on shortened read lengths indicated that single-source OTUs are accurately produced, unknown matching using shorter 150 bp

sequences still needs to be evaluated. It is highly recommended that LOTUS be tested using MiniSeq reads, first using entropy and purity to evaluate the reference library OTUs, and second evaluating the accuracy of these "shorter" OTUs in unknown source matching.

Resolving Misidentification

The misidentification of unknown samples that occurred during evaluation can have either a biological or a computational cause. The challenge going forward is to distinguish between these. One possible computational cause of the misidentification is the order of input sequences during open picking. This can be tested in two ways: 1) by reordering the sample input used to build the reference library, or 2) by matching unknown samples to *de novo* OTUs since sequence abundance rather than order is used. It is recommended to test unknown matching with these two alternatives to rule out a computational cause of misidentification. The special case of restricted *de novo* matching still produced the cat misidentification, suggesting that reordering would not change this outcome. However, it is still recommended to test different orders of input because if input order drastically changes the OTUs produced, then open picking would not be the appropriate picking approach to build the library for the MST. Unknown matching to *de novo* OTUs was only implemented for the special evaluation case and needs to be fully incorporated into LOTUS.

Library Size

As with any library-dependent MST method, the size of the library determines the usefulness for MST. A minimum number of samples per host species needs to be established. In other words, how many different cats need to provide samples to give an accurate representation of all OTUs that could be classified as "cat"? The reference

library was built with 12 samples from 12 different host species. It is unknown how many samples are the minimum needed for a reference library of OTUs, but more samples are needed from the current host species to determine the effect on purity and entropy of library OTUs. It is recommended that the samples come from multiple different host individuals (e.g., many different cats).

Testing on a larger library could potentially help resolve misidentification due to computational conditions. If sample input order is found to affect unknown matching, then having more samples in the reference library would presumably encompass more genetic variation between individuals and help rule out that possibility.

More Sample Data

In general, more sample data from both known and unknown samples needs to be collected and tested to more conclusively evaluate the reference OTUs and the unknown matching capability. The dataset used for evaluation has 12 different individuals representing a host species. It is possible this does not cover all the variation possible in a host species. The cat unknown sample hints at this possibility. Twelve different cats were used to construct the reference library OTUs, but the one unknown cat instead clustered with turkey and human OTUs. Why did the unknown cat's sequences not match with the other cats? More sample data from more cats could help answer this question.

There also seemed to be a lot of crossover in general between humans, cats, and dogs with sequences from all three species matching to OTUs from all three species. Is this due to lab error or is this reflective of some underlying biological principle? Are these cats and dogs pets? As pets, do they have more overlapping strains with humans and would they have different microbiota than feral cats and dogs? More sample data from all three species with subcategories of pet cats versus feral cats

would help answer these questions.

More human sample data could also help with the human misidentification that occurred at 99% purity. The evaluation used samples from 14 different humans (the 12 known Humans and the unknowns Human1 and Human2). In this case, the unknown Human1 did not match as accurately as the unknown Human2 did to the known Human reference sample. At all purity cutoffs, Human2 had a higher percentage of matching sequences than did Human1. This highly suggests individual variation and the need for more representative samples in the library.

Quality Control

Quality control measures such as the minimum abundance criteria for filtering OTUs also need to be established. In other words, how reliable are OTUs that consist of only two sequences? Three sequences? Are OTUs with fewer sequences more source-specific than OTUs with more sequences? Perhaps a weighting scheme can be implemented to use OTUs of a certain size for unknown matching. Figure 6.1 shows the abundance information by species for 100% pure OTUs created in the special restricted case of the dn_12 strategy clustered at 100% similarity. Figure 6.2 shows the abundance information by species for 100% pure OTUs created by the default op3_12 strategy clustered at 97% similarity. LOTUS currently has a minimum of two sequences per OTU cluster, but further research is needed to determine the appropriate minimum or maximum cutoffs.

Source-specific OTUs

The goal of this thesis was to answer whether OTUs could act as molecular signatures for source tracking. The work in the thesis confirms that OTUs can potentially be used for MST. Using limited sample data, testing shows that single-source OTUs

numOTUs	lowAbundance	highAbundance	commonName	percent
1033	2	10	Cat	100.00
2249	2	214	Cow	100.00
1235	2	27	Deer	100.00
561	2	9	Dog	100.00
1584	2	34	Goat	100.00
1166	2	79	Horse	100.00
3392	2	312	Human	100.00
1322	2	85	Pig	100.00
3265	2	147	Pigeon	100.00
808	2	16	Seagull	100.00
557	2	21	Sheep	100.00
2100	2	60	Turkey	100.00

Figure 6.1: SQL Results showing the abundance information of 100% pure OTUs by species in the dn_12 strategy clustered at 100% similarity. The minimum size of an OTU cluster is denoted by the lowAbundance column and is two sequences. The maximum size of an OTU cluster is denoted by the highAbundance column and ranges from 9 to 312 sequences.

numOTUs	lowAbundance	highAbundance	commonName	percent
43	2	10	Cat	100.00
38	2	258	Cow	100.00
43	2	21	Deer	100.00
15	2	5	Dog	100.00
40	2	8	Goat	100.00
126	2	171	Horse	100.00
72	2	96	Human	100.00
70	2	108	Pig	100.00
119	2	72	Pigeon	100.00
30	2	32	Seagull	100.00
21	2	10	Sheep	100.00
166	2	183	Turkey	100.00

Figure 6.2: SQL Results showing the abundance information of 100% pure OTUs by species in the op3_12 strategy clustered at 97% similarity. The minimum size of an OTU cluster is denoted by the lowAbundance column and is two sequences. The maximum size of an OTU cluster is denoted by the highAbundance column and ranges from 5 to 258 sequences.

are promising, but may not be enough for source tracking. The unknown matching methodology used in this thesis was designed to test unknowns that came from a single individual host animal as that was the data available for testing. However, the MST method envisioned by the biologists will need a methodology that adequately handles "mixed" samples with multiple potential source host species as the current methodology may lead to many false positives. Given that the results for unknown matching even at the most restricted OTU criteria were not ideal (i.e., pure Human OTUs were matching to sequences from many species in the database), further OTU filtering criteria needs to be explored. The challenge for the future is to determine which of these single-source OTUs is only found in a specific species and not in any other.

Unknown Matching To Outside Reference Database

Finally, it is recommended to get a baseline comparison for unknown matching against an outside reference OTU database such as Greengenes [30], RDP [27] or SILVA [113]. These taxonomically annotated databases have two points of interest for comparison with LOTUS: 1) the databases have a larger sample pool as they are built from more representative samples and contain more sequences, and 2) database OTUs are constructed from the full 1500bp 16S rRNA gene.

Matching unknown sequences produced by LOTUS to an outside reference database can provide additional insight into the use of these target regions for MST. The larger sample pool found in the outside databases should mean that reference OTUs provide coverage of more *Bacteroides* strains than LOTUS can with limited sample data. Further, LOTUS presumably produces more OTU clusters from the same sequences due to the shorter read length, meaning the sequences that cluster into an OTU in LOTUS may not cluster together when clustered into OTUs constructed from the

entire 1500bp gene length. In other words, the outside database OTUs are clustered with less granular variation due to the longer read length. The following rough approximation shows the difference in picking OTUs for two different read lengths:

- 97% of 1500bp = 1,455 nucleotide matches to be included in OTU cluster
- 97% of 300bp = 291 nucleotide matches to be included in OTU cluster

LOTUS needs fewer nucleotide matches at the 300bp length, resulting in potentially different final OTU clusters. Evaluating the same unknown samples used in LOTUS testing against the outside reference library OTUs can help provide investigators with further information using a known set of standards to determine the next steps going forward in this MST investigation.

6.2.2 Performance

C3PO was constructed from existing software tools to create consistent and accurate OTUs that can be used as the reference library in LOTUS. With the order of the pipeline steps now decided as seen in Figure 3.5, each step can be improved upon by future developers.

One suggestion to greatly speed up performance is to write a new demultiplexer. The QIIME script `split_libraries.py` combines quality filtering, demultiplexing, barcode/primer removal, and trimming into a single command. Because there is no dependency between the sequences, this step would benefit greatly from parallelization. To replace `split_libraries.py`, the demulitplexer must also perform quality filtering and sequence trimming. There is no need to remove homopolymers for Illumina data. Another time saving solution is to write the demultiplexer to handle `fastq` files directly rather than needing to convert to `fasta`/`qual` files first. Even

more time efficient would be to write the demultiplexer in a pre-compiled language such as C++ rather than an interpreted language like Python.

Another option for improving performance involves investigating other OTU picking methods. Greedy-centroid based OTU clustering was chosen for LOTUS as the best option for time and computer resources, but other OTU picking options as shown in Table 2.1 could be investigated to test their efficiency for comparison.

6.2.3 Web Application

The most important functionality to be added to LOTUS is full deployment as a working website for CAB biologists to use in their investigation. The application is written as a Django web application and needs to be deployed to a server such as one belonging to Cal Poly's Computer Science Department or a cloud service such as Amazon Web Services (AWS). Cal Poly's servers offer a free option for continuing development of this application with the OBMM, but may have hardware constraints that could affect computational ability due to the amount of NGS data generated by increasing numbers of samples and sequences. Cloud services offer a scalable solution, but do incur a financial cost.

6.2.4 Additional Analytical Capabilities

LOTUS currently provides only basic matching functionality for unknown samples, but the data it contains can be used for multiple other analysis. Once the reference library OTUs are finalized, questions such as "Do OTUs change over time?" or "Are there differences in OTUs by geographic location?" can be answered through database queries, but need further back-end and front-end development for use as an MST application. For example, researchers might want to study the differences between OTUs in given locations or see if there are region-specific OTUs. While the database

incorporates latitude and longitude data, current users do not have access to it, and future studies in this area would benefit from adding a graphical map to the website as well as the ability to search and select by location. Other features such an enhanced graphs and tables would be beneficial once tailored to the researchers' study requirements.

BIBLIOGRAPHY

[1] PEAR: Paired-End reAd mergeR. https://cme.h-its.org/exelixis/web/software/pear/doc.html. Date last accessed: 6/18/2020.

[2] SeqPrep. https://github.com/jstjohn/SeqPrep. Date last accessed: 6/18/2020.

[3] E. Adam, S. Collier, K. Fullerton, J. Gargano, and M. Beach. Prevalence and Direct Costs of Emergency Department Visits and Hospitalizations for Selected Diseases that can be Transmitted by Water, United States. *Journal of Water and Health*, 15(5):673–683, 2017.

[4] G. A. Al-Ghalith, E. Montassier, H. N. Ward, and D. Knights. NINJA-OPS: Fast Accurate Marker Gene Alignment using Concatenated Ribosomes. *PLoS Computational Biology*, 12(1):e1004658, 2016.

[5] I. Allali, J. W. Arnold, J. Roach, M. B. Cadenas, N. Butz, H. M. Hassan, M. Koci, A. Ballou, M. Mendoza, R. Ali, et al. A Comparison of Sequencing Platforms and Bioinformatics Pipelines for Compositional Analysis of the Gut Microbiome. *BMC Microbiology*, 17(1):194, 2017.

[6] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[7] S. V. Angiuoli, M. Matalka, A. Gussman, K. Galens, M. Vangala, D. R. Riley, C. Arze, J. R. White, O. White, and W. F. Fricke. CloVR: A Virtual Machine for Automated and Portable Sequence Analysis from the Desktop using Cloud Computing. *BMC Bioinformatics*, 12(1):356, 2011.

[8] E. Aronesty. ea-utils: Command-line tools for processing biological sequencing data, 2011.

[9] J. Åström, T. J. Pettersson, G. H. Reischer, T. Norberg, and M. Hermansson. Incorporating Expert Judgments in Utility Evaluation of *Bacteroidales* qPCR Assays for Microbial Source Tracking in a Drinking Water Source. *Environmental Science & Technology*, 49(3):1311–1318, 2015.

[10] R. Bain, R. Cronk, R. Hossain, S. Bonjour, K. Onda, J. Wright, H. Yang, T. Slaymaker, P. Hunter, A. Prüss-Ustün, et al. Global Assessment of Exposure to Faecal Contamination through Drinking Water Based on a Systematic Review. *Tropical Medicine & International Health*, 19(8):917–927, 2014.

[11] A. E. Bernhard and K. G. Field. A PCR Assay To Discriminate Human and Ruminant Feces on the Basis of Host Differences in *Bacteroides-Prevotella* Genes Encoding 16S rRNA. *Applied and Environmental Microbiology*, 66(10):4571–4574, 2000.

[12] G. Bitton. Microbial Source Tracking. In *Wastewater Microbiology (Fourth Edition)*, pages 197–215. John Wiley & Sons, Inc., 2010.

[13] M. W. Black, J. VanderKelen, A. Montana, A. Dekhtyar, E. Neal, A. Goodman, and C. L. Kitts. Pyroprinting: A rapid and flexible genotypic fingerprinting method for typing bacterial strains. *Journal of Microbiological Methods*, 105:121 – 129, 2014.

[14] A. B. Boehm, L. C. Van De Werfhorst, J. F. Griffith, P. A. Holden, J. A. Jay, O. C. Shanks, D. Wang, and S. B. Weisberg. Performance of Forty-one Microbial Source Tracking Methods: A Twenty-seven Lab Evaluation Study. *Water Research*, 47(18):6812–6828, 2013.

[15] N. A. Bokulich, S. Subramanian, J. J. Faith, D. Gevers, J. I. Gordon, R. Knight, D. A. Mills, and J. G. Caporaso. Quality-filtering Vastly Improves Diversity Estimates from Illumina Amplicon Sequencing. *Nature Methods*, 10(1):57–59, 2013.

[16] C. Brislawn. QIIME 1 Forum: vsearch for otu picking. https://groups.google.com/d/msg/qiime-forum/zCr6SVmtxCo/CJoD2VDrhVYJ, July 2015. Comment in QIIME 1 Forum. 16 July 2015. Google Groups. Date last accessed: 6/14/2020.

[17] C. Brislawn. QIIME 1 Forum: VSEARCH: chimera and otu picking - steps. https://groups.google.com/forum/#!topic/qiime-forum/zRiEF3wmtQo, April 2016. Comment in QIIME 1 Forum. 8 April 2016. Google Groups. Date last accessed: 8/29/2019.

[18] C. M. Brown, C. Staley, P. Wang, B. Dalzell, C. L. Chun, and M. J. Sadowsky. A High-Throughput DNA-Sequencing Approach for Determining Sources of Fecal Bacteria in a Lake Superior Estuary. *Environmental Science & Technology*, 51(15):8263–8271, 2017.

[19] G. Caporaso. An Introduction to Applied Bioinformatics. http://readiab.org/, May 2020. Online interactive text written by Greg Caporaso at Northern Arizona University. Date last accessed: 5/31/2020.

[20] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Pena, J. K. Goodrich, J. I. Gordon, et al. QIIME Allows Analysis of High-throughput Community Sequencing Data. *Nature Methods*, 7(5):335, 2010.

[21] CDC. Current Waterborne Disease Burden Data & Gaps.

https://www.cdc.gov/healthywater/burden/current-data.html. Accessed: 2020-04-01.

[22] M. Chatzou, C. Magis, J.-M. Chang, C. Kemena, G. Bussotti, I. Erb, and C. Notredame. Multiple Sequence Alignment Modeling: Methods and Applications. *Briefings in Bioinformatics*, 17(6):1009–1023, 2016.

[23] S.-Y. Chen, F. Deng, Y. Huang, X. Jia, Y.-P. Liu, and S.-J. Lai. bioOTU: An Improved Method for Simultaneous Taxonomic Assignments and Operational Taxonomic Units Clustering of 16s rRNA Gene Sequences. *Journal of Computational Biology*, 23(4):229–238, 2016.

[24] W. Chen, C. K. Zhang, Y. Cheng, S. Zhang, and H. Zhao. A Comparison of Methods for Clustering 16S rRNA Sequences into OTUs. *PloS One*, 8(8):e70837, 2013.

[25] X. Chen, S. Johnson, P. Jeraldo, J. Wang, N. Chia, J.-P. A. Kocher, and J. Chen. *Hybrid-denovo*: A *De novo* OTU-picking Pipeline Integrating Single-end and Paired-end 16S Sequence Tags. *Gigascience*, 7(3):gix129, 2018.

[26] J. Cline, J. C. Braman, and H. H. Hogrefe. PCR Fidelity of *Pfu* DNA Polymerase and other Thermostable DNA Polymerases. *Nucleic Acids Research*, 24(18):3546–3551, 1996.

[27] J. R. Cole, Q. Wang, E. Cardenas, J. Fish, B. Chai, R. J. Farris, A. Kulam-Syed-Mohideen, D. M. McGarrell, T. Marsh, G. M. Garrity, et al. The Ribosomal Database Project: Improved Alignments and New Tools for rRNA Analysis. *Nucleic Acids Research*, 37(suppl_1):D141–D145, 2009.

[28] A. M. Comeau, G. M. Douglas, and M. G. Langille. Microbiome Helper: A Custom and Streamlined Workflow for Microbiome Research. *mSystems*, 2(1), 2017.

[29] S. DeFlorio-Barker, C. Wing, R. M. Jones, and S. Dorevitch. Estimate of incidence and cost of recreational waterborne illness on United States surface waters. *Environmental Health*, 17(1):3, 2018.

[30] T. Z. DeSantis, P. Hugenholtz, N. Larsen, M. Rojas, E. L. Brodie, K. Keller, T. Huber, D. Dalevi, P. Hu, and G. L. Andersen. Greengenes, a Chimera-Checked 16S rRNA Gene Database and Workbench Compatible with ARB. *Applied and Environmental Microbiology*, 72(7):5069–5072, 2006.

[31] L. K. Dick, A. E. Bernhard, T. J. Brodeur, J. W. Santo Domingo, J. M. Simpson, S. P. Walters, and K. G. Field. Host Distributions of Uncultivated Fecal *Bacteroidales* Bacteria Reveal Genetic Markers for Fecal Source Identification. *Appl. Environ. Microbiol.*, 71(6):3184–3191, 2005.

[32] K. Duncan. Microbiome/Metagenome Analysis Workshop: QIIME. https://youtu.be/nWeRN2lKIto?t=1, April 2018. YouTube. Brown University. Date last accessed: 3/28/2020.

[33] R. Edgar. USEARCH Online Manual. https://drive5.com/usearch/manual/. drive5 online manual. Date last accessed: 6/14/2020.

[34] R. C. Edgar. MUSCLE: Multiple Sequence Alignment with High Accuracy and High Throughput. *Nucleic Acids Research*, 32(5):1792–1797, 2004.

[35] R. C. Edgar. Search and Clustering Orders of Magnitude Faster than BLAST. *Bioinformatics*, 26(19):2460–2461, 2010.

[36] R. C. Edgar. UPARSE: Highly Accurate OTU Sequences from Microbial Amplicon Reads. *Nature Methods*, 10(10):996–998, 2013.

[37] R. C. Edgar. Accuracy of Microbial Community Diversity Estimated by Closed- and Open-Reference OTUs. *PeerJ*, 5:e3889, 2017.

[38] R. C. Edgar, B. J. Haas, J. C. Clemente, C. Quince, and R. Knight. UCHIME Improves Sensitivity and Speed of Chimera Detection. *Bioinformatics*, 27(16):2194–2200, 2011.

[39] R. Edwards. Computational Genomics. https://linsalrob.github.io/ComputationalGenomicsManual/, 2018. San Diego State University Course. Date last accessed: 5/24/2020.

[40] R. Edwards. Illumina Paired End Sequencing. https://www.youtube.com/watch?v=WneZp3fSJIk&list=PLpPXw4zFa0uLMHwSZ7DMeLGjIUgo1IBbn&index=15, November 2018. YouTube. San Diego State University. Date last accessed: 5/2/2020.

[41] S. El-Metwally, O. M. Ouda, and M. Helmy. Algorithms and Data Structures in Next-Generation Sequencing. In *Next Generation Sequencing Technologies and Challenges in Sequence Assembly*, pages 15–26. Springer, 2014.

[42] S. El-Metwally, O. M. Ouda, and M. Helmy. Basics of Molecular Biology for Next-Generation Sequencing. In *Next Generation Sequencing Technologies and Challenges in Sequence Assembly*, pages 3–13. Springer, 2014.

[43] R. T. Espejo and N. Plaza. Multiple Ribosomal RNA Operons in Bacteria; Their Concerted Evolution and Potential Consequences on the Rate of Evolution of Their 16S rRNA. *Frontiers in Microbiology*, 9:1232, 2018.

[44] B. Ewing and P. Green. Base-Calling of Automated Sequencer Traces using Phred. II. Error Probabilities. *Genome Research*, 8(3):186–194, 1998.

[45] M. Fernandez, V. Aguiar-Pulido, J. Riveros, W. Huang, J. Segal, E. Zeng, M. Campos, K. Mathee, and G. Narasimhan. Microbiome Analysis: State of the Art and Future Trends. In *Computational Methods for Next Generation Sequencing Data Analysis*, pages 401–424. Wiley Online Library, 2016.

[46] L. Fewtrell and D. Kay. Recreational Water and Infection: A Review of Recent Findings. *Current Environmental Health Reports*, 2(1):85–94, 2015.

[47] K. G. Field and M. Samadpour. Fecal Source Tracking, the Indicator Paradigm, and Managing Water Quality. *Water Research*, 41(16):3517–3538, 2007.

[48] L. R. Fogarty and M. A. Voytek. Comparison of Bacteroides-Prevotella 16S rRNA Genetic Markers for Fecal Samples from Different Animal Species. *Appl. Environ. Microbiol.*, 71(10):5999–6007, 2005.

[49] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni. Tools for Mapping High-Throughput Sequencing Data. *Bioinformatics*, 28(24):3169–3177, 2012.

[50] N. O. Forstinus, N. E. Ikechukwu, M. P. Emenike, and A. O. Christiana. Water and Waterborne Diseases: A Review. *International Journal of Tropical Diseases and Health*, 12(4):1–14, 2016.

[51] O. Franzén, J. Hu, X. Bao, S. H. Itzkowitz, I. Peter, and A. Bashir. Improved OTU-Picking using Long-read 16S rRNA Gene Amplicon Sequencing and Generic Hierarchical Clustering. *Microbiome*, 3(1):43, 2015.

[52] B. Fremaux, T. Boa, and C. K. Yost. Quantitative Real-Time PCR Assays for Sensitive Detection of Canada Goose-Specific Fecal Pollution in Water Sources. *Appl. Environ. Microbiol.*, 76(14):4886–4889, 2010.

[53] L.-L. Fu and J.-R. Li. Microbial Source Tracking: A Tool for Identifying Sources of Microbial Contamination in the Food Chain. *Critical Reviews in Food Science and Nutrition*, 54(6):699–707, 2014.

[54] J. Gargano, E. Adam, S. Collier, K. Fullerton, S. Feinman, and M. Beach.

Mortality from Selected Diseases that can be Transmitted by Water–United States, 2003–2009. *Journal of Water and Health*, 15(3):438–450, 2017.

[55] C. Glaser, E. Powers, and C. Greene. Zoonotic Infections of Medical Importance in Immunocompromised Humans. In *Infectious Diseases of the Dog and Cat (Fourth Edition)*, pages 1141–62. St Louis: Elsevier Saunders, 2012.

[56] R. Gomi, T. Matsuda, Y. Matsui, and M. Yoneda. Fecal Source Tracking in Water by Next-Generation Sequencing Technologies Using Host-Specific *Escherichia coli* Genetic Markers. *Environmental Science & Technology*, 48(16):9616–9623, 2014.

[57] A. Gonzalez, J. A. Navas-Molina, T. Kosciolek, D. McDonald, Y. Vázquez-Baeza, G. Ackermann, J. DeReus, S. Janssen, A. D. Swafford, S. B. Orchanian, et al. Qiita: Rapid, Web-Enabled Microbiome Meta-Analysis. *Nature Methods*, 15(10):796–798, 2018.

[58] J. K. Goodrich, S. C. Di Rienzi, A. C. Poole, O. Koren, W. A. Walters, J. G. Caporaso, R. Knight, and R. E. Ley. Conducting a Microbiome Study. *Cell*, 158(2):250–262, 2014.

[59] A. Grada and K. Weinbrecht. Next-Generation Sequencing: Methodology and Application. *The Journal of Investigative Dermatology*, 133(8):e11, 2013.

[60] C. Hagedorn, A. R. Blanch, and V. J. Harwood. *Microbial Source Tracking: Methods, Applications, and Case Studies*. Springer Science & Business Media, 2011.

[61] J. Handelsman. Metagenomics: Application of Genomics to Uncultured Microorganisms. *Microbiol. Mol. Biol. Rev.*, 68(4):669–685, 2004.

[62] V. J. Harwood, C. Staley, B. D. Badgley, K. Borges, and A. Korajkic. Microbial Source Tracking Markers for Detection of Fecal Contamination in Environmental Waters: Relationships Between Pathogens and Human Health Outcomes. *FEMS Microbiology Reviews*, 38(1):1–40, 2014.

[63] Y. He, J. G. Caporaso, X.-T. Jiang, H.-F. Sheng, S. M. Huse, J. R. Rideout, R. C. Edgar, E. Kopylova, W. A. Walters, R. Knight, et al. Stability of Operational Taxonomic Units: An Important but Neglected Property for Analyzing Microbial Diversity. *Microbiome*, 3(1):20, 2015.

[64] F. Hildebrand, R. Tadeo, A. Y. Voigt, P. Bork, and J. Raes. LotuS: An Efficient and User-Friendly OTU Processing Pipeline. *Microbiome*, 2(1):30, 2014.

[65] S. M. Huse, D. M. Welch, H. G. Morrison, and M. L. Sogin. Ironing out the Wrinkles in the Rare Biosphere through Improved OTU Clustering. *Environmental Microbiology*, 12(7):1889–1898, 2010.

[66] Illumina. MiniSeq™ Sequencing System. https://science-docs.illumina.com/documents/Instruments/miniseq-system-spec-sheet-770-2015-039/miniseq-system-spec-sheet-770-2015-039.pdf. Illumina. Date last accessed: 5/3/2020.

[67] Illumina. MiSeq™ Sequencing System. https://www.illumina.com/content/dam/illumina-marketing/documents/products/datasheets/datasheet_miseq.pdf. Illumina. Date last accessed: 10/10/2020.

[68] Illumina. Quality Scores for Next-Generation Sequencing. https://www.illumina.com/content/dam/illumina-

marketing/documents/products/technotes/technote_Q-Scores.pdf, October 2011. Technical Note. Illumina. Date last accessed: 5/10/2020.

[69] Illumina. An Introduction to Next-Generation Sequencing Technology. https://www.illumina.com/content/dam/illumina-marketing/documents/products/illumina_sequencing_introduction.pdf, 2015. Illumina. Date last accessed: 5/3/2020.

[70] M. A. Jackson, J. T. Bell, T. D. Spector, and C. J. Steves. A Heritability-Based Comparison of Methods used to Cluster 16S rRNA Gene Sequences into Operational Taxonomic Units. *PeerJ*, 4:e2341, 2016.

[71] P. Jeraldo, K. Kalari, X. Chen, J. Bhavsar, A. Mangalam, B. White, H. Nelson, J.-P. Kocher, and N. Chia. IM-TORNADO: A Tool for Comparison of 16S Reads from Paired-End Libraries. *PloS One*, 9(12), 2014.

[72] X.-T. Jiang, H. Zhang, H.-F. Sheng, Y. Wang, Y. He, F. Zou, and H.-W. Zhou. Two-Stage Clustering (TSC): A Pipeline for Selecting Operational Taxonomic Units for the High-Throughput Sequencing of PCR Amplicons. *PLoS One*, 7(1):e30230, 2012.

[73] E. Johnson. Density-Based Clustering of High-Dimensional DNA Fingerprints for Library-Dependent Microbial Source Tracking. Master's thesis, California Polytechnic State University San Luis Obispo, 2015.

[74] N. C. Jones and P. A. Pevzner. Dynamic Programming Algorithms: Edit Distance and Alignments. In *An Introduction to Bioinformatics Algorithms*, pages 167–185. MIT press, 2004.

[75] F. Ju and T. Zhang. 16S rRNA Gene High-Throughput Sequencing Data Mining of Microbial Diversity and Interactions. *Applied Microbiology and Biotechnology*, 99(10):4119–4129, 2015.

[76] B. J. Kildare, C. M. Leutenegger, B. S. McSwain, D. G. Bambic, V. B. Rajal, and S. Wuertz. 16S rRNA-Based Assays for Quantitative Detection of Universal, Human-, Cow-, and Dog-Specific Fecal *Bacteroidales*: A Bayesian Approach. *Water Research*, 41(16):3701–3715, 2007.

[77] M. Kirs, R. A. Caffaro-Filho, M. Wong, V. J. Harwood, P. Moravcik, and R. S. Fujioka. Human-associated *Bacteroides* spp. and Human Polyomaviruses as Microbial Source Tracking Markers in Hawaii. *Applied and Environmental Microbiology*, 82(22):6757–6767, 2016.

[78] D. Knights. Microbiome Discovery 4: QIIME. https://youtu.be/iy0JWgzmM_A, January 2016. YouTube. University of Minnesota. Date last accessed: 3/28/2020.

[79] D. Knights. Microbiome Discovery 5: Picking OTUs. https://youtu.be/Ok5h24KZbAE, January 2016. YouTube. University of Minnesota. Date last accessed: 3/28/2020.

[80] J. K. Kulski. Next-Generation Sequencing—An Overview of the History, Tools, and "Omic" Applications. *Next Generation Sequencing–Advances, Applications and Challenges*, pages 3–60, 2016.

[81] T. Lai. Enhancements to the Microbial Source Tracking Process Through the Utilization of Clustering and k-Nearest Clusters Algorithm. Master's thesis, California Polytechnic State University San Luis Obispo, 2018.

[82] A. Layton, L. McKay, D. Williams, V. Garrett, R. Gentry, and G. Sayler. Development of *Bacteroides* 16S rRNA Gene TaqMan-Based Real-Time PCR Assays for Estimation of Total, Human, and Bovine Fecal Pollution in Water. *Applied and Environmental Microbiology*, 72(6):4214–4224, 2006.

[83] B. A. Layton, Y. Cao, D. L. Ebentier, K. Hanley, E. Ballesté, J. Brandão, M. Byappanahalli, R. Converse, A. H. Farnleitner, J. Gentry-Shields, et al. Performance of Human Fecal Anaerobe-Associated PCR-based Assays in a Multi-Laboratory Method Evaluation Study. *Water Research*, 47(18):6897–6908, 2013.

[84] H. Leclerc, L. Schwartzbrod, and E. Dei-Cas. Microbial Agents Associated with Waterborne Diseases. *Critical Reviews in Microbiology*, 28(4):371–409, 2002.

[85] W. Li and A. Godzik. Cd-hit: a Fast Program for Clustering and Comparing Large Sets of Protein or Nucleotide Sequences. *Bioinformatics*, 22(13):1658–1659, 2006.

[86] M. G. Links, B. Chaban, S. M. Hemmingsen, K. Muirhead, and J. E. Hill. mPUMA: A Computational Approach to Microbiota Analysis by *de novo* Assembly of Operational Taxonomic Units based on Protein-coding Barcode Sequences. *Microbiome*, 1(1):23, 2013.

[87] J. Lu, J. Santo Domingo, and O. C. Shanks. Identification of Chicken-Specific Fecal Microbial Sequences using a Metagenomic Approach. *Water Research*, 41(16):3561–3574, 2007.

[88] J. Lu, J. W. Santo Domingo, R. Lamendella, T. Edge, and S. Hill. Phylogenetic Diversity and Molecular Detection of Bacteria in Gull Feces. *Appl. Environ. Microbiol.*, 74(13):3969–3976, 2008.

[89] F. Mahé, T. Rognes, C. Quince, C. de Vargas, and M. Dunthorn. Swarm: Robust and Fast Clustering Method for Amplicon-Based Studies. *PeerJ*, 2:e593, 2014.

[90] F. Mahé. QIIME 1 Forum: VSEARCH: chimera and OTU picking. https://groups.google.com/d/msg/vsearch-forum/Hh-AaYVTQpg/UVizu7iaBAAJ, April 2016. Comment in QIIME 1 Forum. 7 April 2016. Google Groups. Date last accessed: 6/14/2020.

[91] R. Marti, Y. Zhang, D. R. Lapen, and E. Topp. Development and Validation of a Microbial Source Tracking Marker for the Detection of Fecal Pollution by Muskrats. *Journal of Microbiological Methods*, 87(1):82–88, 2011.

[92] R. Marti, Y. Zhang, Y.-C. Tien, D. R. Lapen, and E. Topp. Assessment of a New *Bacteroidales* Marker Targeting North American Beaver (*Castor canadensis*) Fecal Pollution by Real-Time PCR. *Journal of Microbiological Methods*, 95(2):201–206, 2013.

[93] A. P. Masella, A. K. Bartram, J. M. Truszkowski, D. G. Brown, and J. D. Neufeld. PANDAseq: Paired-end Assembler for Illumina Sequences. *BMC Bioinformatics*, 13(1):31, 2012.

[94] J. McGovern. Investigating the k-Nearest Neighbors Resolution Algorithms for Pyroprints and Clustering for Bacterial Strains. Master's thesis, California Polytechnic State University San Luis Obispo, 2016.

[95] S. L. McLellan and A. M. Eren. Discovering New Indicators of Fecal Pollution. *Trends in Microbiology*, 22(12):697–706, 2014.

[96] C. Mercier, F. Boyer, A. Bonin, and E. Coissac. SUMATRA and SUMACLUST: Fast and Exact Comparison and Clustering of Sequences. In *Programs and Abstracts of the SeqBio 2013 Workshop. Abstract*, pages 27–29. Citeseer, 2013.

[97] M. L. Metzker and C. T. Caskey. Polymerase Chain Reaction (PCR). *eLS*, 2009.

[98] S. Mieszkin, J.-P. Furet, G. Corthier, and M. Gourmelon. Estimation of Pig Fecal Contamination in a River Catchment by Real-Time PCR using Two Pig-Specific *Bacteroidales* 16S rRNA Genetic Markers. *Appl. Environ. Microbiol.*, 75(10):3045–3054, 2009.

[99] J. R. Miller, S. Koren, and G. Sutton. Assembly Algorithms for Next-Generation Sequencing Data. *Genomics*, 95(6):315–327, 2010.

[100] A. Montana. Algorithms for Library-based Microbial Source Tracking. Master's thesis, California Polytechnic State University San Luis Obispo, 2013.

[101] J. Mott and A. Smith. Library-Dependent Source Tracking Methods. In *Microbial Source Tracking: Methods, Applications, and Case Studies*, pages 31–60. Springer, 2011.

[102] M. Mysara, M. Njima, N. Leys, J. Raes, and P. Monsieurs. From Reads to Operational Taxonomic Units: An Ensemble Processing Pipeline for MiSeq Amplicon Sequencing Data. *Gigascience*, 6(2):giw017, 2017.

[103] J. A. Navas-Molina, J. M. Peralta-Sánchez, A. González, P. J. McMurdie, Y. Vázquez-Baeza, Z. Xu, L. K. Ursell, C. Lauber, H. Zhou, S. J. Song, et al. Advancing Our Understanding of the Human Microbiome Using QIIME. In *Methods in Enzymology*, volume 531, pages 371–444. Elsevier, 2013.

[104] S. B. Needleman and C. D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[105] N.-P. Nguyen, T. Warnow, M. Pop, and B. White. A Perspective on 16S rRNA Operational Taxonomic Unit Clustering using Sequence Similarity. *NPJ Biofilms and Microbiomes*, 2(1):1–8, 2016.

[106] S. Okabe, N. Okayama, O. Savichtcheva, and T. Ito. Quantification of Host-Specific *Bacteroides–Prevotella* 16S rRNA Genetic Markers for Assessment of Fecal Pollution in Freshwater. *Applied Microbiology and Biotechnology*, 74(4):890–901, 2007.

[107] A. Oulas, C. Pavloudi, P. Polymenakou, G. A. Pavlopoulos, N. Papanikolaou, G. Kotoulas, C. Arvanitidis, and L. Iliopoulos. Metagenomics: Tools and Insights for Analyzing Next-Generation Sequencing Data Derived from Biodiversity Studies. *Bioinformatics and Biology Insights*, 9:BBI–S12462, 2015.

[108] P. K. Pandey, P. H. Kass, M. L. Soupir, S. Biswas, and V. P. Singh. Contamination of Water Resources by Pathogenic Bacteria. *AMB Express*, 4(1):51, 2014.

[109] L. Paruch, A. M. Paruch, A.-G. Buseth Blankenberg, M. Bechmann, and T. Mæhlum. Application of Host-Specific Genetic Markers for Microbial Source Tracking of Faecal Water Contamination in an Agricultural Catchment. *Acta Agriculturae Scandinavica, Section B—Soil & Plant Science*, 65(sup2):164–172, 2015.

[110] C. Poussin, N. Sierro, S. Boué, J. Battey, E. Scotti, V. Belcastro, M. C. Peitsch, N. V. Ivanov, and J. Hoeng. Interrogating the Microbiome: Experimental and Computational Considerations in Support of Study Reproducibility. *Drug Discovery Today*, 23(9):1644–1657, 2018.

[111] S. P. Preheim, A. R. Perrotta, A. M. Martin-Platero, A. Gupta, and E. J. Alm. Distribution-Based Clustering: Using Ecology to Refine the Operational Taxonomic Unit. *Appl. Environ. Microbiol.*, 79(21):6593–6603, 2013.

[112] V. S. Pylro, L. F. W. Roesch, D. K. Morais, I. M. Clark, P. R. Hirsch, and

M. R. Tótola. Data Analysis for 16S Microbial Profiling from Different Benchtop Sequencing Platforms. *Journal of Microbiological Methods*, 107:30–37, 2014.

[113] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner. The SILVA ribosomal RNA Gene Database Project: Improved Data Processing and Web-based Tools. *Nucleic Acids Research*, 41(D1):D590–D596, 2012.

[114] M. R. Raith, C. A. Kelty, J. F. Griffith, A. Schriewer, S. Wuertz, S. Mieszkin, M. Gourmelon, G. H. Reischer, A. H. Farnleitner, J. S. Ervin, et al. Comparison of PCR and Quantitative Real-Time PCR Methods for the Characterization of Ruminant and Cattle Fecal Pollution Sources. *Water Research*, 47(18):6921–6928, 2013.

[115] J. Ramiro-Garcia, G. D. Hermes, C. Giatsis, D. Sipkema, E. G. Zoetendal, P. J. Schaap, and H. Smidt. NG-Tax, A Highly Accurate and Validated Pipeline for Analysis of 16S rRNA Amplicons from Complex Biomes. *F1000Research*, 5, 2016.

[116] K. Ravaliya, J. Gentry-Shields, S. Garcia, N. Heredia, A. F. de Aceituno, F. E. Bartz, J. S. Leon, and L.-A. Jaykus. Use of *Bacteroidales* Microbial Source Tracking to Monitor Fecal Contamination in Fresh Produce Production. *Applied and Environmental Microbiology*, 80(2):612–617, 2014.

[117] G. H. Reischer, J. E. Ebdon, J. M. Bauer, N. Schuster, W. Ahmed, J. Åström, A. R. Blanch, G. Blöschl, D. Byamukama, T. Coakley, et al. Performance Characteristics of qPCR Assays Targeting Human-and Ruminant-Associated *Bacteroidetes* for Microbial Source Tracking Across Sixteen Countries on Six Continents. *Environmental Science & Technology*, 47(15):8548–8556, 2013.

[118] J. A. Reuter, D. V. Spacek, and M. P. Snyder. High-Throughput Sequencing Technologies. *Molecular Cell*, 58(4):586–597, 2015.

[119] J. R. Rideout, Y. He, J. A. Navas-Molina, W. A. Walters, L. K. Ursell, S. M. Gibbons, J. Chase, D. McDonald, A. Gonzalez, A. Robbins-Pianka, et al. Subsampled Open-Reference Clustering Creates Consistent, Comprehensive OTU Definitions and Scales to Billions of Sequences. *PeerJ*, 2:e545, 2014.

[120] K. Riehle, C. Coarfa, A. Jackson, J. Ma, A. Tandon, S. Paithankar, S. Raghuraman, T.-A. Mistretta, D. Saulnier, S. Raza, et al. The Genboree Microbiome Toolset and the Analysis of 16S rRNA Microbial Sequences. *BMC Bioinformatics*, 13(13):1–11, 2012.

[121] C. Rock, B. Rivera, and C. P. Gerba. Microbial Source Tracking. In *Environmental Microbiology (Third Edition)*, pages 309–317. Elsevier, 2015.

[122] T. Rognes. VSEARCH Source Code. https://github.com/torognes/vsearch. Github Repo. Date last accessed: 6/7/2020.

[123] T. Rognes, T. Flouri, B. Nichols, C. Quince, and F. Mahé. VSEARCH: A Versatile Open Source Tool for Metagenomics. *PeerJ*, 4:e2584, 2016.

[124] M. S. Rosenberg. Sequence Alignment: Concepts and History. In *Sequence Alignment: Methods, Models, Concepts, and Strategies*, pages 1–22. Univ of California Press, 2009.

[125] F. Sanger, S. Nicklen, and A. R. Coulson. DNA Sequencing with Chain-terminating Inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467, 1977.

[126] D. Sargeant, W. R. Kammin, and S. Collyard. *Review and Critique of Current*

Microbial Source Tracking (MST) Techniques. Environmental Assessment Program, Washington State Department of Ecology, 2011.

[127] E. P. Sauer, J. L. VandeWalle, M. J. Bootsma, and S. L. McLellan. Detection of the Human Specific *Bacteroides* Genetic Marker Provides Evidence of Widespread Sewage Contamination of Stormwater in the Urban Environment. *Water Research*, 45(14):4081–4091, 2011.

[128] P. D. Schloss. Reintroducing mothur: 10 Years Later. *Applied and Environmental Microbiology*, 86(2), 2020.

[129] P. D. Schloss, D. Gevers, and S. L. Westcott. Reducing the Effects of PCR Amplification and Sequencing Artifacts on 16S rRNA-based Studies. *PloS One*, 6(12), 2011.

[130] P. D. Schloss and J. Handelsman. Introducing DOTUR, a Computer Program for Defining Operational Taxonomic Units and Estimating Species Richness. *Applied and Environmental Microbiology*, 71(3):1501–1506, 2005.

[131] P. D. Schloss and S. L. Westcott. Assessing and Improving Methods used in Operational Taxonomic Unit-Based Approaches for 16S rRNA Gene Sequence Analysis. *Appl. Environ. Microbiol.*, 77(10):3219–3226, 2011.

[132] P. D. 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, et al. Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities. *Applied and Environmental Microbiology*, 75(23):7537–7541, 2009.

[133] E. Schroeder and S. Wuertz. Chapter 3: Bacteria. In *Handbook of Water and Wastewater Microbiology*, pages 57–68. Elsevier, 2003.

[134] T. M. Scott, J. B. Rose, T. M. Jenkins, S. R. Farrah, and J. Lukasik. Microbial Source Tracking: Current Methodology and Future Directions. *Applied and Environmental Microbiology*, 68(12):5796–5803, 2002.

[135] S. Seurinck, T. Defoirdt, W. Verstraete, and S. D. Siciliano. Detection and Quantification of the Human-specific HF183 *Bacteroides* 16S rRNA Genetic Marker with Real-time PCR for Assessment of Human Faecal Pollution in Freshwater. *Environmental Microbiology*, 7(2):249–259, 2005.

[136] O. C. Shanks, K. White, C. A. Kelty, S. Hayes, M. Sivaganesan, M. Jenkins, M. Varma, and R. A. Haugland. Performance Assessment PCR-Based Assays Targeting *Bacteroidales* Genetic Markers of Bovine Fecal Pollution. *Appl. Environ. Microbiol.*, 76(5):1359–1366, 2010.

[137] O. C. Shanks, K. White, C. A. Kelty, M. Sivaganesan, J. Blannon, M. Meckes, M. Varma, and R. A. Haugland. Performance of PCR-Based Assays Targeting *Bacteroidales* Genetic Markers of Human Fecal Pollution in Sewage and Fecal Samples. *Environmental Science & Technology*, 44(16):6281–6288, 2010.

[138] J. M. Simpson, J. W. Santo Domingo, and D. J. Reasoner. Microbial Source Tracking: State of the Science. *Environmental Science & Technology*, 36(24):5279–5288, 2002.

[139] J. Soh, X. Dong, S. M. Caffrey, G. Voordouw, and C. W. Sensen. Phoenix 2: A Locally Installable Large-scale 16S rRNA Gene Sequence Analysis Pipeline with Web Interface. *Journal of Biotechnology*, 167(4):393–403, 2013.

[140] J. L. Soliman, A. Dekhtyar, J. Vanderkellen, A. Montana, M. Black, E. Neal, K. Webb, C. Kitts, and A. Goodman. Microbial Source Tracking by Molecular Fingerprinting. In *Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 617–619. ACM, 2012.

[141] J. L. V. Soliman. CPLOP: The Cal Poly Library Of Pyroprints. Master's thesis, California Polytechnic State University San Luis Obispo, 2013.

[142] D. M. Stoeckel and V. J. Harwood. Performance, Design, and Analysis in Microbial Source Tracking Studies. *Applied and Environmental Microbiology*, 73(8):2405–2415, 2007.

[143] Y. Sun, Y. Cai, S. M. Huse, R. Knight, W. G. Farmerie, X. Wang, and V. Mai. A Large-scale Benchmark Study of Existing Algorithms for Taxonomy-Independent Microbial Community Analysis. *Briefings in Bioinformatics*, 13(1):107–121, 2012.

[144] Y. Sun, Y. Cai, L. Liu, F. Yu, M. L. Farrell, W. McKendree, and W. Farmerie. ESPRIT: Estimating Species Richness using Large Collections of 16S rRNA Pyrosequences. *Nucleic Acids Research*, 37(10):e76–e76, 2009.

[145] D. D. Tambalo, B. Fremaux, T. Boa, and C. K. Yost. Persistence of Host-Associated *Bacteroidales* Gene Markers and their Quantitative Detection in an Urban and Agricultural Mixed Prairie Watershed. *Water Research*, 46(9):2891–2904, 2012.

[146] B. Tan, C. M. Ng, J. P. Nshimyimana, L.-L. Loh, K. Y.-H. Gin, and J. R. Thompson. Next-Generation Sequencing (NGS) for Assessment of Microbial Water Quality: Current Progress, Challenges, and Future Opportunities. *Frontiers in Microbiology*, 6:1027, 2015.

[147] P.-N. Tan, M. Steinbach, A. Karpatne, and V. Kumar. Chapter 8 Cluster Analysis: Basic Concepts and Algorithms . In *Introduction to Data Mining (Second Edition)*, pages 487–568. Pearson Education India, 2019.

[148] T. Unno, C. Staley, C. M. Brown, D. Han, M. J. Sadowsky, and H.-G. Hur. Fecal Pollution: New Trends and Challenges in Microbial Source Tracking

using Next-Generation Sequencing. *Environmental Microbiology*, 20(9):3132–3140, 2018.

[149] X. Wang, J. Yao, Y. Sun, and V. Mai. M-pick, A Modularity-Based Method for OTU Picking of 16S rRNA Sequences. *BMC Bioinformatics*, 14(1):43, 2013.

[150] Z.-G. Wei, S.-W. Zhang, and Y.-Z. Zhang. DMclust, a Density-based Modularity Method for Accurate OTU Picking of 16S rRNA Sequences. *Molecular Informatics*, 36(12):1600059, 2017.

[151] J. J. Werner, D. Zhou, J. G. Caporaso, R. Knight, and L. T. Angenent. Comparison of Illumina Paired-end and Single-direction Sequencing for Microbial 16S rRNA Gene Amplicon Surveys. *The ISME Journal*, 6(7):1273–1276, 2012.

[152] S. L. Westcott and P. D. Schloss. De novo Clustering Methods Outperform Reference-based Methods for Assigning 16S rRNA Gene Sequences to Operational Taxonomic Units. *PeerJ*, 3:e1487, 2015.

[153] S. L. Westcott and P. D. Schloss. OptiClust, an Improved Method for Assigning Amplicon-Based Sequence Data to Operational Taxonomic Units. *MSphere*, 2(2), 2017.

[154] World Health Organization and others. *Preventing Diarrhoea through Better Water, Sanitation and Hygiene: Exposures and Impacts in Low-and Middle-Income Countries*. World Health Organization, 2014.

[155] World Health Organization and others. *Water, Sanitation, Hygiene and Health: A Primer for Health Professionals*. World Health Organization, 2019. No. WHO/CED/PHE/WSH/19.149.

[156] D. Wu, L. Doroud, and J. A. Eisen. TreeOTU: Operational Taxonomic Unit Classification Based on Phylogenetic Trees. *arXiv preprint arXiv:1308.6333*, 2013.

[157] S. Wuertz, D. Wang, G. H. Reischer, and A. H. Farnleitner. Library-Independent Bacterial Source Tracking Methods. In *Microbial Source Tracking: Methods, Applications, and Case Studies*, pages 61–112. Springer, 2011.

[158] D. Yadav, A. Dutta, and S. S. Mande. OTUX: V-region Specific OTU Database for Improved 16S rRNA OTU Picking and Efficient Cross-Study Taxonomic Comparison of Microbiomes. *DNA Research*, 26(2):147–156, 2019.

[159] Q. Zheng, C. Bartow-McKenney, J. S. Meisel, and E. A. Grice. HmmUFOtu: An HMM and Phylogenetic Placement Based Ultra-fast Taxonomic Assignment and OTU Picking Tool for Microbiome Amplicon Sequencing Studies. *Genome Biology*, 19(1):82, 2018.

APPENDICES

Appendix A

QUALITY SCORES

In the sequencing step of Illumina's Sequencing By Synthesis process, the platform identifies a base from the emission of fluorescently labeled nucleotides [69]. The base is inferred (or "called") from the wavelength and intensity of the emission. Anything that effects the light signal can change which base is called, potentially introducing an error in sequencing. Quality scores are a way for the sequencer to indicate the confidence that the correct base was called at a specific position.

The Phred quality score (Q score) is the metric used to evaluate the accuracy of a sequencing platform [68]. High quality scores ensure that the output read is representative of the true biological sequence and not the result of a sequencing error. Quality scores (Q) are calculated as:

$$Q = -10\ log_{10}\ P \tag{A.1}$$

and equivalently:

$$P = 10^{\frac{-Q}{10}} \tag{A.2}$$

where P is the base calling error probability (i.e., the probability that the base is wrong) [44].

A quality score of 30 has a P of 0.001 which means the chance that the sequencer called the wrong base is 0.1% or conversely, the chance that it was correct is 99.9%. A base call accuracy of 99.9% is equivalent to an incorrect base call 1 in 1,000 times [68]. In other words, for every 1,000 bp read, there is likelihood of one incorrect base pair. A quality score of 20 (99.0% accuracy) means for every 100 bp read, there is

one incorrect base pair.

Quality scores can be stored numerically (`qual` files) or ASCII encoded (`fastq` files). Fastq files use ASCII encoding to compress the quality scores and reduce the overall file size. Figure 3.6a shows a fastq file entry. Line 4 contains the quality values for each nucleotide base in that read (the sequence in Line 2). The encoded scores are represented by ASCII characters and can be converted to Q scores using Phred-33 encoding:

$$Q\ score = ASCII\ value - 33 \tag{A.3}$$

The encoding can be understood using examples from a simple fastq entry:

```
@Sample1Identifier
ATGTGATC
+
+*0)'))I
```

Example 1: Line 4 starts with "`+`", so the Q score for the first nucleotide in Line 2 "`A`" is calculated as follows: the ASCII value of "`+`" is 43, so $43 - 33 = 10$. A quality score of 10 is low quality ($P = 0.1$, accuracy = 90%) and indicates the sequencer likely made a mistake in 1 out of every 10 bases (i.e., the sequencer is not confident that this "`A`" is the correct base).

Example 2: the encoded ASCII quality score for the last nucleotide "`C`" is "`I`". The ASCII value of "`I`" is 73, making the quality score $73 - 33 = 40$. In this case, a high quality score of 40 means the probability of an error is only 0.0001 (or the sequencer is 99.99% confident that the nucleotide "`C`" is the correct base).

Quality scores are used to evaluate the correctness of a read and reduce errors in estimating diversity [15]. High quality reads are kept for analysis while low quality reads are truncated or discarded. Quality filtering software uses a "sliding window" technique to evaluate overall read quality [32]. A window size of 10 means that 10 nucleotides are used in calculating an average quality score for that window. If the

average quality score falls below a cutoff threshold, the read is truncated at that window, meaning only the nucleotides in the sequence up to that window are kept. Many reads have a reduction in quality toward the end of the read. The sliding window technique allows such reads that are higher quality at the beginning to be included in the analysis. The pipeline in LOTUS uses a threshold of 30 as the default quality score, although investigators have the option to change this parameter at their discretion.

The Phred-33 encoding ASCII conversion table is shown in Table A.1.

Table A.1: ASCII conversion table of Phred-33 quality scores.

ASCII Symbol	ASCII Value	Q Score	ASCII Symbol	ASCII Value	Q Score
!	33	0	6	54	21
”	34	1	7	55	22
#	35	2	8	56	23
$	36	3	9	57	24
%	37	4	:	58	25
&	38	5	;	59	26
’	39	6	<	60	27
(	40	7	=	61	28
)	41	8	>	62	29
*	42	9	?	63	30
+	43	10	@	64	31
,	44	11	A	65	32
-	45	12	B	66	33
.	46	13	C	67	34
\	47	14	D	68	35
0	48	15	E	69	36
1	49	16	F	70	37
2	50	17	G	71	38
3	51	18	H	72	39
4	52	19	I	73	40
5	53	20	J	74	41

Appendix B

SEQUENCE ALIGNMENT

A common analysis in biology is comparing query sequences to each other or to a reference database in order to understand the function or relatedness of the sequences [19]. For example, to determine the function of a new gene, researchers compare how similar it is to a previously studied gene with a known function [41]. In the context of the OBMM presented in this thesis, biologists group similar sequences together into OTUs with the assumption that related sequences come from the same phylogenetic group. Determining functionality or relatedness is accomplished by measuring the similarity of the sequences.

Similarity is measured by comparing nucleotides between sequences. Sequences that have the same nucleotides in the same positions are assumed to be more similar, and hence more related, than sequences with nucleotides in different positions. However, direct comparison of biological sequences may not provide an accurate representation of similarity. During cell replication, DNA sequences are subject to insertion, deletion, and substitution operations that can result in slight differences between two otherwise related sequences [74].

An insertion occurs when a nucleotide is added to a sequence, a deletion occurs when a nucleotide is removed from a sequence, and a substitution occurs when one nucleotide is replaced by another. A nucleotide sequence is represented as a string of letters consisting of the DNA alphabet: {`A, C, G, T`}. Insertions and deletions, called **gaps**, are represented by the symbol **"-"** when showing the alignment. Since it is not possible to know if the difference between the sequences is the result of an insertion in the first sequence or a deletion in the second sequence, a gap is also referred to as an **"indel"** [19].

Consider the small example below. Sequence 1 is DNA from the parent cell, and Sequence 2 is the DNA of the child cell.

```
Sequence 1:  GTGTGT
Sequence 2:  TGTGT
```

These two sequences should be similar since they are directly related (parent and child), but a direct comparison of the nucleotide at the first position of both sequences does not match. There is in fact no match at any nucleotide position, giving the false conclusion that these sequences are not related since there is no similarity. However, if a gap is added at the beginning of the second sequence (representing either adding a G to the second sequence or deleting a G from the first sequence), the sequences can be seen to be similar and correctly reflect the DNA mutation that occurred during replication from parent to child.

```
Sequence 1:  GTGTGT
Sequence 2:  -TGTGT
```

Sequence alignment is the process of aligning sequences to maximize the similarity between them to better reflect the underlying biology [41]. Sequence alignment can be global or local. Global alignments compare entire sequences, where local alignments compare portions of sequences with each other [41, 124]. Sequence alignment methods can also be classified as pairwise or multiple. Pairwise alignment compares two sequences to each other while multiple sequence alignment aligns three or more sequences at the same time [41]. An example of a pairwise global alignment is shown in Figure B.1.

Numerous alignment algorithms and tools have been developed [124]. As of 2012, Fonseca et al. [49] surveyed over 60 pairwise mappers available for DNA, RNA, and protein including Bowtie, BWA, Novoalign, SHRiMP, SOAP, and TopHat. In 2016, Chatzou et al. [22] reviewed top multiple aligners such as MUSCLE, ClustalW, MAFFT, and NAST.

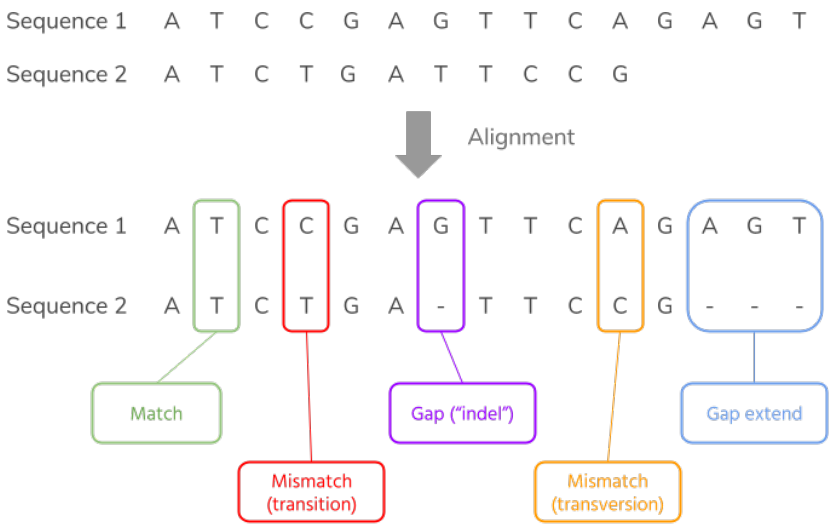


Figure B.1: Sequence Alignment. An overview of pairwise global sequence alignment showing matches, mismatches, and gaps representing insertions/deletions ("indels"). Transversions, transitions, and differences in gaps are additional considerations in finding optimal biological alignments.

Sequence alignment is the core procedure for both clustering OTUs by similarity and matching OTUs to a reference database. Though either method can be used, Sun et al. [143, 144] found that multiple sequence alignment was not as suitable for analyzing the closely related hypervariable regions of the 16S gene. Further, pairwise sequence alignment reportedly produces fewer OTUs than multiple sequence alignment [65]. Many OTU clustering algorithms use pairwise alignment as the basis for calculating the percent identity of two sequences. Percent identity is explained in Appendix C.

The biological issue of aligning two sequences corresponds to the well-known computer science problem of calculating the distance between two strings, called the edit distance [41]. The **Levenshtein edit distance** is the minimum number of operations (insertion, deletion, or substitution) needed to transform one string into another [74]. Edit distance is an example of a cost-benefit scoring function. The simplest scoring function counts the number of operations equally. The edit distance between sequence

1 and sequence 2 in Figure B.1 is 6 since six operations are needed to transform sequence 1 into sequence 2 (4 deletions and 2 substitutions). More complex scoring functions measure the benefit of matches and the cost of mismatches or gaps. The goal of maximizing the similarity can objectively be accomplished by minimizing the edit distance [124].

The original algorithm used to minimize the edit distance and find the optimal pairwise global alignment between two sequences is the **Needleman-Wunsch algorithm** [104]. This algorithm uses a scoring matrix that assigns values to matches, mismatches, and gaps. Once the matrix is completed, a traceback finds the optimal alignment based on the scores in each cell of the matrix. A detailed explanation of the algorithm can be found in numerous online and print resources [19, 124]. A basic example of a completed scoring and traceback matrix is shown in Figure B.2. Optimal alignment is determined by the values chosen for the scoring function [124]. In other words, changing values for the scores can give different alignments.

Biological sequence alignment includes additional considerations to find the most optimal alignment. The scoring function should reflect biological conditions as much as possible. For example, the shape of the nucleotide base determines the relative importance of a substitution. Purine nucleotides (A and G) have a double ring molecular structure and pyrimidine nucleotides (C and T) have a single ring structure. A **transition** is a substitution of a purine for a purine (A $\longleftrightarrow$ G) or a pyrimidine for a pyrimidine (C $\longleftrightarrow$ T). A **transversion** is a substitution of a purine nucleotide for a pyrimidine nucleotide (A,G $\longleftrightarrow$ C,T). Biologically, transitions (similar base shapes) are more common than transversions, so the mismatch penalty can be altered to reflect this biological fact by giving a higher penalty to a transversion [124].

Gap penalties are another area of consideration. During DNA replication, it is more common for a group of nucelotides to be inserted or deleted rather than a single

Seq 1: TGACTTA
Seq 2: GGATAC

Scoring Function:
Match = 2
Mismatch:
Transversion (A,G ↔ C,T) = -2
Transition (A ↔ G or C ↔ T) = -1
Gap = -2

	␣	G	G	A	T	A	C
␣	0	-2	-4	-6	-8	-10	-12
T	-2	-2	-4	-6	-4	-10	-11
G	-4	0	0	-2	-4	-5	-7
A	-6	-2	-1	2	0	-2	-4
C	-8	-4	-3	0	1	-1	0
T	-10	-6	-5	-2	2	0	-2
T	-12	-8	-7	-4	0	0	-1
A	-14	-10	-9	-5	-2	2	0

2 possible optimal alignments:

Seq 1: TGACTTA-
Seq 2: GGA--TAC
Alignment 1
(Biologically more correct)

Seq 1: TGACTTA-
Seq 2: GGA-T-AC
Alignment 2

Figure B.2: Example of a Needleman-Wunsch Alignment Scoring Matrix and Traceback. The completed scoring matrix with the traceback arrows shows two possible optimal alignments. This example uses a constant gap penalty.

nucleotide. The two possible alignments shown in Figure B.2 are both optimal, but Alignment 1 (`GGA--TAC`) is more accurate from a biological viewpoint since it is more common for DNA mutations to delete a region of nucleotides rather than deleting one nucleotide at a time. Current sequence aligners use an **affine gap penalty** to produce alignments that are more likely from a biological perspective by scoring opening gaps more harshly than extending existing gaps [19, 74, 124].

Runtime and computer memory are limiting factors in sequence alignment. Pairwise alignment using Needlmean-Wunsch has a time complexity of $O(mn)$ where m is the length of the first sequence and n is the length of the second sequence. If the two sequences are the same length, the alignment runs in $O(n^2)$ time (quadratic time). For s sequences, the algorithm takes $s * O(mn)$ time. This approach does not scale to millions of reads.

To reduce runtime for millions of reads, OTU clustering algorithms are often a combination of heuristic[1] and optimal alignments. As an obvious example, it would be unnecessary to do a pairwise alignment (and calculate a percent identity) on the following two sequences as they are clearly dissimilar and will not cluster together:

```
Sequence 1:  AAAAAA
Sequence 2:  CGTTGT
```

Rather than perform computationally expensive pairwise alignment on every sequence, OTU picking methods speed up runtime with heuristic algorithms as an initial screen to decide which sequences are close enough to proceed with pairwise alignment and percent identity scoring. The most widely used and well-known heuristic algorithm is BLAST (Basic Local Alignment Search Tool) [6]. Variations of BLAST are used by different OTU clustering algorithms such as USEARCH [35] and VSEARCH [123].

[1] Heuristic algorithms are not guaranteed to find an optimal solution, but often find an acceptable solution in a much faster time.

Appendix C

PERCENT IDENTITY

The **percent identity** is a measure of the similarity (or relatedness) of two aligned sequences. It is used as a cutoff threshold for defining OTUs. Since an OTU is a cluster of related sequences that represent a taxonomic grouping, all sequences in a given OTU cluster must be above a certain percent identity threshold. The level of relation depends on the percent identity. Microbial studies commonly use 97% as the accepted cutoff for a species level identification [105, 131, 70].

This thesis uses the open source software tool VSEARCH to perform OTU clustering. The source code for VSEARCH at `https://github.com/torognes/vsearch` shows that alignment is done using a modified Needleman-Wunsch algorithm with biological scoring features. The percent identity is calculated *after* the sequences are aligned. The VSEARCH manual gives the equation for calculating percent identity [123]:

$$PairwiseIdentity = \frac{Numberof Matches}{(AlignmentLength - TerminalGaps)} \tag{C.1}$$

Using the aligned sequences from Figure B.1 as an example, the percent identity can be calculated as follows:

Sequence 1	A	T	C	C	G	A	G	T	T	C	A	G	A	G	T
Sequence 2	A	T	C	T	G	A	-	T	T	C	C	G	-	-	-

$$PairwiseIdentity = \frac{9}{(15-3)} = 0.75 = 75\% \tag{C.2}$$

Percent identity is the parameter that is used as a cutoff threshold for OTU clustering, both for building OTUs and for matching sequences against reference OTUs in the database. The default percent identity used in LOTUS for open picking OTUs

is 97%. LOTUS currently provide researchers the option to change this parameter only during *de novo* OTU reclustering of the library.

$$PairwiseIdentity = \frac{9}{(AlignmentLength - TerminalGaps)} \tag{C.3}$$

Appendix D

LOTUS DATABASE MYSQL STATEMENTS

The following describes the MySQL CREATE TABLE statements for the LOTUS database. These statements are also found in the `v7.0.5_lotus_base.sql` script in the github repo at `https://github.com/gdewitte06/lotus`.

```
/* Version 7.0.5 - Updated: 2020-Feb-29 */

/* Need utf8 for Django web framework */
CREATE DATABASE lotus CHARACTER SET utf8 COLLATE utf8_bin;
USE lotus;

/* Stores the provenance information for a sample.
Parsed from lotus_metadata_template.csv (or.xlsx) */
CREATE TABLE Sample (
   sampleID  INTEGER(11) PRIMARY KEY AUTO_INCREMENT, /* Internal db
   ID */
   sampleLabel VARCHAR(50), /* Individual label from template: MB.
   Hu2, MB.Ca1 */
   isUnknown TINYINT(1), /* Flag if sample is known/unknown */
   location  VARCHAR(50) DEFAULT NULL, /* Where the sample was
   collected */
   latitude  DECIMAL(10, 8) DEFAULT NULL, /* -90 to +90 degrees */
   longitude DECIMAL(11, 8) DEFAULT NULL, /* -180 to +180 degrees */
   dateCollected DATE, /* Date format: YYYY-MM-DD */
   contributor VARCHAR(50) DEFAULT NULL, /* CA Dept of Fish & Game,
   Pacific Wildlife Care */
   investigator VARCHAR(50) DEFAULT NULL, /* Technician name */
   dateUploaded DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
   CURRENT_TIMESTAMP, /* Date sample was uploaded to db */
   CONSTRAINT UKSample_sampleLabel UNIQUE KEY(sampleLabel)
);

/*--------------------------------------------------------*/

CREATE TABLE HostSpecies (
   latinName  VARCHAR(50) DEFAULT NULL, /* Felis catus or ' ' */
   commonName VARCHAR(50) NOT NULL,     /* Cat or Dog */
   PRIMARY KEY (commonName)
);

/*--------------------------------------------------------*/

/* Stores information for a known individual animal host.
Parsed from lotus_metadata_template.csv (or.xlsx) */
CREATE TABLE Host (
   hostID     INTEGER(11) PRIMARY KEY AUTO_INCREMENT, /* Internal db
    ID */
   commonName VARCHAR(50),   /* Cat or Dog */
   hostLabel  VARCHAR(50),   /* Label from template for this
   specific host individual: Cat1, Horse23 */
   CONSTRAINT UKcommonNameHostLabel UNIQUE KEY(commonName, hostLabel
   ),
   CONSTRAINT FKHost_commonName FOREIGN KEY (commonName) REFERENCES
   HostSpecies (commonName)
```

```
);

/*-------------------------------------------------------*/

/* Stores information for an unknown environmental site.
Parsed from lotus_metadata_template.csv (or.xlsx) */
CREATE TABLE Site (
   siteID      INTEGER(11) PRIMARY KEY AUTO_INCREMENT, /* Internal db
    ID */
   siteName    VARCHAR(50),    /* General area from template: Pacific
   Ocean Water, SLO Creek Water, Freshwater Lagoon */
   siteLabel   VARCHAR(50),    /* Specific area from template: Site 1,
    Mission Tunnel Mouth */
   CONSTRAINT UKsiteLabelName UNIQUE KEY(siteLabel, siteName)
);

/*-------------------------------------------------------*/

CREATE TABLE SampleToSite (
   sampleID    INTEGER(11) NOT NULL,
   siteID      INTEGER(11) NOT NULL,
   PRIMARY KEY (sampleID, siteID),
   CONSTRAINT FKSampleSite_sampleID FOREIGN KEY (sampleID)
   REFERENCES Sample (sampleID),
   CONSTRAINT FKSampleSite_siteID FOREIGN KEY (siteID) REFERENCES
   Site (siteID)
);

/*-------------------------------------------------------*/

CREATE TABLE SampleToHost (
   sampleID    INTEGER(11) NOT NULL,
   hostID      INTEGER(11) NOT NULL,
   PRIMARY KEY (sampleID, hostID),
   CONSTRAINT FKSample_sampleID FOREIGN KEY (sampleID) REFERENCES
   Sample (sampleID),
   CONSTRAINT FKSample_hostID FOREIGN KEY (hostID) REFERENCES Host (
   hostID)
);

/*-------------------------------------------------------*/

/* Sequence table will contain only unique (dereplicated) sequences
   created in the open-picking pipeline - excludes singletons and
   chimeras.
Parsed from matches.fna and non-matches.fna files. */
CREATE TABLE Sequence (
   seqID       INTEGER(11) PRIMARY KEY AUTO_INCREMENT,  /* Internal
   db ID */
   qiimeLabel VARCHAR(60),    /* The "representative" label applied
   by qiime split_library script after dereplication. ex: MB.Do1_9
   */
   seqDNA      VARCHAR(600),  /* The actual nucleotide sequence */
   CONSTRAINT UKSequence_seqDNA UNIQUE KEY(seqDNA),
   CONSTRAINT UKSequence_qiimeLabel UNIQUE KEY(qiimeLabel)
);
/* Note: For seqDNA column with VARCHAR(600) and the UNIQUE key,
MySQL has a 767 byte limit for unique indexes. MySQL assumes 3 bytes
    per utf8 character, so the maximum varchar is 255 (256 x 3 =
   768). It will only index the first 255 characters. */

/*-------------------------------------------------------*/

/* Same as Sequence table, but for Denovo Sequences (sequences
   created in the denovo picking pipeline).
Parsed from nonchimeras.fna file. */
```

```
CREATE TABLE Denovo_Sequence (
   seqID       INTEGER(11) PRIMARY KEY AUTO_INCREMENT,
   qiimeLabel  VARCHAR(60),
   seqDNA      VARCHAR(600),
   CONSTRAINT UKDenovo_Sequence_seqDNA UNIQUE KEY(seqDNA),
   CONSTRAINT UKDenovo_Sequence_qiimeLabel UNIQUE KEY(qiimeLabel)
);

/*-------------------------------------------------------------*/

/* Same as Sequence table, but for Unknown Sequences (sequences
   created in the unknown matching pipeline).
Parsed from nonchimeras.fna file. */
CREATE TABLE Unknown_Sequence (
   seqID       INTEGER(11) PRIMARY KEY AUTO_INCREMENT,
   qiimeLabel  VARCHAR(60),
   seqDNA      VARCHAR(600),
   CONSTRAINT UKUnknown_Sequence_seqDNA UNIQUE KEY(seqDNA),
   CONSTRAINT UKUnknown_Sequence_qiimeLabel UNIQUE KEY(qiimeLabel)
);

/*-------------------------------------------------------------*/

/* SeqSampleMapping contains the mapping of all sequence labels for
   a sequence ID (the duplicate labels) from the open picking
   pipeline - excludes chimeras and singletons.
Parsed from the derep_out.uc file. */
CREATE TABLE SeqSampleMapping (
   seqID       INTEGER(11) NOT NULL,
   qiimeLabel  VARCHAR(100),  /* The sequence label assigned by
   split_libraries script - NOT a foreign key Ex: MB.Do1_1 */
   sampleLabel VARCHAR(50),  /* The sampleLabel the sequence label
   came from Ex: MB.Do1 */
   PRIMARY KEY (seqID, qiimeLabel),
   CONSTRAINT FKSeqSampleMapping_seqID FOREIGN KEY (seqID)
   REFERENCES Sequence (seqID),
   CONSTRAINT FKSeqSampleMapping_sampleLabel FOREIGN KEY (
   sampleLabel) REFERENCES Sample (sampleLabel)
);

/*-------------------------------------------------------------*/

/* Same as SeqSampleMapping table, but for Denovo Sequences. */
CREATE TABLE Denovo_SeqSampleMapping (
   seqID        INTEGER(11) NOT NULL,
   qiimeLabel   VARCHAR(100),
   sampleLabel  VARCHAR(50),
   PRIMARY KEY (seqID, qiimeLabel),
   CONSTRAINT FKDenovo_SeqSampleMapping_seqID FOREIGN KEY (seqID)
   REFERENCES Denovo_Sequence (seqID),
   CONSTRAINT FKDenovo_SeqSampleMapping_sampleLabel FOREIGN KEY (
   sampleLabel) REFERENCES Sample (sampleLabel)
);

/*-------------------------------------------------------------*/

/* Same as SeqSampleMapping table, but for Unknown Sequences. */
CREATE TABLE Unknown_SeqSampleMapping (
   seqID        INTEGER(11) NOT NULL,
   qiimeLabel   VARCHAR(100),
   sampleLabel  VARCHAR(50),
   PRIMARY KEY (seqID, qiimeLabel),
   CONSTRAINT FKUnknown_SeqSampleMapping_seqID FOREIGN KEY (seqID)
   REFERENCES Unknown_Sequence (seqID),
   CONSTRAINT FKUnknown_SeqSampleMapping_sampleLabel FOREIGN KEY (
   sampleLabel) REFERENCES Sample (sampleLabel)
```

```
);

/*-------------------------------------------------------*/

/* Holds the OTUs generated by Cal Poly OTU Open Picking Protocol.
Parsed from otus.fna file. */
CREATE TABLE OTU (
   otuID      INTEGER(11) PRIMARY KEY AUTO_INCREMENT,  /* Internal
   DB id */
   otuLabel   VARCHAR(50),   /* <proj name>_OTU_2 label from
   clustering OTU step*/
   qiimeLabel VARCHAR(60),   /* The centroid label, a "
   reprensentative" label applied by split_library script after
   clustering = qiimeLabel in Sequence table */
   numSeqs    INTEGER(11) DEFAULT 0,  /* Holds the number of
   sequences that make up this otu */
   CONSTRAINT FKOTU_qiimeLabel FOREIGN KEY (qiimeLabel) REFERENCES
   Sequence (qiimeLabel),
   CONSTRAINT UKOTU_qiimeLabel UNIQUE KEY(qiimeLabel),
   CONSTRAINT UKOTU_otuLabel UNIQUE KEY (otuLabel)
);

/*-------------------------------------------------------*/

/* Same as OTU table, but for Denovo OTUs (OTUs generated by Cal
   Poly OTU Denovo Picking Protocol)  */
CREATE TABLE Denovo_OTU (
   otuID      INTEGER(11) PRIMARY KEY AUTO_INCREMENT,
   otuLabel   VARCHAR(50),
   qiimeLabel VARCHAR(60),
   numSeqs    INTEGER(11) DEFAULT 0,
   CONSTRAINT FKDenovo_OTU_qiimeLabel FOREIGN KEY (qiimeLabel)
   REFERENCES Denovo_Sequence (qiimeLabel),
   CONSTRAINT UKDenovo_OTU_qiimeLabel UNIQUE KEY(qiimeLabel)
);

/*-------------------------------------------------------*/

/* Must link to SeqSampleMapping to know the sequences (and
   associated samples) that clustered into an OTU.
Parsed from otu_table_seq_mapping.uc file.*/
CREATE TABLE OTUSeqMapping (
   otuID      INTEGER(11) NOT NULL,
   seqID      INTEGER(11) NOT NULL,
   PRIMARY KEY (otuID, seqID),
   CONSTRAINT FKOTUSeqMapping_otuID FOREIGN KEY (otuID) REFERENCES
   OTU (otuID),
   CONSTRAINT FKOTUSeqMapping_seqID FOREIGN KEY (seqID) REFERENCES
   Sequence (seqID)
);

/*-------------------------------------------------------*/

/* Same as OTUSeqMapping but for Denovo Sequences and OTUs. */
CREATE TABLE Denovo_OTUSeqMapping (
   otuID      INTEGER(11) NOT NULL,
   seqID      INTEGER(11) NOT NULL,
   PRIMARY KEY (otuID, seqID),
   CONSTRAINT FKDenovo_OTUSeqMapping_otuID FOREIGN KEY (otuID)
   REFERENCES Denovo_OTU (otuID),
   CONSTRAINT FKDenovo_OTUSeqMapping_seqID FOREIGN KEY (seqID)
   REFERENCES Denovo_Sequence (seqID)
);

/*-------------------------------------------------------*/
```

```
/* Stores the nucleotide information for each position in the final
   OTU sequence.
Parsed from otu_profile.txt file.*/
CREATE TABLE OTUProfile (
   otuID        INTEGER(11) NOT NULL,
   position     INTEGER(11) NOT NULL, /* 0-based index position in
   sequence string */
   consensusNT CHAR(1),  /* Consensus nucleotide at this position */
   numA         INTEGER(11), /* Number of A's at this position */
   numC         INTEGER(11), /* Number of C's at this position */
   numG         INTEGER(11), /* Number of G's at this position */
   numT         INTEGER(11), /* Number of T's at this position */
   numGaps      INTEGER(11), /* Number of gaps at this position */
   numAmbig     INTEGER(11), /* Number of N's (ambiguous) at this
   position */
   PRIMARY KEY (otuID, position),
   CONSTRAINT FKOTUProfile_otuID FOREIGN KEY (otuID) REFERENCES OTU
   (otuID)
);

/*-------------------------------------------------------------*/

/* Same as OTUProfile, but for Denovo OTUs. */
CREATE TABLE Denovo_OTUProfile (
   otuID        INTEGER(11) NOT NULL,
   position     INTEGER(11) NOT NULL,
   consensusNT CHAR(1),
   numA         INTEGER(11),
   numC         INTEGER(11),
   numG         INTEGER(11),
   numT         INTEGER(11),
   numGaps      INTEGER(11),
   numAmbig     INTEGER(11),
   PRIMARY KEY (otuID, position),
   CONSTRAINT FKDenovo_OTUProfile_otuID FOREIGN KEY (otuID)
   REFERENCES Denovo_OTU (otuID)
);

/*-------------------------------------------------------------*/

 /* Matches unknown samples to OTUs created by Open picking
 Parsed from otu_table_sample_mapping.uc file.*/
CREATE TABLE Unk_OTUSampleMapping (
   otuID       INTEGER(11) NOT NULL,
   seqID       INTEGER(11) NOT NULL,
   otuSpecies VARCHAR(50), /* The species assigned taxonomy for
   given otuID from view v_otutospecies */
   PRIMARY KEY (otuID, seqID),
   CONSTRAINT FKUnk_OTUSampleMapping_seqID FOREIGN KEY (seqID)
   REFERENCES Unknown_Sequence (seqID),
   CONSTRAINT FKUnk_OTUSampleMapping_otuID FOREIGN KEY (otuID)
   REFERENCES OTU (otuID)
);

/*-------------------------------------------------------------*/

/* Same as Unk_OTUSampleMapping but matches to denovo OTUs */
CREATE TABLE Unk_Denovo_OTUSampleMapping (
   otuID       INTEGER(11) NOT NULL,
   seqID       INTEGER(11) NOT NULL,
   otuSpecies VARCHAR(50),
   PRIMARY KEY (otuID, seqID),
   CONSTRAINT FKUnkDenovo_OTUSampleMapping_seqID FOREIGN KEY (seqID)
    REFERENCES Unknown_Sequence (seqID),
   CONSTRAINT FKUnkDenovo_OTUSampleMapping_otuID FOREIGN KEY (otuID)
    REFERENCES Denovo_OTU (otuID)
```

```
);

/*--------------------------------------------------------*/

/* Records when database was updated via open or denovo picking */
CREATE TABLE HistoryOTUPickMethod (
   method       VARCHAR(10) NOT NULL,
   projectAdded VARCHAR(128),
   dateUpdated  DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
   CURRENT_TIMESTAMP,
   PRIMARY KEY (method, dateUpdated)
);

/*--------------------------------------------------------*/

INSERT INTO HostSpecies VALUES
    ('Homo sapiens', 'Human'),
    ('Aeorestes sp.', 'Bat'),
    ('Lynx sp.', 'Bobcat'),
    ('Felis silvestris', 'Cat'),
    ('Gallus sp.', 'Chicken'),
    ('Bos taurus', 'Cow'),
    ('Corvus sp.', 'Crow'),
    ('Odocoileus sp.', 'Deer'),
    ('Canis lupus', 'Dog'),
    ('Cervus elaphus', 'Elk'),
    ('Buteo sp.', 'Hawk'),
    ('Equus caballus', 'Horse'),
    ('Capra hircus', 'Goat'),
    ('Lama sp.', 'Llama'),
    ('Puma sp.', 'Mountain Lion'),
    ('Didelphis virginiana', 'Opossum'),
    ('Struthio sp.', 'Ostrich'),
    ('Bubo sp.', 'Owl'),
    ('Pelecanus sp.', 'Pelican'),
    ('Sus scrofa', 'Pig'),
    ('Columba sp.', 'Pigeon'),
    ('Procyon lotor', 'Raccoon'),
    ('Larus sp.', 'Seagull'),
    ('Ovis aries', 'Sheep'),
    ('Mephitis mephitis', 'Skunk'),
    ('Passer sp.', 'Sparrow'),
    ('Sciurus sp.', 'Squirrel'),
    ('Oryctolagus cuniculus', 'Rabbit'),
    ('Meleagris sp.', 'Turkey');
```

Appendix E

LOTUS DATABASE SUPPLEMENTAL TABLES

In addition to the base tables created for LOTUS based on the ER diagram in Figure 3.1, the Django web application requires some built-in core tables for website functionality. Figure E.1 shows the core tables and attributes created by the Django framework using the `python manage.py migrate` command as outlined in the Django documentation[1].

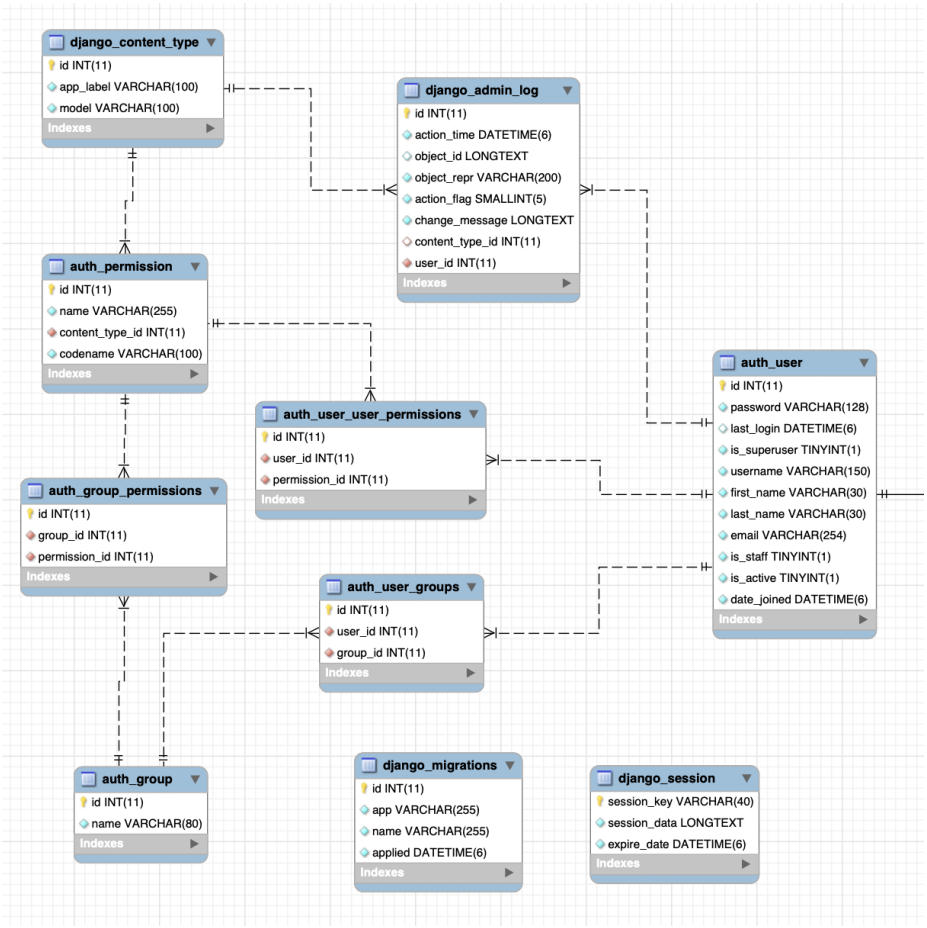


Figure E.1: LOTUS Django-created supplemental Tables

One additional table is required after the creation of the Django core tables. A

[1] `https://docs.djangoproject.com/en/1.11/howto/legacy-databases/`

UserToSample table is needed to record the Samples that each User submitted. The CREATE TABLE statement for the UserToSample table is shown below.

```
01 |   /* This table CANNOT be created without Django tables being
       created first! */
02 |  /* UserToSample table to match Django users to the samples
       they uploaded */
03 |  /* auth_user is a Django-generated table */
04 |  DROP TABLE IF EXISTS UserToSample;
05 |  CREATE TABLE UserToSample (
06 |     userID     INTEGER(11) NOT NULL,
07 |     sampleID   INTEGER(11) NOT NULL,
08 |     PRIMARY KEY (userID, sampleID),
09 |     CONSTRAINT FKUserToSample_userID FOREIGN KEY (userID)
       REFERENCES auth_user (id),
10 |     CONSTRAINT FKUserToSample_sampleID FOREIGN KEY (sampleID)
       REFERENCES Sample (sampleID)
11 |  );
```

Appendix F

ADDITIONAL TIMING COMPARISON GRAPHS

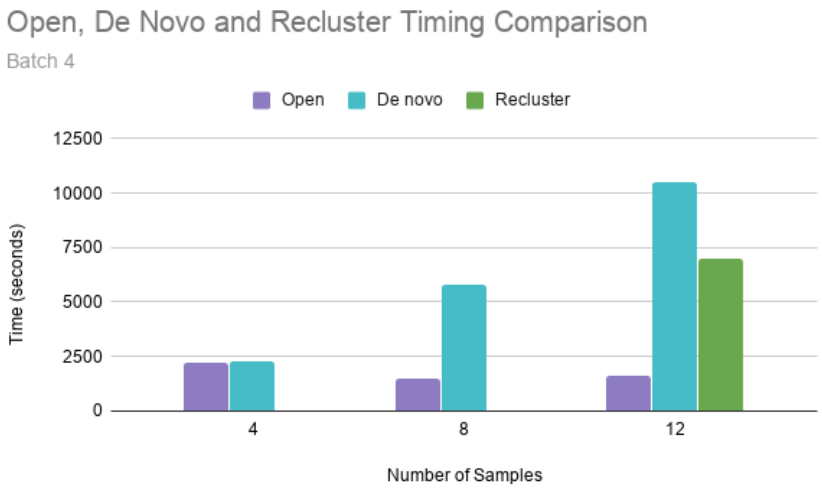


Figure F.1: Bar graph showing total processing times for different batch 4 strategies based on number of samples. The recluster strategy op4_12_rc is only done for 12 samples. The comparison is made by number of samples. For example, dn_8 is the denovo strategy for 8 samples. It is compared to op4_8, the open strategy for 8 samples.

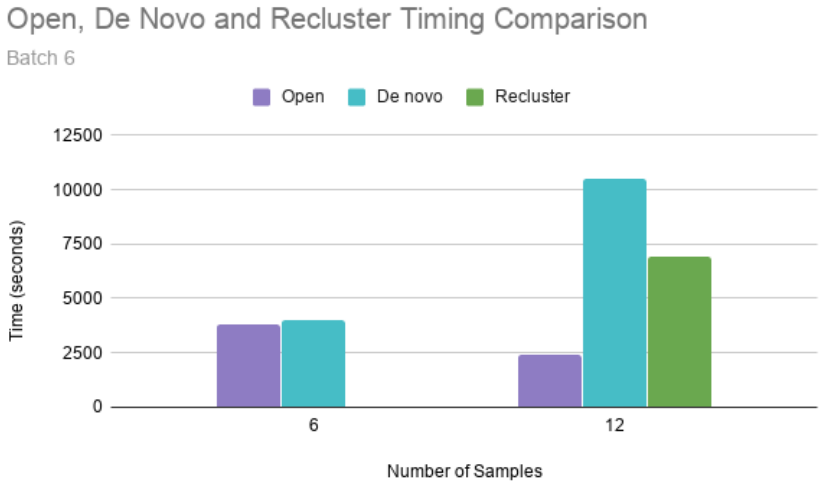


Figure F.2: Bar graph showing total processing times for different batch 6 strategies based on number of samples.

Appendix G

SINGLE-SOURCE VS. MULTI-SOURCE GRAPHS

G.1 Batch 3 Graphs

G.2 Batch 4 Graphs

G.3 Batch 6 Graphs

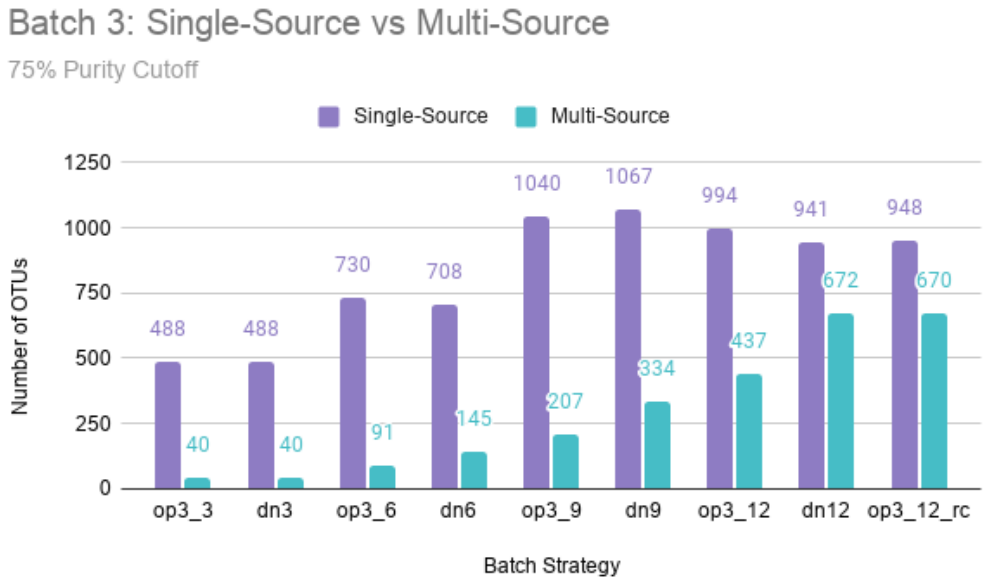


Figure G.1: Graph comparing single-source vs multi-source OTUs for different batch 3 strategies using 75% purity cutoff to define single-source OTUs.

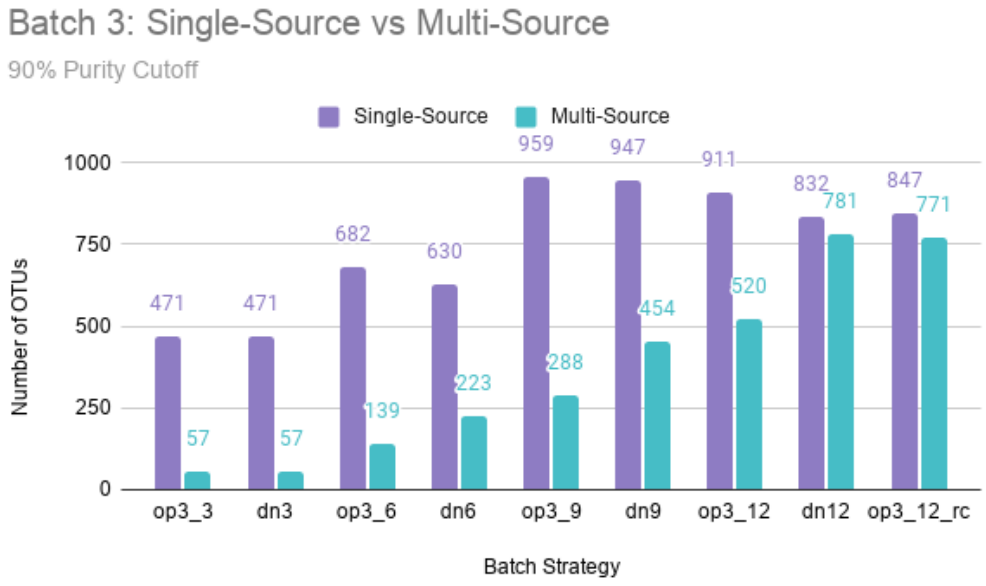


Figure G.2: Graph comparing single-source vs multi-source OTUs for different batch 3 strategies using 90% purity cutoff to define single-source OTUs.

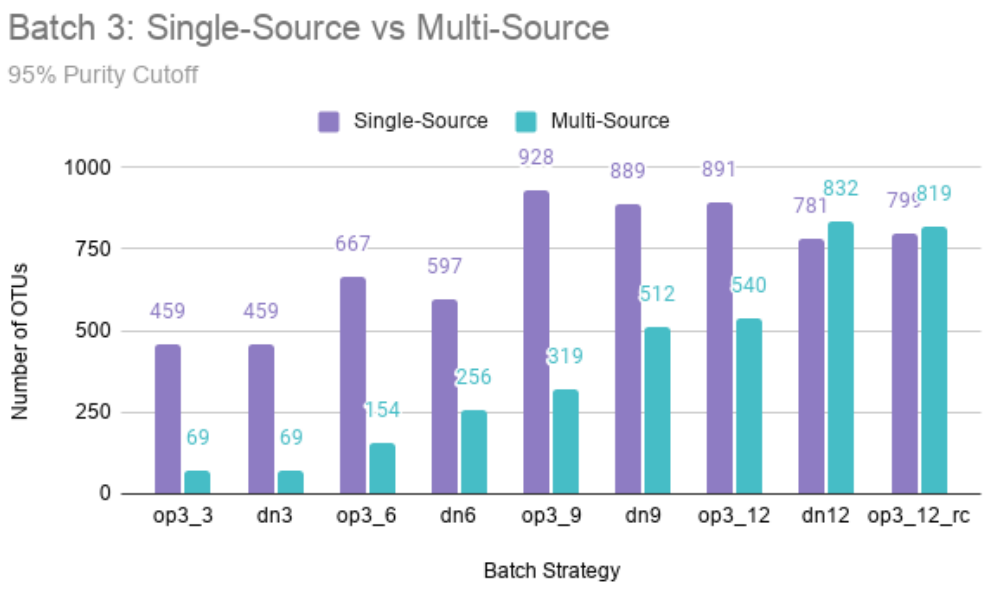


Figure G.3: Graph comparing single-source vs multi-source OTUs for different batch 3 strategies using 95% purity cutoff to define single-source OTUs.

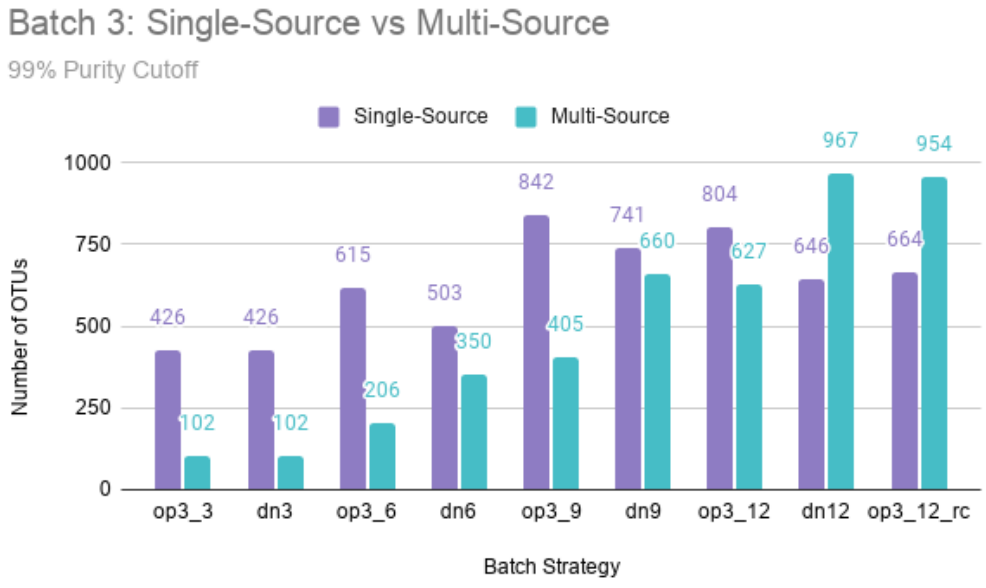


Figure G.4: Graph comparing single-source vs multi-source OTUs for different batch 3 strategies using 99% purity cutoff to define single-source OTUs.

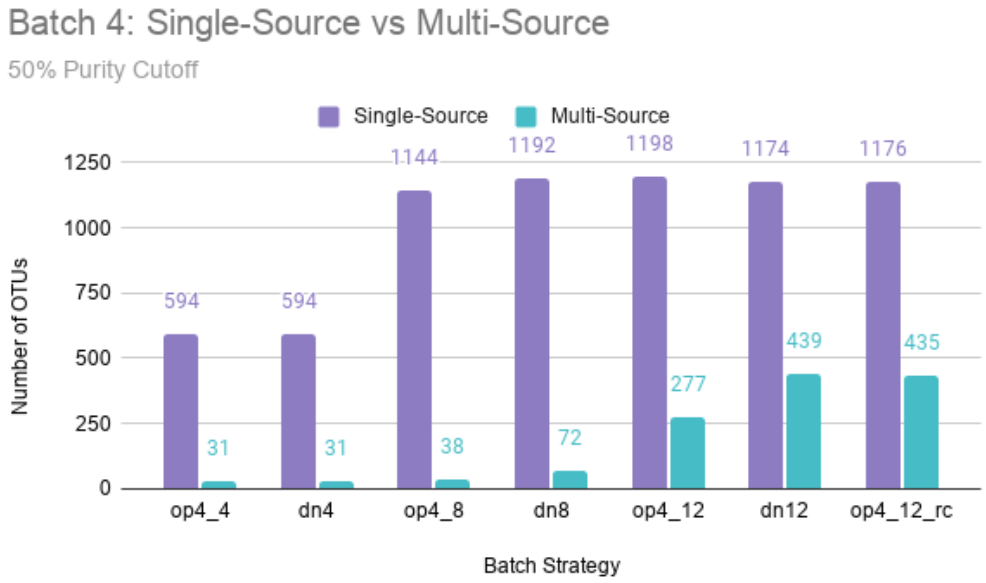


Figure G.5: Graph comparing single-source vs multi-source OTUs for different batch 4 strategies using 50% purity cutoff to define single-source OTUs.

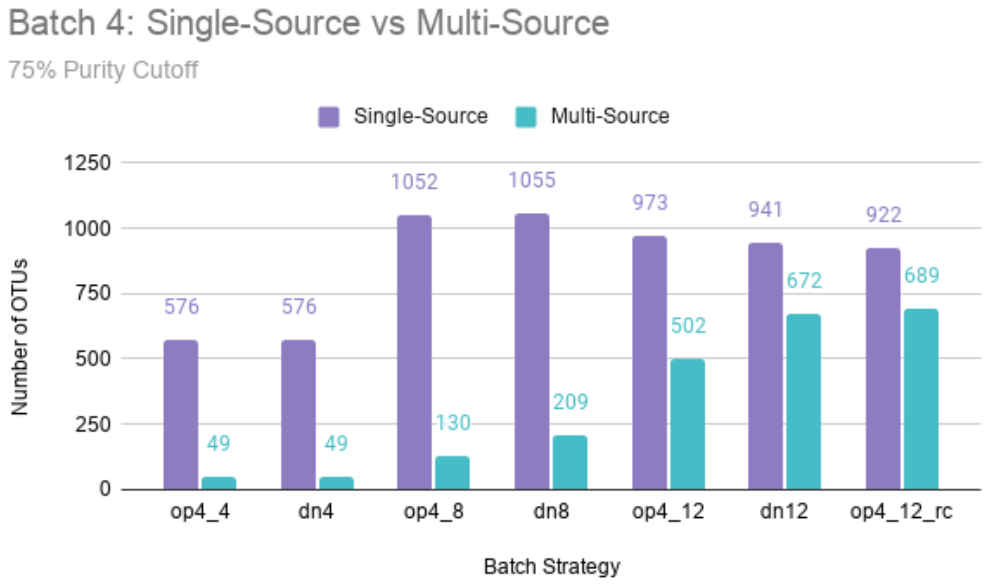


Figure G.6: Graph comparing single-source vs multi-source OTUs for different batch 4 strategies using 75% purity cutoff to define single-source OTUs.

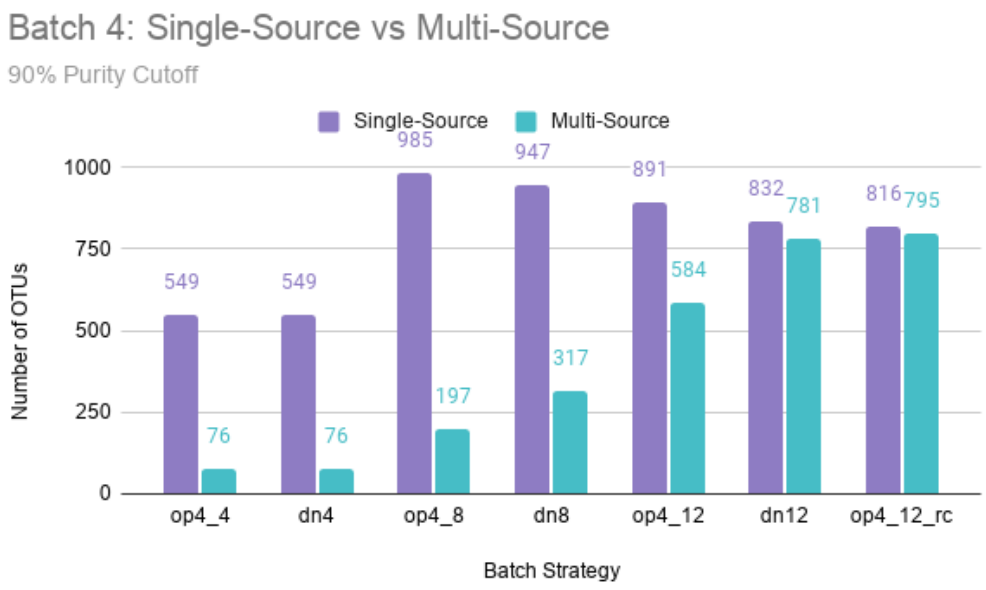


Figure G.7: Graph comparing single-source vs multi-source OTUs for different batch 4 strategies using 90% purity cutoff to define single-source OTUs.

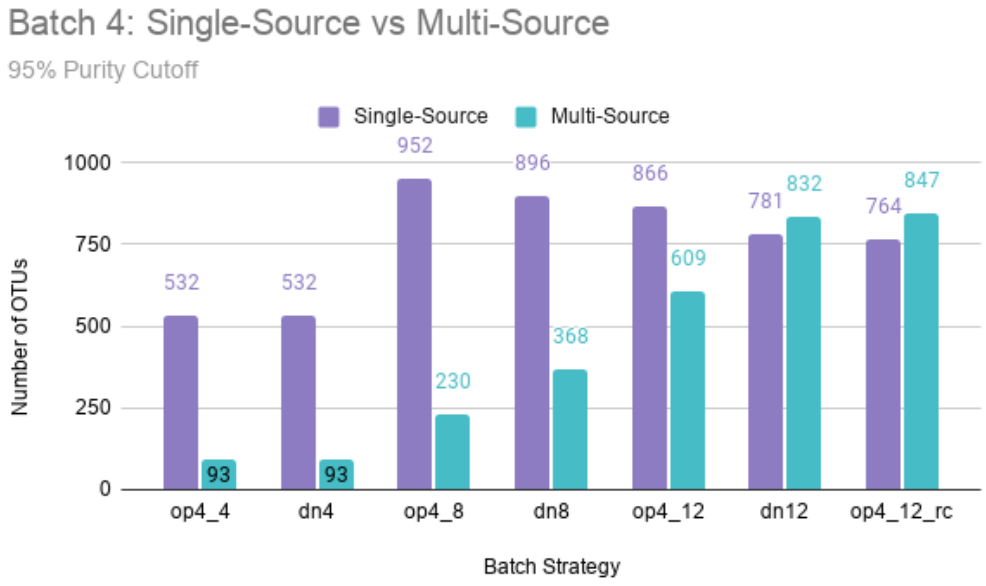


Figure G.8: Graph comparing single-source vs multi-source OTUs for different batch 4 strategies using 95% purity cutoff to define single-source OTUs.

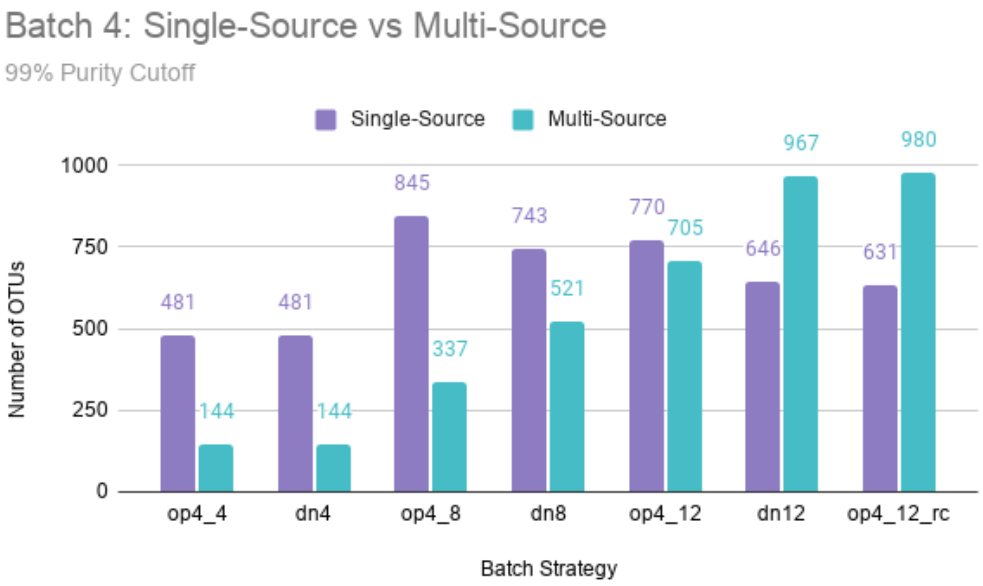


Figure G.9: Graph comparing single-source vs multi-source OTUs for different batch 4 strategies using 99% purity cutoff to define single-source OTUs.

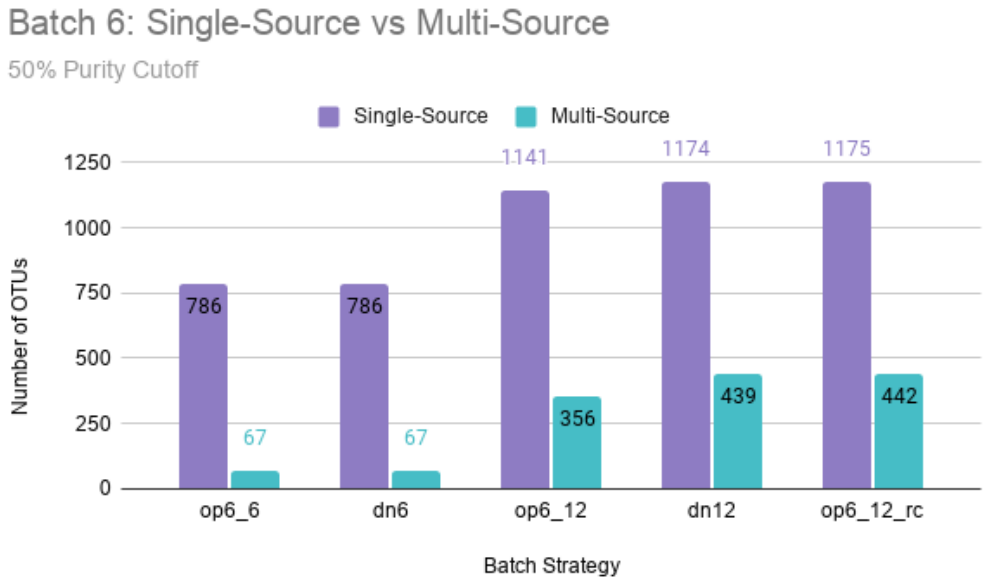


Figure G.10: Graph comparing single-source vs multi-source OTUs for different batch 6 strategies using 50% purity cutoff to define single-source OTUs.

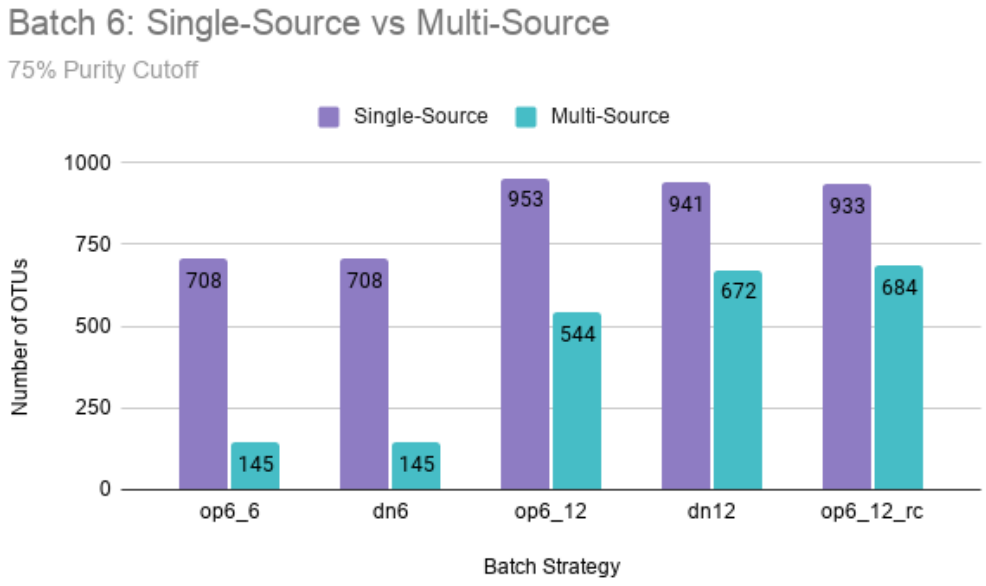


Figure G.11: Graph comparing single-source vs multi-source OTUs for different batch 6 strategies using 75% purity cutoff to define single-source OTUs.

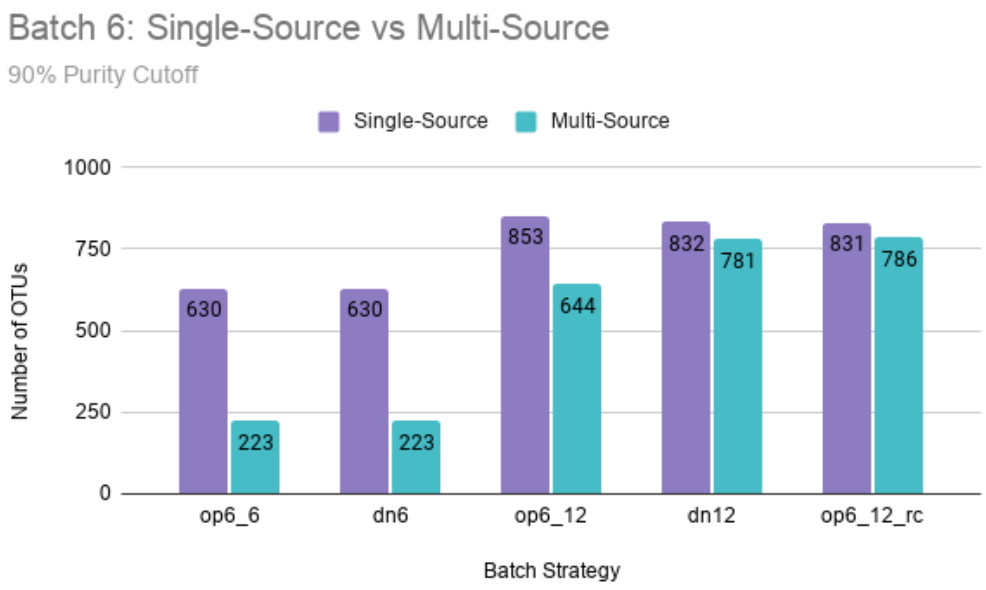


Figure G.12: Graph comparing single-source vs multi-source OTUs for different batch 6 strategies using 90% purity cutoff to define single-source OTUs.

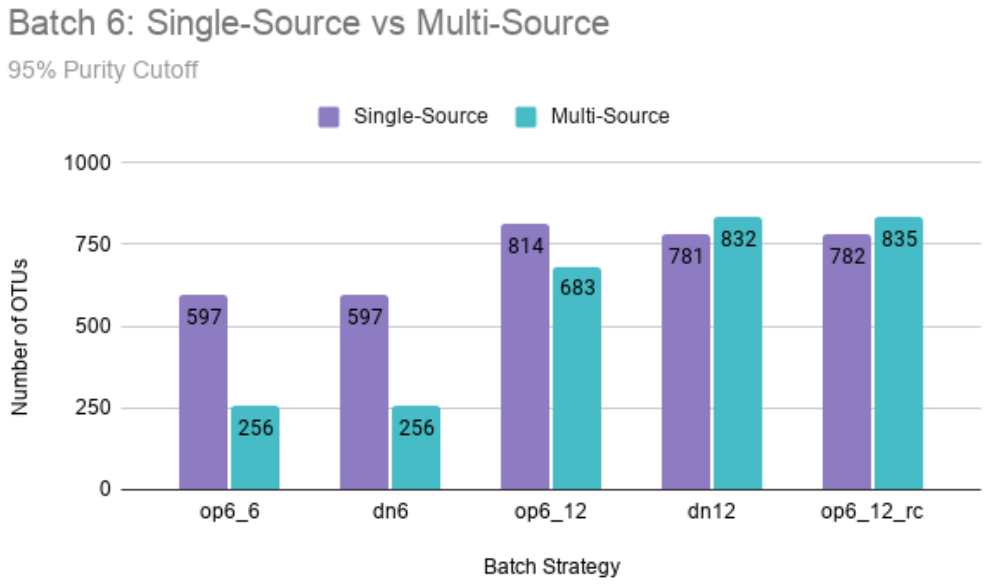


Figure G.13: Graph comparing single-source vs multi-source OTUs for different batch 6 strategies using 95% purity cutoff to define single-source OTUs.

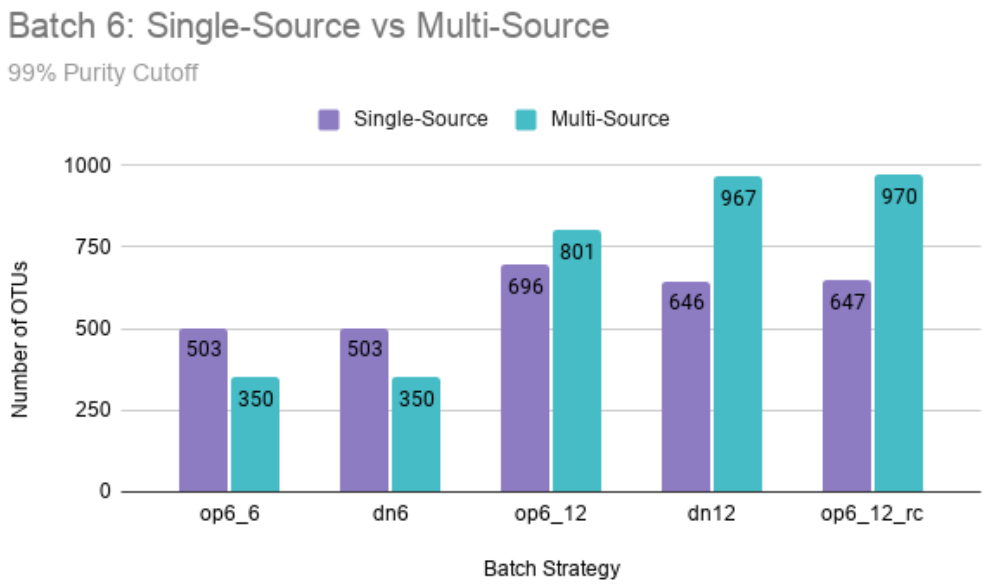


Figure G.14: Graph comparing single-source vs multi-source OTUs for different batch 6 strategies using 99% purity cutoff to define single-source OTUs.

Appendix H

OPEN VS. *DE NOVO* STRATEGY COMPARISON BY SAMPLE SIZE GRAPHS

The following graphs represent visualizations for comparison of open and *de novo* OTU picking strategies by sample size as a complement to Table 5.11.

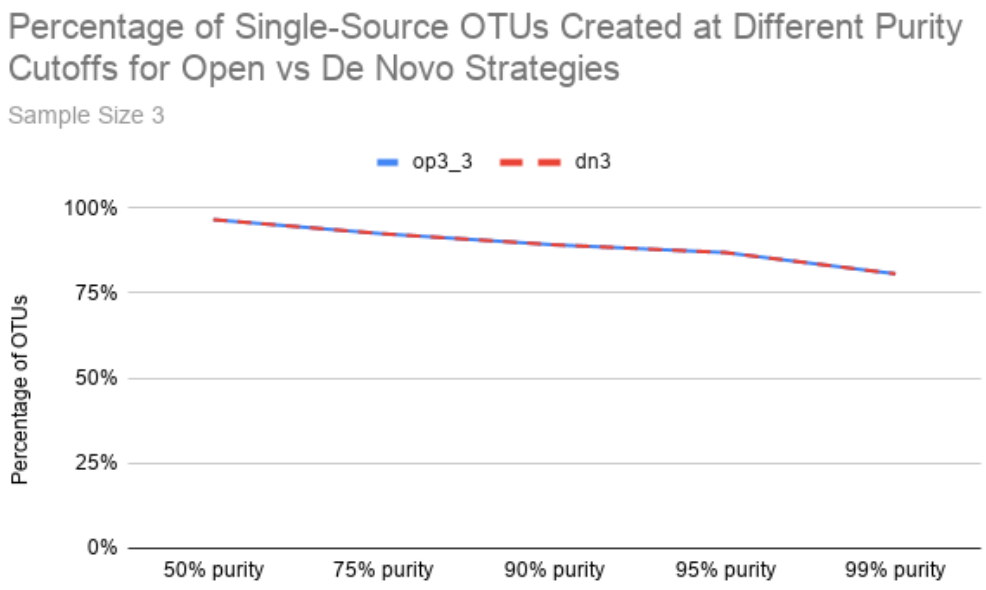


Figure H.1: Graph comparing open vs *de novo* strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 3. As op3_3 is the initial run, it uses the *de novo* picking algorithm, hence both open and *de novo* strategies are the same and produce the same percentage of OTUs at every purity cutoff for this sample size. The *de novo* strategy dn_3 is represented by a dashed red line and can be seen to overlap the op3_3 blue line.

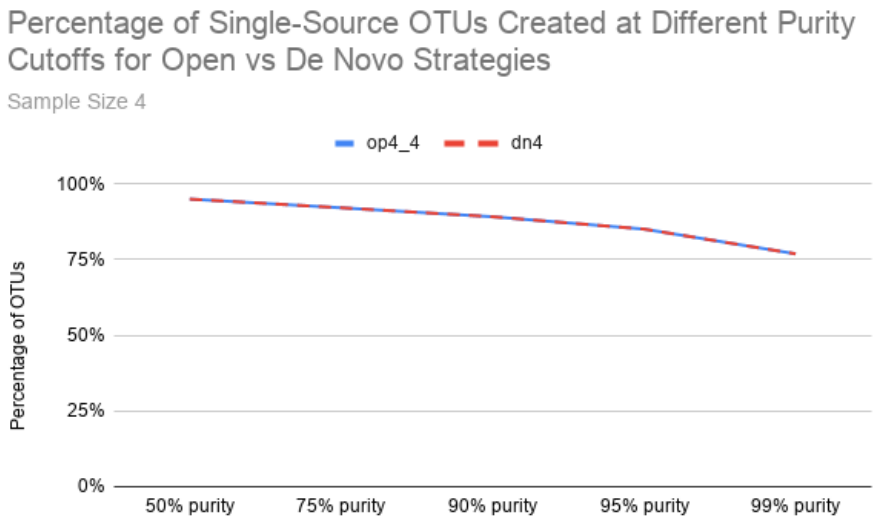


Figure H.2: Graph comparing open vs *de novo* strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 4. As op4_4 is the initial run, it uses the *de novo* picking algorithm, hence both open and *de novo* strategies are the same and produce the same percentage of OTUs at every purity cutoff for this sample size. The *de novo* strategy dn_4 is represented by a dashed red line and can be seen to overlap the op4_4 blue line.

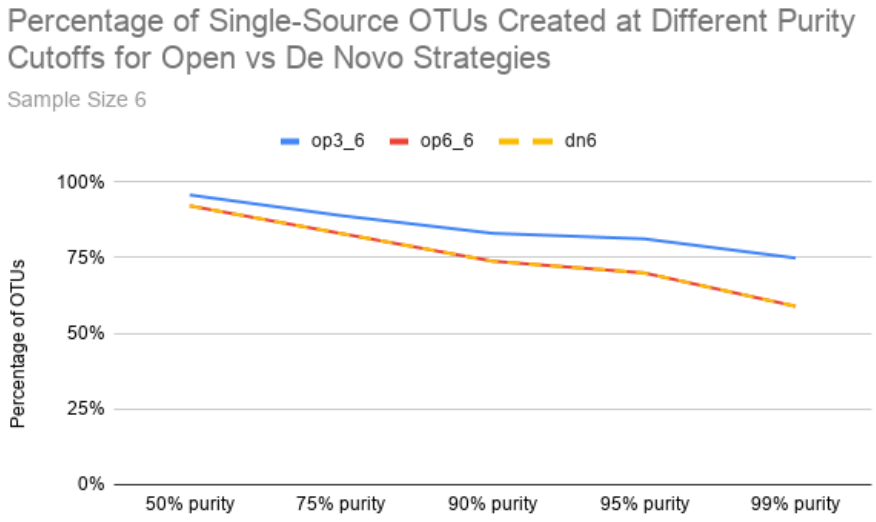


Figure H.3: Graph comparing open vs *de novo* strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 6. As op6_6 is the initial run, it uses the *de novo* picking algorithm, hence both op6_6 and dn_6 strategies are the same and produce the same percentage of OTUs at every purity cutoff. The op3_6 strategy actually represents open picking and produces higher percentages of single-source OTUs than the *de novo* strategy.

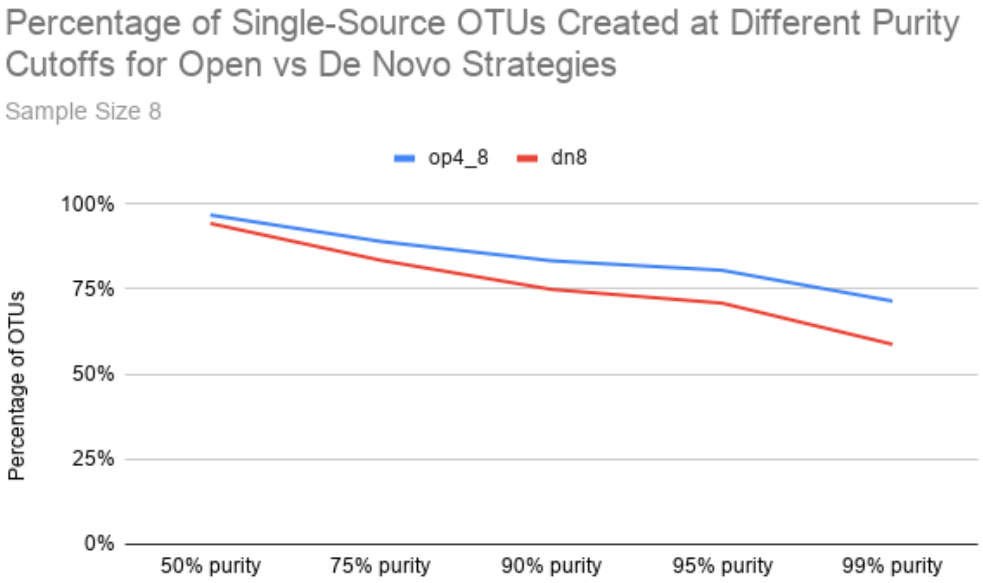


Figure H.4: Graph comparing open vs *de novo* strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 8. The op4_8 strategy produces a higher percentage of single-source OTUs than its *de novo* counterpart at every purity cutoff.

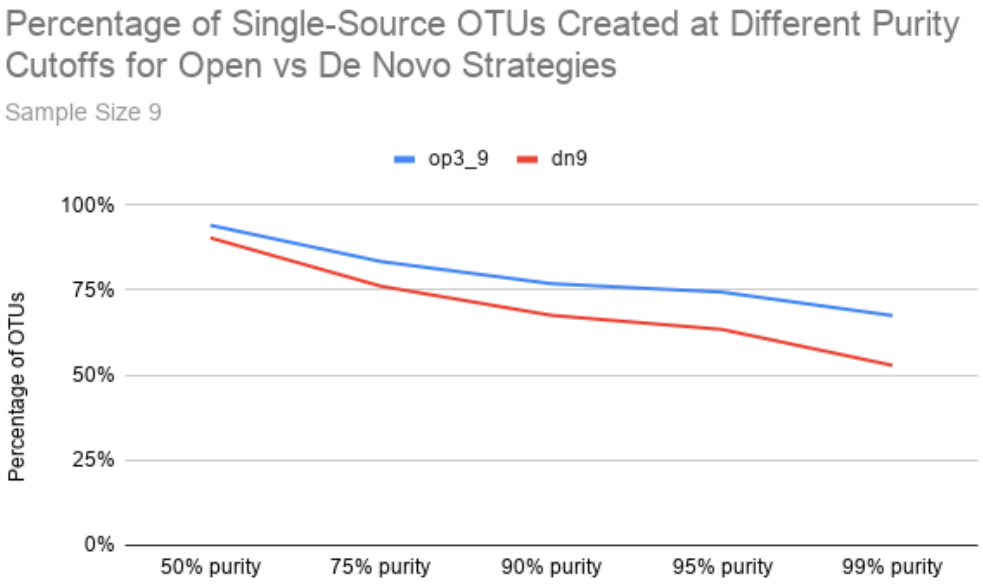


Figure H.5: Graph comparing open vs *de novo* strategies by percentage of single-source OTUs produced at different purity cutoffs for sample size 9. The op3_9 strategy produces a higher percentage of single-source OTUs than its *de novo* counterpart at every purity cutoff.

Appendix I

SINGLE-SOURCE OTU ANALYSIS WITH OTU PURITY GRAPHS

LOTUS provides summary information in graphical format for insights about the reference library OTUs. These graphs, previously shown in Figure 4.19, are found on the web application OTU Purity Graph page and show the OTU purity breakdown for an individual host species. **Single-source OTUs are defined by the purity cutoff.** For example, if the purity threshold is 90%, then single-source cat OTUs are those in which 90% or more of the sequences come from a cat host. This appendix includes 12 graphs (Figures I.1 – I.12) from all 12 host species used to create the reference library of OTUs for evaluation testing.

Each graph shows an overview of OTU purity for all the OTUs associated with a given host species. Information about the size of the OTU clusters (how many sequences constitute a cluster) is not included. For a 75% purity threshold, if a horizontal line is drawn across the graph at 75% purity, only the OTUs above that line would be considered single-source (or host-specific). Looking at the Cow host graph in Figure I.2, there are 235 total OTUs that contain sequences from the host species cow (cow-associated OTUs). Of these, only 64 OTUs are classified as single-source Cow given a 75% purity threshold. Only 52 OTUs are single-source Cow at a 90% purity cutoff, and even fewer at 100% purity. An assumption can therefore be made that cow sequences are found in many OTUs that are not cow-specific.

The information from all the graphs is summarized in Table I.1. Using a 75% purity threshold, turkeys, horses, pigeons, and pigs are species with the highest percentages of host-specific OTUs while seagull, dog, and sheep have the lowest. This suggests there may be a difference in OTU effectiveness in different host species.

Table I.1: Single-source OTU Analysis ordered by species with the most host-specific OTUs by percentage. The data from Figures I.1 – I.12 are shown here in tabular form for comparison between species. The Total OTUs column represents the number of host-associated OTUs that contain any sequences from a given host. OTUs are also shown by purity threshold. For example, there were 317 cat-associated OTUs, or OTUs that contained sequences from the cat host species. Of these 317, there were 43 OTUs that consisted entirely of cat sequences (100% purity), meaning only 13.6% of cat-associated OTUs were cat-specific. If cat-specific OTUs are defined at 75% purity, then 64 OTUs (20.2%) of the 317 cat-associated OTUs are cat-specific.

Host Species		**100% purity**		**90% purity**		**75% purity**	
	Total OTUs	**Num OTUs**	**%**	**Num OTUs**	**%**	**Num OTUs**	**%**
Turkey	281	166	59.1	191	68.0	196	69.8
Horse	229	126	55.0	155	67.7	158	69.0
Pigeon	247	119	48.2	131	53.0	132	53.4
Pig	197	70	35.5	87	44.2	90	45.7
Human	271	72	26.6	91	33.6	103	38.0
Cow	235	38	16.2	52	22.1	64	27.2
Cat	317	43	13.6	47	14.8	64	20.2
Deer	337	43	12.8	46	13.6	51	15.1
Goat	420	40	9.5	41	9.8	55	13.1
Seagull	349	30	8.6	34	9.7	39	11.2
Dog	214	15	7.0	15	7.0	16	7.5
Sheep	382	21	5.5	21	5.5	26	6.8

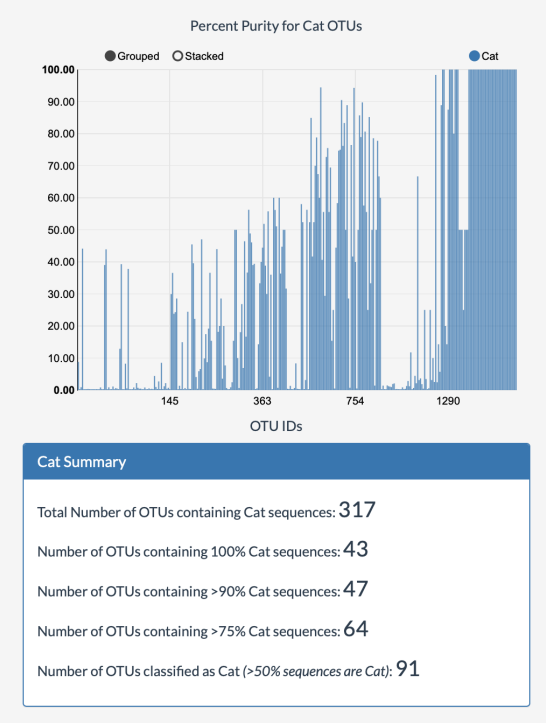


Figure I.1: Cat

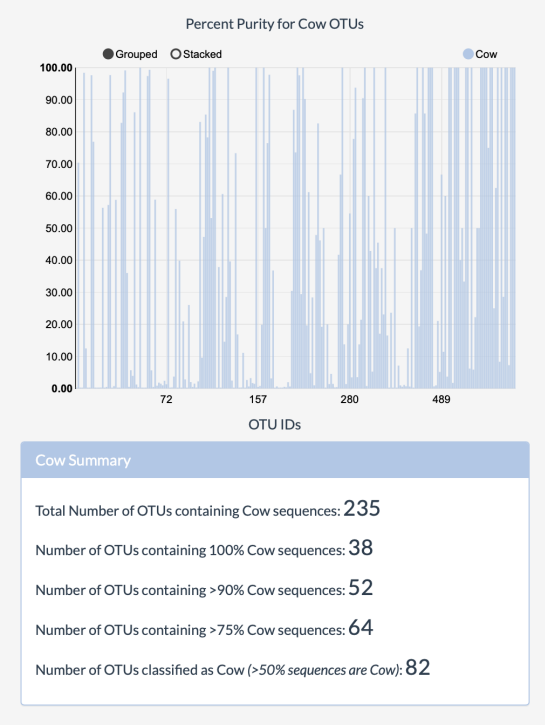


Figure I.2: Cow

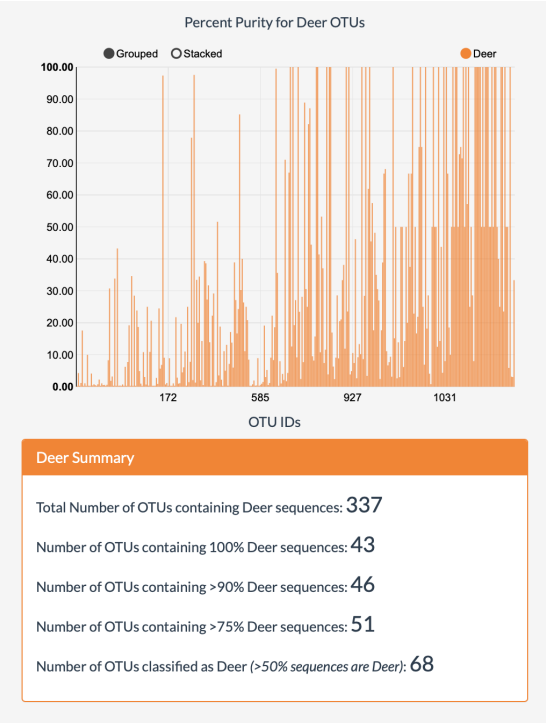


Figure I.3: Deer

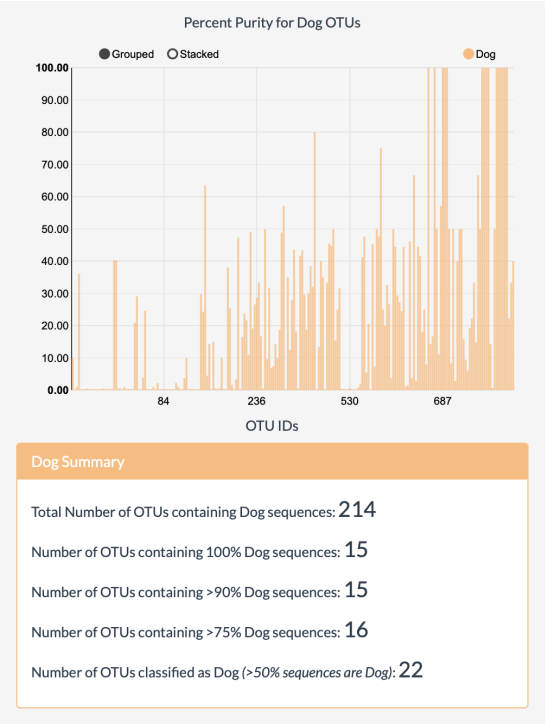


Figure I.4: Dog

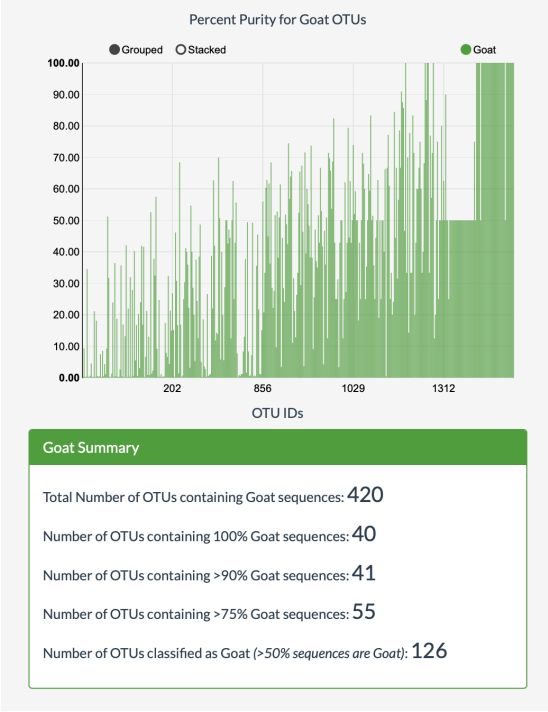


Figure I.5: Goat

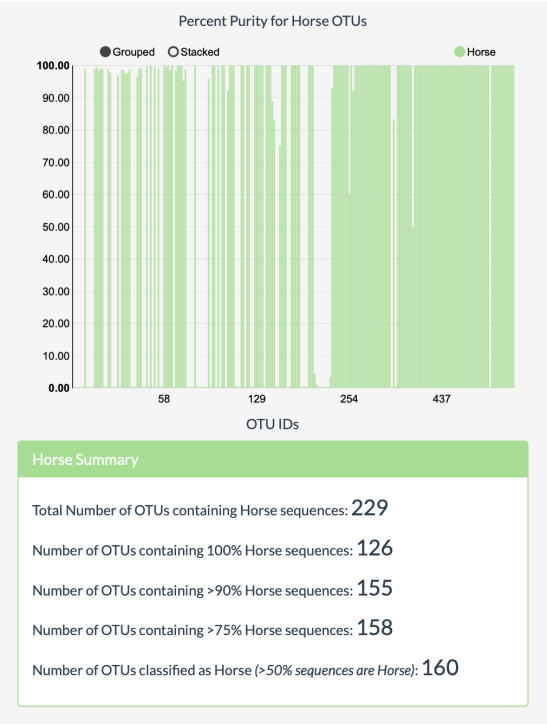


Figure I.6: Horse

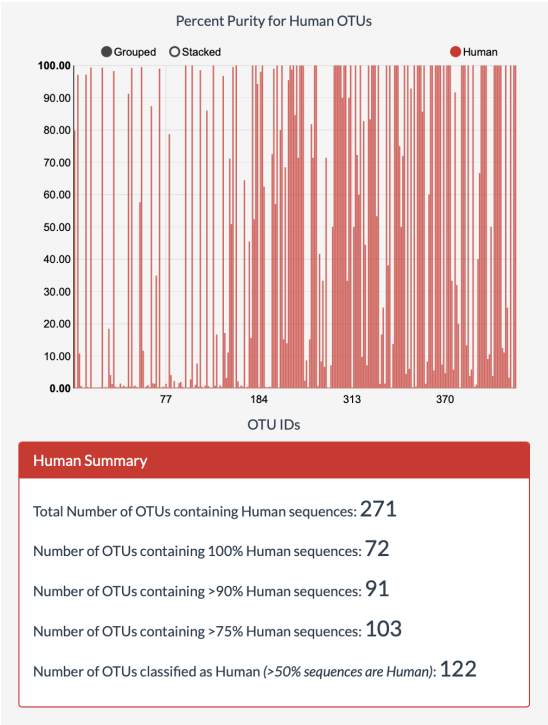


Figure I.7: Human

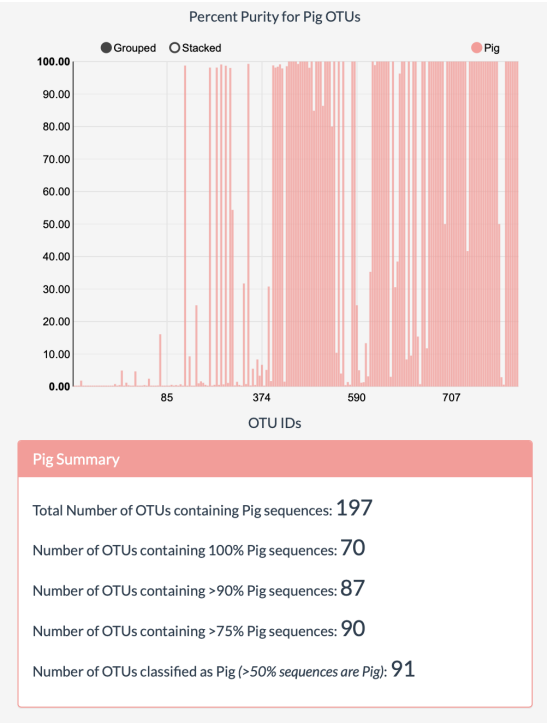
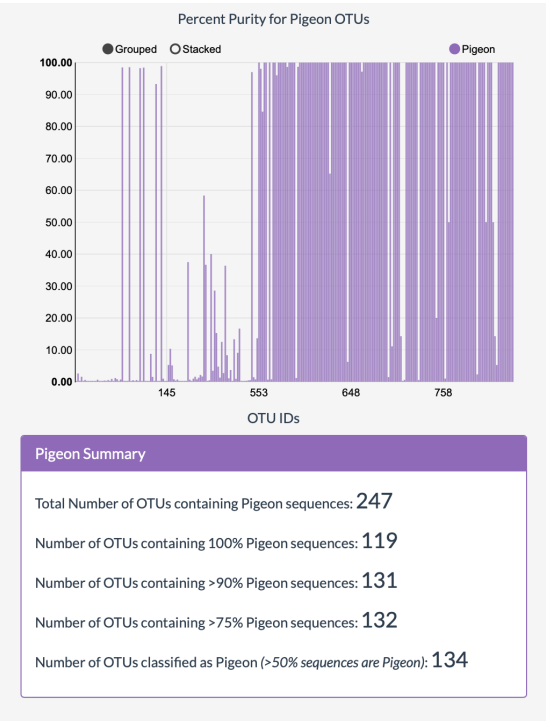


Figure I.8: Pig



Pigeon Summary

Total Number of OTUs containing Pigeon sequences: 247

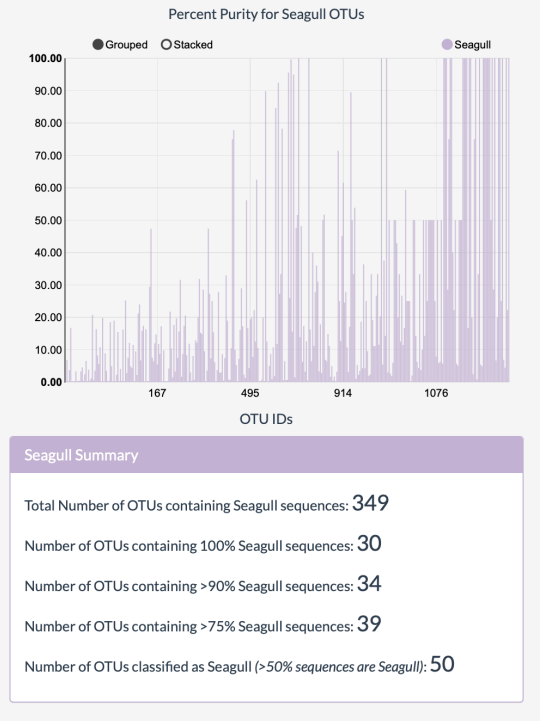
Number of OTUs containing 100% Pigeon sequences: 119

Number of OTUs containing >90% Pigeon sequences: 131

Number of OTUs containing >75% Pigeon sequences: 132

Number of OTUs classified as Pigeon *(>50% sequences are Pigeon)*: 134

Figure I.9: Pigeon



Seagull Summary

Total Number of OTUs containing Seagull sequences: 349

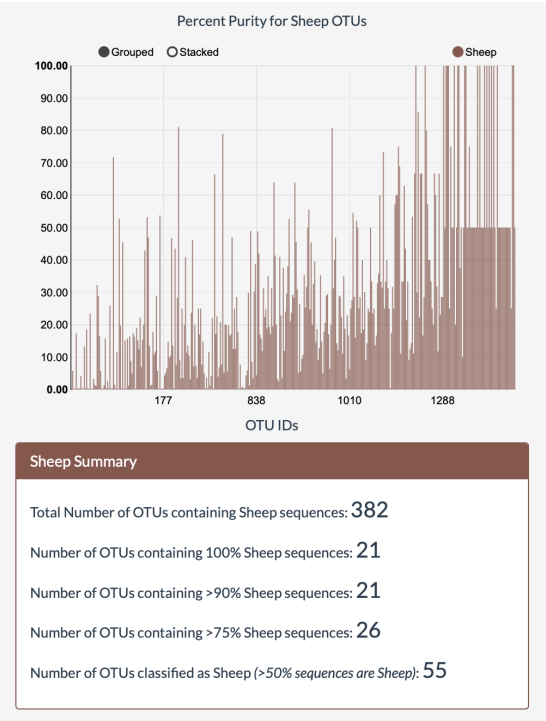
Number of OTUs containing 100% Seagull sequences: 30

Number of OTUs containing >90% Seagull sequences: 34

Number of OTUs containing >75% Seagull sequences: 39

Number of OTUs classified as Seagull *(>50% sequences are Seagull)*: 50

Figure I.10: Seagull



Sheep Summary

Total Number of OTUs containing Sheep sequences: 382

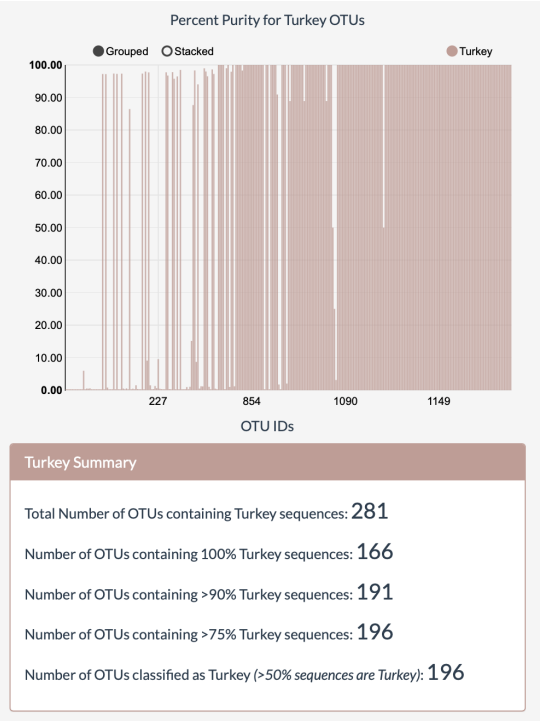
Number of OTUs containing 100% Sheep sequences: 21

Number of OTUs containing >90% Sheep sequences: 21

Number of OTUs containing >75% Sheep sequences: 26

Number of OTUs classified as Sheep *(>50% sequences are Sheep)*: 55

Figure I.11: Sheep



Turkey Summary

Total Number of OTUs containing Turkey sequences: 281

Number of OTUs containing 100% Turkey sequences: 166

Number of OTUs containing >90% Turkey sequences: 191

Number of OTUs containing >75% Turkey sequences: 196

Number of OTUs classified as Turkey *(>50% sequences are Turkey)*: 196

Figure I.12: Turkey

Appendix J

ABBREVIATIONS

ASCII American Standard Code for Information Interchange

bp Base Pair

CDC Centers for Disease Control and Prevention

CPLOP Cal Poly Library Of Pyroprints

DNA Deoxyribonucleic Acid

FIB Fecal Indicator Bacteria

HTS High-Throughput Sequencing

LOTUS Library of OTUs

MST Microbial Source Tracking

NGS Next-Generation Sequencing

nt Nucleotides

- **A** Adenine
- **C** Cytosine
- **G** Guanine
- **T** Thymine

OBMM OTU-Based MST Method

OTU Operational Taxonomic Unit

PCR Polymerase Chain Reaction

PE Paired-End [Sequencing]

RNA Ribonucleic Acid

rRNA Ribosomal Ribonucleic Acid

SE Single-End [Sequencing]

WHO World Health Organization

Appendix K

DEFINITIONS

Adapter

Index sequences ligated to the ends of DNA fragments during the preparation step of NGS. Adapters are used as starting points for read synthesis during the cluster amplification step. Also called Linker or Primer.

Amplicon

DNA sequence produced during PCR amplification. A copy of the original template strand.

Anaerobe

A bacteria that can grow without using oxygen. An obligate anaerobe is a bacteria that cannot grow in the presence of oxygen.

Culture

A colony of bacteria grown on a growth medium such as an agar plate in an incubator

Demultiplex

The process of assigning output reads to the samples they came from using barcodes after Illumina sequencing run

Entropy

Represents the disorder in a cluster. A measure of the extent to which a cluster contains objects of a single class [147].

Indel

Concatenated term referring to either an **in**sertion or a **del**etion of nucleotides

in sequence alignment. Synonym for "gap"

Metagenomics

The study of an entire microbial community in an environment without the need for culturing. Also known as environmental or population genomics. Can also be used to refer to marker gene amplification genomics or 16S metagenomics.

Microbial Source Tracking (MST)

A discipline in Biology that attempts to identify the source of fecal contamination in bodies of water, particularly water used for human consumption or recreation

Microbiome

All the microorganisms in an environment (e.g., human gut microbiome)

Multiplex

The process of applying a unique index sequence (barcode) to the DNA molecules of a given sample before pooling samples in Illumina NGS run

Pathogen

A disease-causing microorganism

Primer

A short sequence of nucleotides used in PCR as a starting point for DNA synthesis

Project

Term used in this document for the final files produced by a single sequencing run for a given user along with all the sample metadata. Includes forward `fastq`, reverse `fastq`, and sample metadata.

Purity

A measure of the extent to which a cluster contains objects of a single class [147]. For OTUs, purity is the measure of the extent to which an OTU contains sequences of a single species.

Read

The output of a sequencing run. A read typically includes both the actual nucleotide sequence and the associated quality score for each nucleotide.

Remediation

Fixing or remedying a problem, specifically attempting to reverse environmental damage

rRNA

The ribosomal RNA which is a cellular component used in protein synthesis

Sequencing Run

Term used in this document for the process of running the experimental sample material through the Illumina NGS platform to obtain the sequencing results. A single sequencing run can contain multiple samples due to the multiplexing feature of Illumina platforms.

Taxonomy

The science of classification of organisms based on shared traits

Zoonosis

An infectious disease transmitted from animals to humans [55]