Preprocessing Algorithms and Software for Genomic Studies with High-Throughput Sequencing Data

A Dissertation

Presented in Partial Fulfillment of the Requirements for the

Degree of Doctorate of Philosophy

with a

Major in Bioinformatics and Computational Biology

in the

College of Graduate Studies

University of Idaho

by

Ilya Y. Zhbannikov

Major Professor: James A. Foster, Ph.D.

Committee Members: Larry J. Forney, Ph.D.; Robert B. Heckendorn, Ph.D.;

Barrie D. Robison, Ph.D.

Department Administrator: Eva M. Top, Ph.D.

May 2015

AUTHORIZATION TO SUBMIT DISSERTATION

This Dissertation of Ilya Y. Zhbannikov, submitted for the degree of Doctor of Philosophy with a major in Bioinformatics and Computational Biology and titled "Preprocessing Algorithms and Software for Genomic Studies with High-Throughput Sequencing Data", has been reviewed in final form. Permission, as indicated by the signatures and dates given below, is now granted to submit final copies to the College of Graduate Studies for approval.

Major Professor: _____ Date: _____
James A. Foster, Ph.D.

Committee
Members: _____ Date: _____
Larry J. Forney, Ph.D.

_____ Date: _____
Robert B. Heckendorn, Ph.D.

_____ Date: _____
Barrie D. Robison, Ph.D.

Department
Administrator: _____ Date: _____
Eva M. Top, Ph.D.

ABSTRACT

DNA sequencing technologies address problems, the solutions of which were not possible before, such as whole genome sequencing or microbial community characterization without pre-cultivation. Current High-Throughput Sequencing (HTS) techniques allow genomic studies in small labs as well as in large genomic centers. Together with modern computational software, HTS becomes a powerful tool, which allows researchers to answer important biological questions in novel ways.

Despite the advantages of modern HTS technologies, large amounts of data and accompanying noise in HTS library confound bioinformatic analysis. Data preprocessing is needed in order to prepare data for subsequent analysis. Data preprocessing includes noise removal as well as techniques such as data reduction.

In this dissertation I present a set of software tools that may be used in genomic studies in order to prepare HTS data for subsequent bioinformatic analysis. The first two chapters in this dissertation describe preprocessing tools developed for data denoising. In the last two chapters I explore the use of multiple genomic markers in 16S data analysis with a meta-amplicon analysis algorithm, which facilitates usage of all the information that can be obtained with 16S amplicon sequencing. Meta-amplicon analysis represents improvements on current methods used to characterize bacterial composition and community structure.

ACKNOWLEDGMENTS

I am deeply thankful to James Foster for his constant support, guidance, and advice during the course of my research. This dissertation would have otherwise been impossible.

I am particularly grateful for the assistance given by Dr. Matthew Settles, who provided valuable and constructive suggestions during the planning and development of this research work.

I would like to thank the other members of my committee, Larry Forney, Robert Heckendorn, and Barrie Robison who provided critical discussions and advice.

I am also thankful to Dr. Samuel Hunter, who provided critical suggestions in the development and completion of the software described in this work.

I also thank my fellow students and the IBEST community, for valuable conversations and discussions.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

High-throughput sequencing technologies are now a routine part of research in biology and medicine, often allowing researchers to address important biological questions in new ways, such as analysis of the whole genome and cultivation-independent microbial community characterization [1-5]. Together with modern computational tools, scientists can perform genomic and metagenomic studies in small labs as well as in state-of-the art genome centers [6-10].

High-Throughput Sequencing (HTS) produces many short sequences or "reads". A typical HTS library may contain billion reads ($x10^6$), 8 billion for an Illumina MiSeq for example. Important problems in genomic studies with HTS data are the large amount of sequence data and accompanying noise [11-13]. Raw sequence data needs to be correctly prepared for any subsequent analysis in order to fully take advantage of the data.

Preprocessing is needed to reduce noise in raw sequence data. There are two main types of noise in raw sequence data: 1) Technical noise caused by sequencing errors, remnants of sequencing adapters, vectors, contaminants or imprecision in the instrument itself; 2) Biological noise, which originates from sequences that are irrelevant to the question, i.e. sequences that do not convey information relevant to the question of interest [14,15].

Technical noise is one of the major problems in sequence analysis. In this dissertation technical noise is divided into two parts: technical noise caused by adapters, poly A/T tails, chimeric sequences and low quality regions; and biological technical noise that includes contaminants and vector remnants. All of these artifacts may significantly degrade bioinformatics analysis. Several application tools have been

proposed to reduce the technical noise in sequence data, from simple trimming of low-quality bases [11], to sophisticated error-correction [13,14]. However, many of these proposed tools do not have desirable capabilities, including the identification of more than one kind of noise and the ability to process sequence data from different technologies.

In Chapter 1 several aspects of sequence pre-processing problems that address reduction of technical noise are considered. I present and evaluate a comprehensive preprocessing software tool, SeqyClean. SeqyClean reduces most technical noise, which, in turn, facilitates comprehensive sequence cleaning, making the downstream analysis more accurate.

Biological noise confounds downstream analysis when one needs to work with reads originating from relatively small discrete units of interest (such as single genes, mitochondria or plasmids), as in exome capture data [16-18], and transcriptomes [19,20]. Researchers in this case can apply any general-purpose mapper, such as some BLAST-like tool [26-31], to recruit all reads that are similar to particular target of interest. This preprocessing task can be a difficult problem, requiring sophisticated algorithms to detect the reads of interest and discard the others. These reads are to be grouped together into a sub-genome unit for downstream analysis. When one needs only the reads similar to the unit of interest, using a general-purpose mappers often is not an optimal choice, because: (1) a general-purpose mapper aligns the whole sequence library to the reference and outputs all reads regardless of alignment to the resulted file, which will require additional work to identify only the reads that align perfectly; (2) most mappers provide no option that would address the read's similarity to the provided

reference; (3) general-purpose mappers usually do not handle reads that belong to the borders of the reference unit of interest.

Chapter 2 develops, implements, and evaluates a reference-based, alignment-free read recruiter tool for high-throughput sequence data using sloppy mapping. This software tool, SlopMap, is developed to serve these purposes. Unlike traditional alignment software, this read-recruiting tool only recruits reads that are similar to the reference unit and can be grouped together. This tool discards irrelevant reads and, in turn, provides more accurate, fast and robust analysis, which is relevant to a specific target. This leads to reduction of biological noise, assembly complexity and analysis time.

Current preprocessing techniques do not take full advantage of data that can be obtained from current HTS technologies in analysis of microbial data. Currently, there are three approaches to analysis of microbial communities: (1) metagenomic analysis, used for characterizing community functional potential; (2) metatranscriptomic analysis, used for describing active gene expression; and (3) 16S rRNA gene sequencing used for characterizing microbial community composition and structure. The first two techniques are not addressed in this work.

In 16S rRNA gene sequencing, researchers apply "molecular fingerprinting": extract total microbial DNA from a sample, and then sequence single, highly conserved and vertically transmitted marker regions that all microbial species likely share. There are two current approaches to fingerprinting analysis: phylogenetic and OTU (Operational Taxonomic Unit) based.

The phylogenetic approach uses marker sequences to identify likely taxonomies for each organism in a sample. Taxonomic units are identified with a naïve Bayes classifier or BLAST-like search against existing microbial databases such as Silva [21], RDP [22], or GenBank. Taxonomic resolution in phylogenetic approach is often limited to the genera or even family level.

The traditional OTU approach clusters marker sequences by similarity. Clustering is highly dependent on the clustering algorithm chosen, the metric, and the threshold that defines the taxonomic unit. It also possible that phylogenies estimated from different markers can differ substantially from phylogenies deduced from known, complete 16S rRNA genes, since clustering can be different for different markers and no single region of a gene carries as much information as the full gene.

These all are important issues in microbial community analyses that can be addressed using multiple markers. This extends traditional OTU analysis techniques from single to multiple markers. In most studies a single marker region is used for community analysis. But data needs to be prepared for subsequent analysis since it becomes non-trivial to combine data sequenced with different markers.

In Chapter 3, Meta-Amplicon analysis, an algorithm for combining sequencing data from different markers is presented. This algorithm utilizes phylogenetic estimates for microbial populations with known genomes. This in turn represents improvements on current methods used to characterize bacterial composition and community structures. The algorithm is highly interdisciplinary, borrowing techniques from image processing, data ordination, and sequence comparison.

Chapter 4 presents the software tool “MetAmp” developed for microbial 16S data analysis with multiple markers. This tool implements the meta-amplicon analysis algorithm presented in the previous chapter, and it works with different types of HTS data.

Chapter 1 is a manuscript in preparation for journal submission: “SeqyClean: a pipeline for high-throughput sequence data preprocessing” by Ilya Y. Zhbannikov, Samuel S. Hunter, James A. Foster and Matthew L. Settles is currently being prepared for publication. Chapter 2 was published in a Special Issue on "Bioinformatics for High-throughput Sequencing” of the Journal of Data Mining in Genomics & Proteomics as “SlopMap: a software application tool for quick and flexible identification of similar sequences using exact k-mer matching”, Ilya Y. Zhbannikov, Samuel S. Hunter, Matthew L. Settles, and James A. Foster, 2013. Chapter 3 is a manuscript in preparation for journal submission: “Analysis of High-Throughput Microbial Amplicon Sequence Data Using Multiple Markers” by Ilya Y. Zhbannikov, Janet E. Williams and James A. Foster. Chapter 4 was published as “MetAmp: combining amplicon data from multiple markers for OTU analysis,” Ilya Y. Zhbannikov, James A. Foster, Bioinformatics 2015; doi: 10.1093/bioinformatics/btv049.

To summarize, this dissertation is organized as follows. Chapter 1 presents SeqyClean, a preprocessing software tool for reduction of technical noise in sequence data. Chapter 2 presents SlopMap, a specific mapping tools that choses sequences relevant to the user’s research question. Chapters 3 and 4 present the meta-amplicon analysis algorithm and corresponding software tool MetAmp, developed for preprocessing microbial 16S data sequenced with multiple genomic markers.

0.1 References

[1] Schatz M.C., Delcher A.L., Salzberg S.L., Assembly of large genomes using second-generation sequencing, Genome Res, 2010:1165-73.

[2] Wooley J.C., Godzik A., Friedberg I., A Primer on Metagenomics, PLOS Computational Biology, PLoS Comput Biol, 2010, 6(2): e1000667.

[3] Collins F.S., McKusick V.A. (2001) Implications of the human genome project for medical science. JAMA 285: 540–544. doi: 10.1001/jama.285.5.540

[4] Metzker M.L., Sequencing technologies - the next generation, Nature Reviews Genetics, 2010.

[5] NIH HMP Working Group, Peterson J. et al., The NIH Human Microbiome Project, Genome Res, 2012 22:1985-1994.

[6] Kuczynski J., Lauber C.L., Walters W.A., et al., Experimental and analytical tools for studying the human microbiome, Nature Reviews Genetics, 2011, 13:47-58.

[7] Li R., Zhu H., Ruan J., et al. De novo assembly of human genomes with massively parallel short read sequencing, Genome Res, 2010, 20:265-72.

[8] Langmead B., Salzbert S.L., Fast gapped-read alignment with Bowtie 2, Nat Methods, 2012; 9(4): 357–359.

[9] Salzberg S.L. and Pertea M., Do-it-yourself genetic testing, Genome Biology, 2010, 11:404.

[10] Schatz M.C., Delcher A.L. and Salzberg S.L., Assembly of large genomes using second-generation sequencing, Genome Res, 2010 20: 1165-1173.

[11] Chou H.H., Holmes M.H., DNA sequence quality trimming and vector removal, Bioinformatics, 2001, 17: 1093–1104.

[12] Huse S.M., Huber J.A., Morrison H.G., Sogin M.L., Welch D.M., Accuracy and quality of massively parallel DNA pyrosequencing, Genome Biol. 2007;8(7):R143.

[13] Kelley D.R., Schatz M.C., Salzberg S.L., Quake: quality-aware detection and correction of sequencing errors, Genome Biol. 2010;11(11):R116.

[14] Criscuolo A., Brisse S., AlienTrimmer: A tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads, Genomics, 2013, 102(5-6):500-6.

[15] Falgueras J., Lara A., Pozo N.F., Canton F., Trabado G.P., Claros M.G., SeqTrim: a high-throughput pipeline for pre-processing any type of sequence read, 2010, BMC Bioinformatics, 11(1), 38.

[16] Hasmats J., Gréen H., Orear C., Validire P., Huss M., Käller M., Lundeberg J., Assessment of whole genome amplification for sequence capture and massively parallel sequencing, PLoS One, 2014, 9(1):e84785.

[17] Campagne F., Dorff K.C., Chambwe N., Robinson J.T., Mesirov J.P., Compression of structured high-throughput sequencing data, PLoS One, 2013, 8(11):e79871.

[18] Hu Y., Willer C., Zhan X., Kang H.M., Abecasis G.R., et al., Accurate local-ancestry inference in exome-sequenced admixed individuals via off-target sequence reads, Am J Hum Genet. 2013, 93(5):891-9.

[19] Kalender Atak Z., Gianfelici V., Hulselmans G., et al., Comprehensive analysis of transcriptome variation uncovers known and novel driver events in T-cell acute lymphoblastic leukemia, PLoS Genet. 2013, 9(12):e1003997.

[20] Sabarinathan R., Wenzel A., Novotny P., Tang X., Kalari K.R., Gorodkin J., Transcriptome-Wide Analysis of UTRs in Non-Small Cell Lung Cancer Reveals

Cancer-Related Genes with SNV-Induced Changes on RNA Secondary Structure and miRNA Target Sites, PLoS One. 2014, 9(1):e82699.

[21] Quast C., Pruesse E., Yilmaz P., Gerken J., Schweer T., Yarza P., Peplies J., Glöckner F.O., The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. 2013, Acids Res. 41 (D1): D590-D596.

[22] Cole J.R., Wang Q., Fish J.A., Chai B., McGarrell D.M., Sun Y., Brown C.T., Porras-Alfaro A., Kuske C.R., Tiedje J.M., Ribosomal Database Project: data and tools for high throughput rRNA analysis, Nucleic Acids Res. 2014;42(1):D633-42.

[23] Kim M., Morrison M., Yu Z., Evaluation of different partial 16S rRNA gene sequence regions for phylogenetic analysis of microbiomes. 2011, J Microb Meth 84(1):81–87.

[24] Foster J.A., Bunge J., Gilbert J.A., Moore J.H., Measuring the microbiome: perspectives on advances in DNA-based techniques for exploring microbial life. 2012. Briefings in Bioinformatics 13(4): 420–429.

[25] Lane D.J., Pace B., Olsen G.J., Stahl D.A., Sogin M.L., Pace N.R. (1985) Rapid determination of 16S ribosomal RNA sequences for phylogenetic analyses. Proc Natl Acad Sci USA 82: 6955-6959.

[26] Altschul S.F., Gish W., Miller W., Myers E.W., Lipman D.J., Basic local alignment search tool, J Mol Biol. 1990 Oct 5;215(3):403-10.

[27] Kent W.J., BLAT-the BLAST-like alignment tool, Genome Res. 2002 Apr;12(4):656-64.

[28] Langmead B., Aligning short sequencing reads with Bowtie, Curr Protoc Bioinformatics. 2010 Dec;Chapter 11:Unit 11.7.

[29] Langmead B., Salzberg S.L., Fast gapped-read alignment with Bowtie 2, Nat Methods. 2012 Mar 4;9(4):357-9.

[30] Li R., Li Y., Kristiansen K., Wang J., SOAP: short oligonucleotide alignment program, Bioinformatics. 2008 Mar 1;24(5):713-4.

[31] Li H., Durbin R., Fast and accurate short read alignment with Burrows-Wheeler transform, Bioinformatics. 2009 Jul 15;25(14):1754-60.

Chapter 1

SEQYCLEAN: A PIPELINE FOR HIGH-THROUGHPUT SEQUENCE DATA PREPROCESSING

Chapter 1 describes SeqyClean, a software application written in C++ with the main purpose of pre-processing HTS data in order to prepare it for downstream analysis. SeqyClean performs comprehensive HTS data denoising and is able to work with Illumina, Roche 454 and Ion Torrent sequence libraries. SeqyClean is open source and available at: http://bitbucket.org/izhbannikov/seqyclean

The manuscript titled "SeqyClean: a pipeline for high-throughput sequence data preprocessing" by Ilya Y. Zhbannikov, Samuel S. Hunter, James A. Foster and Matthew L. Settles is currently being prepared for publication.

1.1 Abstract

Background

Important problems in genomic studies are a very large amount of sequence data, and accompanying noise. Although a plethora of preprocessing software applications has been developed in order to reduce the sequence noise, many of them are not able to handle data from multiple technologies, and only a few are able to provide the reduction of more than one type of noise, thereby limiting the applicability of majority of preprocessing applications.

Results

We developed SeqyClean, a comprehensive preprocessing software pipeline to alleviate these limitations. SeqyClean effectively removes most noise in HTS sequence data and, according our tests, outperforms other best available preprocessing tools.

Conclusions

SeqyClean provides the most complete sequence preprocessing solution, offering all sequence preprocessing aspects (except chimeric filtering). We are constantly working on optimization of the algorithms used in SeqyClean. We hope that the scientific community will benefit from using SeqyClean.

1.2 Background

With the ability to easily outsource genome scale DNA sequencing and advances in modern computation tools, small labs are able to handle research tasks that were previously available only to large genome centers [1]. However, analysis of high-throughput DNA sequence data (HTS) is a non-trivial problem itself. Raw data from high-throughput sequencing machines contain various types of artificial inclusions (sequencing adapters, vectors, contaminants), and sequence errors caused by instrument imprecision, and using the raw data often leads to a poor quality of analysis. Therefore there is a need for data preprocessing before downstream analysis tasks such as mapping and assembly. Aggressive preprocessing of HTS data, before mapping and/or assembly, in order to remove a majority of inclusions (vectors, adapters and contaminants) and potentially erroneous nucleotides is important because it can improve the quality, speed and reliability of analysis [2].

Preprocessing of HTS data can include a number of stages to identify and remove non-experimental artifacts. These include: (1) Identification and trimming of various kinds of DNA sequencing adapters. Sequencing technologies use known short sequences added during library preparation, known as adapters. Untrimmed adapters can lead to a fragmented genome assembly (See Figure 1G from Supplementary Materials). (2) Filter

out non-experimental contaminant sequences. Sequence contamination can come from many sources including, those derived from sample collection and preparation, lab equipment/reagents, or from the sequencing process. As an example, the *Anolis Carolinensis* k51:5946385 transcribed RNA sequence (GenBank ID GAGG011006923.1, ~5.3 kb) is most likely the sequence for Phi-X174 phage (99% identity, according to NCBI BLAST), a common spike-in library for Illumina sequencing [28]. Had the reads been screened for this contaminant prior to assembly, this erroneous transcript would most likely not have been reported (3) Filter and trim vector sequences when a BAC method was used during sample preparation. Such methods insert a DNA fragment of interest into bacterial plasmid ("vector"). Bacterium with such plasmid yielding multiple copies of itself and their plasmid DNA are then sequenced. However, it is likely that plasmid and bacterial fragments are still present in reads. (4) Left and Right read edge trimming of poor quality bases (low quality scores and 'N' bases) produced by instrument imprecision. (5) Combine/join overlapping Illumina paired-end reads. Sequencing library fragments, which are shorter than the overall number of cycles being sequenced, can be overlapped and joined to produce a single, longer read. Combining paired-end reads reduces data abundance and base redundancy, thereby potentially reducing analysis time and errors [5]. (6) Trimming poly A/T tails. When sequencing RNA samples poly A/T sequence is not a part of the original genome and may interfere with a read's ability to map. And finally, (7) Removal of PCR duplicates, which do not convey any new information, but largely increase amount of data and thereby, analysis time.

Current sequence preprocessing tools are divided into tools that perform a single preprocessing stage and those that provide several preprocessing stages packaged together. Many of them have only some of the stages described above, or lack the ability to process multiple sequence data file formats from different sequencing technologies. Roche Life Sciences 454 systems and Life Technologies Ion Torrent/Proton systems use the SFF (Standard Flowgram Format), a binary file format [6], while Illumina, and others, use the FASTQ plain text file format to store both sequenced bases and quality information for each nucleotide and each read. FASTA/QUAL formats are similar but contain only sequence bases or quality scores (numeric) respectively. Table 1G from Supplementary Materials provides a summary of file formats and capabilities for a representative set of other preprocessing applications. Briefly, the applications TrimEST [7] (removes poly A/T tails from sequence), VectorStrip [8] and VecScreen [9] (both trim vector sequences) are designed to perform one preprocessing task. Lucy [10] and AlienTrimmer [2], FASTX-Toolkit [27], SeqTrim [11], Ea-Utils [12], AdapterRemoval [13], Skewer [14], Trimmomatic [15], Btrim [16], Cutadapt [17], Sickle [18] are examples of applications that perform multiple preprocessing stages, mostly adapter and quality trimming. Lucy is a preprocessing pipeline for quality, poly A/T and vector trimming. It was designed for lower-throughput Sanger sequence data and does not accept as input the modern SFF and FASTQ file formats or Illumina paired-end libraries. AlienTrimmer accepts both single-end and paired-end reads in FASTQ format, providing adapter and quality trimming. AlienTrimmer does not accept SFF files, requiring the user to first convert them to FASTQ format. FASTX-Toolkit is a bundle of individual preprocessing software tools that perform a set of preprocessing stages. FASTX-Toolkit

provides quality and adapter but also does not handle SFF files or vector/contaminants screening. SeqTrim is a tool for vector, contaminants and quality trimming. SeqTrim utilizes the BLAST algorithm for discovery of adapters, vectors and contaminants. SeqTrim accepts only FASTA/QUAL format and requires standalone BLAST application to be installed on the user's machine. Btrim performs quality and adapter trimming, but no vector and contaminants screening, poly-A/T removal or other preprocessing stages. Ea-Utils performs a number of stages, including quality, poly-A/T, adapter and vector and contaminants screening. It also accepts single- and paired-end reads and, moreover, it performs merging of overlapping paired-end reads. But does not process SFF binary files. Skewer, Trimmomatic, Cutadapt, AdapterRemoval and Sickle perform adapter and quality trimming of single- and paired-end reads. However, none of these applications incorporate all of the stages mentioned above into a single preprocessing pipeline. Therefore there is a need for a universal preprocessing tool that incorporates all of the important preprocessing stages described above and is flexible enough to work on the most common types of data formats available.

In this paper we present SeqyClean, a comprehensive preprocessing software pipeline for high throughput sequence data. The purpose of SeqyClean is to incorporate all of the sequence preprocessing stages together into one bioinformatics pipeline that works with the two most common sequence data formats, SFF and FASTQ files (both single-end and paired-end). SeqyClean successfully recognizes and removes technological components, contaminants and vectors. Further, SeqyClean provides quality trimming, poly A/T trimming, overlapping of paired-end reads and PCR-duplicate detection and removal. Further, SeqyClean allows the user to choose which stages to

perform and to adjust the default parameters within stages, as the experimental conditional may need. We show that SeqyClean greatly improves both genome de-novo assembly and genome mapping. SeqyClean has been extensively used in our lab with continued improvement and we believe that the research community can also benefit from its use.

1.3 Implementation

SeqyClean is an open-source software application that requires only the GNU C/C++ compiler installed and is available for download from the following link: http://bitbucket.org/izhbannikov/seqyclean. A workflow diagram is shown in Figure 1.1 and comprises of the following stages: (1) Input data preprocessing; (2) PCR-Duplicates removal; (3) Overlapping and adapter removal for paired-end reads; (4) Trimming poly A/T tails; (5) Vector trimming; (6) Contaminant removal; (7) Adapter trimming for single-end reads (454, Illumina single-end); (8) Quality trimming; (9) Establishing final trim points; (10) Generating output files and summary statistics. An advantage of SeqyClean is its modular structure, in which the user can specify different preprocessing strategies, rather than a strictly determined workflow. It also incorporates all of the preprocessing stages described in the Background section into single bioinformatics application, providing the most powerful sequence cleaning.

We describe each preprocessing stage in detail in the following sections. SeqyClean also has a number of additional utility functions that may be useful in preprocessing HTS data, such as minimum read length filtering, and conversion from the newer CASAVA v1.8 style FASTQ read IDs to the older pre-CASAVA 1.8 style read IDs. Refer to the user manual for additional information.

Removal PCR-duplicates from sequence data

During library preparation, the amount of DNA is typically amplified using a polymerase chain reaction (PCR) step. If library complexity is low, these duplicated templates are often sequenced multiple times, but because all of the sequenced reads originate from the same template they do not provide any additional information. Identification and removal of reads that are likely the result of PCR duplication reduces the pool of redundant sequence before assembly and mapping, which can potentially reduce both errors and analysis time. In our algorithm we define PCR duplicates as reads originating from the same parent molecule. For paired-end reads, PCR duplicates are identified if they share the same 35bp window (default) starting from basepair 10 (default) from the 5' end (70bp total must be an identical matches, 35bp from each read) and ending at base 45 from the 5' end of each pair. For single-end reads (Roche 454 of single-end Illumina) 35bp must be identical. The window size, start position and maximum number of allowed duplicates can be adjusted by changing parameters '*startdw*' (starting position, 10 by default) and '*sizedw*' (window size, 35bp by default) and '*maxdup*' (maximum number of duplicates to keep in preprocessed librar, 1 by default).

Poly A/T tail trimming for RNA sequence data

SeqyClean provides trimming of poly-A/T of RNA sequence data by implementing the approach previously introduced in the Lucy application [10]. The algorithm begins by searching for the first minimum occurrence of 10bp (by default) or longer of poly-T (or poly-A) fragment within the first initial search range of 50bp, from both the 5' and 3' ends of sequence, and then attempts to extend this poly-T (poly-A) seed outward,

allowing for no more than the maximum (default=3) mismatches between every min span (default=10bp) consecutive T (A) bases in the search region.

Contaminant and vector detection and trimming

To identity contaminants and vectors we use exact k-mer matching, which works as follows: supplied contaminant references sequences are sampled into consecutive k-mers (15 bases long by default), each k-mer is stored in a hash table; Searching for a contaminant, or vector, is performed by sampling each read into consecutive k-mers (15bp length by default) and then performing a hash table lookup for each k-mer in a read. For flexibility, the distance (difference between starting position of the first k-mer and starting position of the second k-mer) between tandem k-mers in a read can be adjusted (default distance is one base). In the case of vector sequences, matches are first approximately found via consecutive k-mer matching, and then the exact coordinates are obtained by extending with a pairwise alignment between the read and corresponding region in a reference vector sequence. The read is trimmed of vector sequence if the match occupies less than 80% of the read and discarded otherwise. In case of contaminants screening the whole read is discarded as soon as it meets three (by default, can be adjusted) consecutive successful k-mer matches. In order to identify contaminant and vector sequences, reference sequence of expected contaminant/vectors must be provided in FASTA format.

Adapter identification and combining overlapping Illumina paired-end sequences

To trim adapters in single-end and reads compressed in SFF files [6], we use the SSAHA algorithm [20] for detection of Roche 454 RL MID adapters in SFF files and Illumina Truseq adapters in single-end reads, by default; however the user can supply

custom adapters. SSAHA is a hashing algorithm used for mapping reads, but implemented for use in adapter identification here.

In the case of *paired-end* reads, reads that contain adapter sequence will also contain a high quality, interior overlapping alignment where the overhangs are the adapter sequences. This technique is also described in AdapterRemoval [13] and Trimmomatic [15] and is based on the following assumption. If an adapter is found in one paired-end read, the corresponding adapter should also be present in the corresponding pair, while the interior sequences (non-adapter sequence) are reverse complements of each other. Otherwise, the read is free of adapter sequence. The algorithm is shown in Figure 2G in the Supplementary Materials and is described as follows:

- First, the read pairs (read 1 and the reverse-complement of read 2) are fully overlapped (Figure 2G(a)) and an overlap score is computed according to the following equation:

$$Score = \frac{\#\,matches}{overlap\_length} \qquad (1)$$

- Then reads are shifted by one base (to the interior), relative to each other (Figure 3G(b)) and overlap scores are computed after each shift. If an overlap score reaches a pre-defined threshold (0.75 by default), the overlap position is recorded, the algorithm terminates and adapters are trimmed (Figure 2G(c)). Otherwise, the algorithm proceeds until the min overlap (=16 bases by default) is achieved.

A potential benefit of the paired-end paradigm is the possibility to join paired-end reads into single, longer sequence with a length up to nearly twice the length of the individual reads, which can reduce assembly complexity. This is implemented as part of

the paired-end adapter-trimming algorithm described previously, except it contains an additional consensus calling stage, in which the paired-end reads are combined into longer single read ('consensus read') with re-computed bases and quality scores. If disagreements are identified while constructing the consensus, the nucleotide call with the higher quality score is selected and the quality is set to the difference in quality scores between the two calls. In cases where quality scores are equal for unmatched calls, the residue for read 1 is reported with a quality of 2.

Quality trimming

SeqyClean utilizes the approach used in the Lucy quality and vector trimming application [10]. Briefly, (1) the quality trimming algorithm searches for the longest region from both the 5' and 3' ends with average error lower than a predefined *bracket_error* threshold. To do this, a sliding window is applied with size of 10 bp by default. Low-quality regions, which do not meet the criteria, are trimmed from each end; (2) At the next step, a quality algorithm identifies regions within a sequence that have high error rates. To do this, two types of windows are applied: first, a large window of 50 bp (default) is applied. Second, a smaller window of size 10 bp (default) is used to identify small low-quality regions that were missed using the larger window. (3) The largest high-quality region with average error lower than *maximum_average_error* (MAE) is then chosen as a candidate for the final high quality range; (4) The last step is to check the last two bases from each end of the read. If they have an average error larger than the *maximum_error_at_ends* (MEE) parameter, trim them off. Parameters *maximum_average_error* and '*maximum_error_at_ends*' are each set to 0.01 (20 Phred) by default and represent the quality trimming threshold. In SeqyClean *bracket_error* is

set to 0.794 and the default maximum average error for both windows (50 and 10 bp) is also set to 0.794.

Final Trim Points

Trim points are positions on the 5' and 3' ends of the read and define the good-quality region that passed the quality trimming stage and are free from adapters, vector and polyA/T sequences.

Output files and summary statistics

This includes processed sequence data, and log files that record all parameters and flags used in preprocessing, summary statistics such as the number of discarded sequences, adapters found, etc., and statistics for each read (adapter position, read length, etc.).

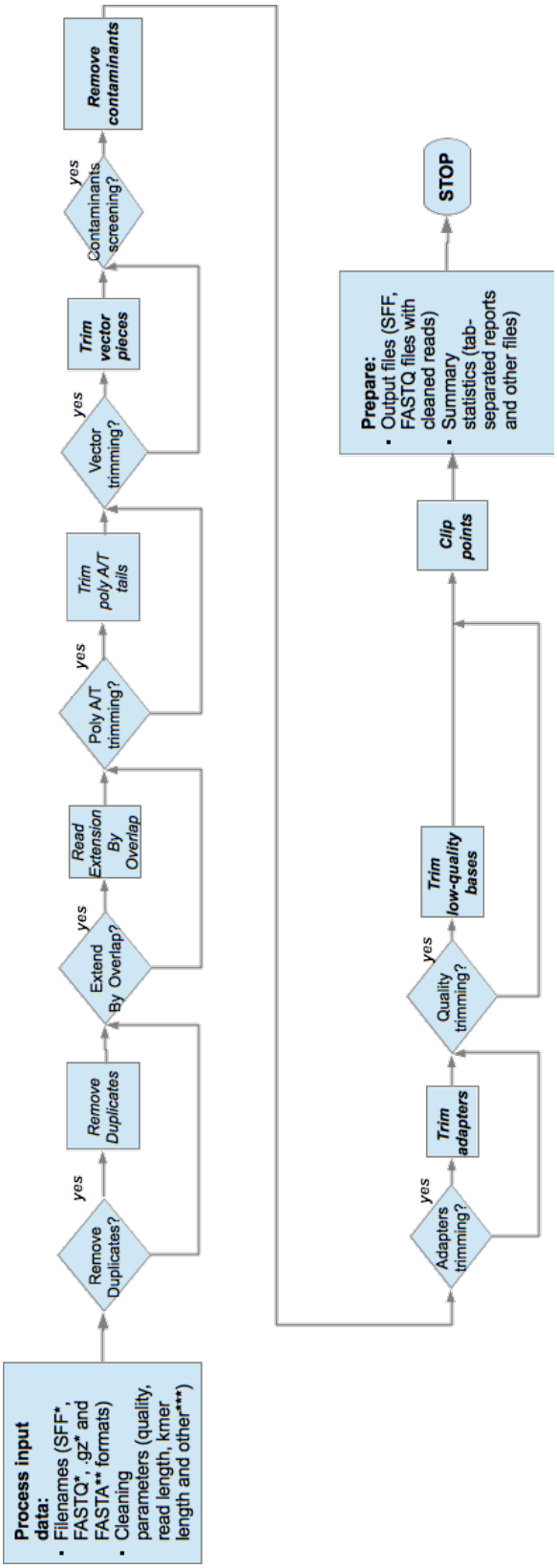

Figure 1.1: The workflow diagram of SeqyClean.

Evaluation

We chose to evaluate SeqyClean by comparing *de-novo* genome assembly and mapping statistics using both original and processed reads. Our goals were to (1) identify a set of optimal quality parameters for general sequence preprocessing tasks; (2) to evaluate the influence of different pre-processing stages on de-novo genome assembly and genome mapping and (3) compare the preprocessed datasets of SeqyClean to other tools. We use the QUAST analytic tool [24] to extract statistics across all tests.

Datasets

We evaluated SeqyClean on three HTS shotgun datasets, in order to compare SeqyClean to other available HTS preprocessing tools and determine the optimal quality parameters for most sequence preprocessing tasks. The first is a public dataset consisting of 400,596 Illumina MiSeq paired-end 250bp reads sequenced from the *E. coli* K-12/MG1655 bacterial genome (SRR519926). The second is a Roche 454 pyrosequencing dataset containing 621,578 reads and 327,471,374 bases derived from *E. coli*. K-12/MG1655. The third also consisted of Illumina MiSeq paired-end 250bp reads sequenced from the eukaryotic genome of *Saccharomyces cerevisiae* (Yeast) W303 with a total of 3,875,453 reads. Roche 454 *E. coli* and Illumina *S. cerevisiae* datasets were sequenced in the University of Idaho IBEST Genomics Resources Core. For reference sequences, we used *E. coli* str. K-12 substr. MG1655 strain [GenBank accession number NC000913.3] and *S. cerevisiae* W303-K60001 strain [GenBank accession number ALAV00000000.1] for the *E. coli* and *S. cerevisiae* datasets respectively [26].

Optimal quality trimming parameters

We conducted 39 preprocessing experiments both on the *E. coli* 454 and *S. cerevisiae* Illumina libraries, in which we applied SeqyClean to these libraries using quality error thresholds ranging from 0.794 (Phred of 1) to 0.000126 (Phred of 39, inclusive). In these tests, we simultaneously changed the *maximum_average_error* (MAE) parameter and *maximum_error_at_ends* (MEE). A value of 0.794 indicates a 79.43% chance that the base was read incorrectly and a value of 0.000126 indicates a 0.0126% chance that a nucleotide base was read incorrectly. Complete Phred quality values and corresponding error rates are in Table 1F of Supplementary Materials. All other parameters were left at their defaults. Detailed execution commands are given in Supplementary Materials. Optimal quality trimming parameters were inferred from de-novo assemblies and mappings of preprocessed *E. coli* pyrosequence library and *S. cerevisiae* paired-end Illumina libraries.

Optimal parameters for Roche 454 preprocessing were estimated from de-novo genome assembly

We performed 39 de-novo assemblies of raw and SeqyClean processed 454 data, using the Roche GS De Novo Assembler (Newbler) v2.9 using assembly parameters given in the Supplementary Materials, section C). We also evaluated Roche's native clip points by supplying a library file to the assembler "as is", letting the assembler determine the optimal preprocessing strategy from clipping information encoded in the SFF file (*E. coli* pyrosequence library only). We used the following assembly assessment statistics: (1) N50 [30]; (2) total number of contigs; (3) average read length; (4) read coverage; (5) insertions/deletions (INDELs); (6) total length of the assembly.

Optimal parameters estimated from genomic mapping

To evaluate the effects of quality and adapter trimming on genomic mapping, we used the same *E.coli* and *S. cerevisiae* test libraries preprocessed with SeqyClean as described above. We estimated the number of single-nucleotide polymorphisms (SNPs) and insertions/deletions (INDELs) and then performed a comparison to the mapping of raw data and data preprocessed with clipping information obtained from the SFF file (not applicable to Illumina libraries). We used Bowtie2 [23] aligner and SAMtools [22] in order to extract SNPs and INDELs. All parameters we used for mapping tests are given in Supplementary Materials.

Evaluation of effects of different pre-processing stages on genome assembly and mapping

We assembled the (1) *E. coli* Illumina, the (2) *S. cerevisiae* Illumina paired-end datasets and mapped the processed reads onto the corresponding reference sequence We evaluated the following stages: de-duplication, merging paired-end reads, quality and adapter trimming. We performed five assembly tests: (1) including all these stages; (2) excluding de-duplication; (3) excluding de-duplication and merging paired-end reads (only adapter and quality trimming are left); (4) excluding de-duplication, merging and quality (only adapter trimming remained); (5) assembling all the raw data. We used N50, total length of assembly, number of large and total contigs and assembly time in order to compare results. For mapping we only used percentage of mapped reads (paired- and single-end reads). Assemblies were performed with Roche 454 Newbler 2.9 assembler on 32-core (64 threads) Unix machine (for these tests we used 4 threads) and mapping was performed with the Bowtie 2 mapping tool using 16 threads, on the same machine.

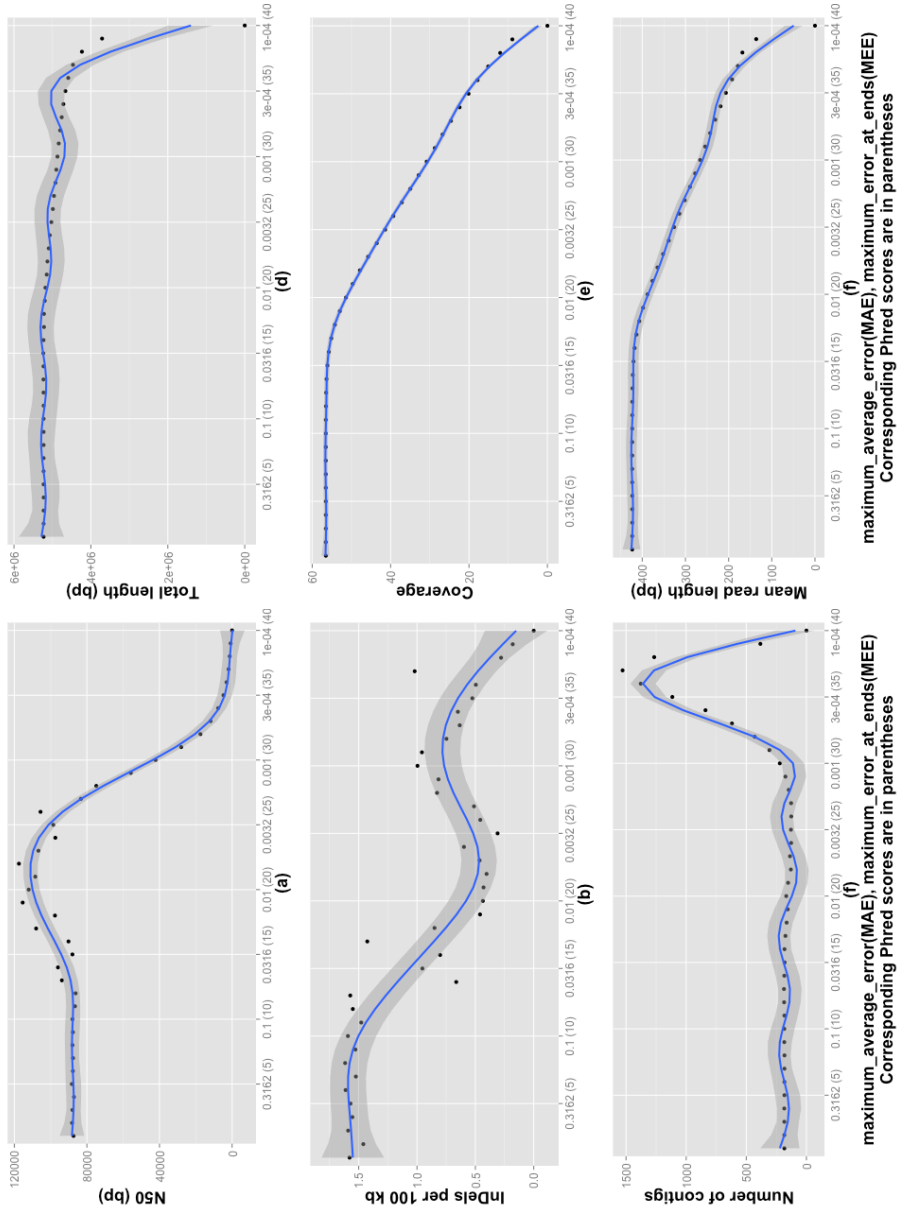

Figure 1.2: Effects of quality and adapter trimming on de-novo assemblies of the *E. coli* pyrosequence data. Horizontal axes for all graphs show the Quality trimming Threshold (QT), which is a pair {*maximum_average_error, maximum_error_at_ends*} and corresponding Phred scores, given in parentheses. QT was varied from 0.794 to 0.00012 (1 to 39 Phred), and parameters for adapter trimming were left to their defaults. (a) Effect of quality and adapter trimming on assembly N50; (b) INDELs per 100 bp; (c) Total number of contigs for SeqyClean; (d) Total length of the assembly as a sum of all contigs; (e) Coverage; (f) Average read length; This statistics was made with QUAST analytic tool.

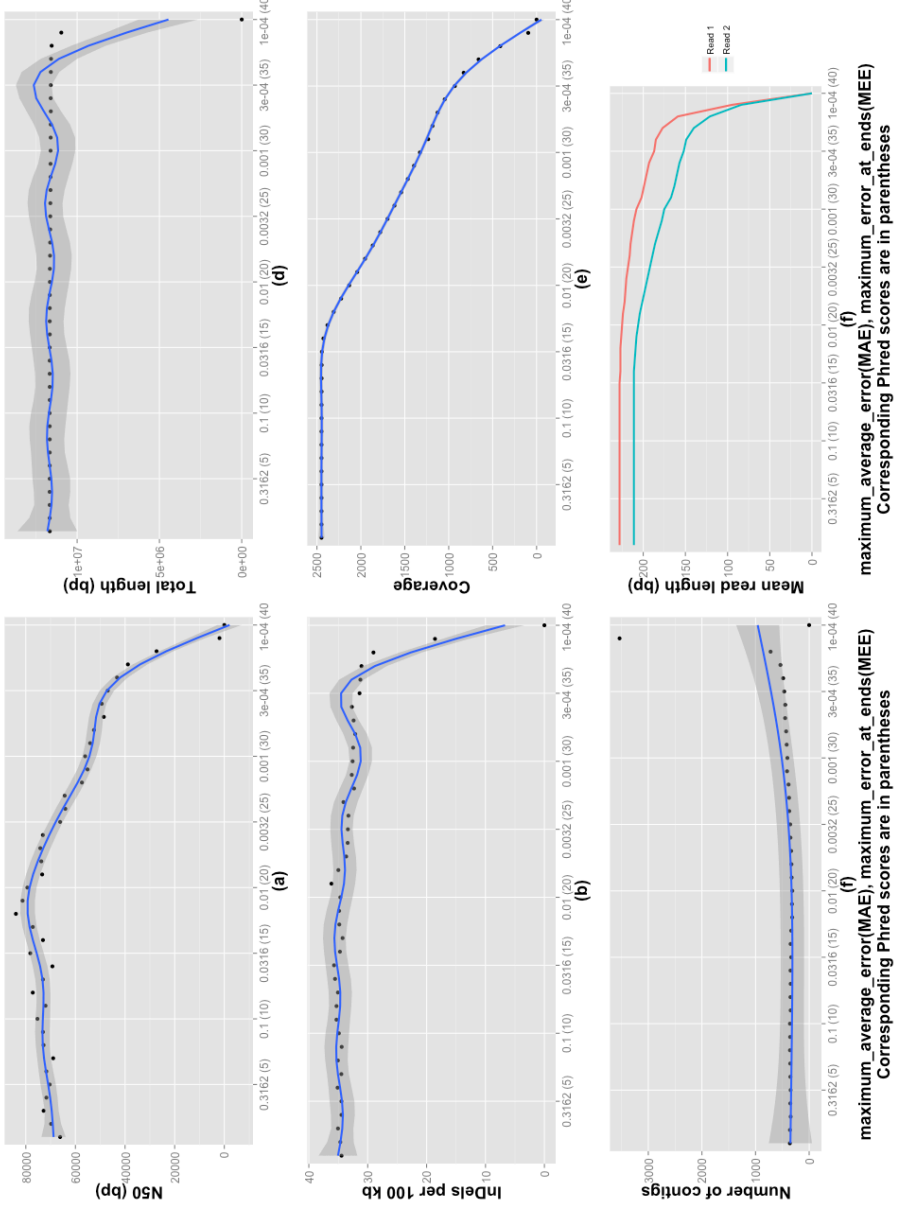

Figure 1.3: Effects of quality and adapter trimming on de-novo assemblies of the *S. cerevisiae* Illumina paired-end data. Horizontal axes for all graphs show the Quality trimming Threshold (QT), which is a pair *{Maximum average error, Maximum error at ends}*, in Phred scores. (a) Effect of quality and adapter trimming on assembly N50. (b) INDELs per 100 bp. (c) Total number of contigs for SeqyClean. (d) Total length of the assembly as a sum of all contigs. (e) and (f) shows coverage and mean read length (for paired-end reads) against different QT values.

Comparison of SeqyClean to other preprocessing tools

We compared SeqyClean to Lucy, AlienTrimmer, Btrim, and Ea-Utils, using the *E. coli* Roche 454 dataset. We compared it to AlienTrimmer, Ea-Utils, Skewer, Trimmomatic, AdapterRemoval, and Sickle using the *E. coli* and *S. cerevisiae* Illumina data. We preprocessed each library using the default/suggested parameters provided by authors of each application. For preprocessing with SeqyClean we used the same parameters found in tests from previous section. The complete set of parameters used for these tests is given in Supplementary Materials. After preprocessing we performed de-novo assemblies with Newbler We compared the following assembly statistics: N50, number of large contigs (length of contig >= 1000 bp), total number of contigs, largest contig length, largest alignment and total length of the assembly. We also performed mapping with Bowtie2 [23]. We used 16 threads and other parameters were left at their defaults. For mapping we compared the number of reads aligned for each application, assuming that higher percentage of aligned reads indicates better preprocessing.

1.4 Results and Discussion

In this paper we have introduced SeqyClean, a comprehensive data pre-processing tool that facilitates de-noising of high-throughput sequence libraries. Data de-noising is a necessary step before analyzing sequence data and SeqyClean is especially designed to perform the most complete sequence cleaning. To test SeqyClean, we: (1) determined optimal default parameters for SeqyClean by determining the effects of quality and adapter trimming on genome assembly and mapping of typical datasets; (2) evaluated of effect of different pre-processing stages on de novo assembly and genome mapping; (3) compared the effectiveness of SeqyClean to other pre-processing tools.

Optimal quality trimming parameters for the majority sequence preprocessing tasks

Effects of quality and adapter trimming on assembly of the E. coli pyrosequence library

The maximum N50 (from 106,740 bp to 117,486 bp) is achieved when SeqyClean trims reads with quality trimming threshold between 0.126 and 0.005 (19 and 23 Phred), Figure 1.2(a). When the quality trimming threshold is below 0.063 (12 Phred), the N50 of the assembly of data preprocessed with SeqyClean is slightly larger than the N50 of assembly using only clipping information from SFF library and significantly larger than assemblies of raw (unprocessed) data. N50 then peaks at around 0.01 (20 Phred) and then quickly drops to zero as the quality trimming threshold becomes more stringent, due to discarding too many reads, reducing overall coverage.

The lowest number of INDELs (from 0.31 to 0.46) is seen for a quality trimming threshold between 0.01 and 0.0025 (20 and 26 Phred), Figure 1.2(b). Assembly recovers fewer INDELs per 100kb with a more stringent quality and far fewer than with raw, untrimmed data. Low-quality bases can produce false-positive INDELs, removing low-quality nucleotides and low-quality regions can then potentially decrease false-positives. On the other hand, the read coverage should be high enough in order to obtain proper estimates of INDELs. Any quality trimming should not be too stringent in order to avoid significant data loss.

The total number of contigs is fewest (varies from 127 to 155) for the quality trimming threshold set between 0.0126 and 0.0032 (19 and 25 Phred), Figure 1.2(c). It is generally better to have a fewer large contigs with total length close to the expected size rather than many relatively short contigs.

Effects of quality and adapter trimming on mapping of the E. coli pyrosequence library

The total number of INDELs and SNPs is lowest (25-30 for SNP, 95-196 for INDELs) for quality trimming threshold between 0.0032 and 0.0016 (25 and 28 Phred), Figure 1.4(a). In this dataset, there are significantly more INDELs than SNPs since Roche 454 technology tends to form homopolymers [25]. The number of INDELs and SNPs decreases significantly after threshold 0.0032 (25 Phred) as coverage becomes too low for proper SNP/InDel calling (Figure 1.4(c)).

Effects of quality and adapter trimming on assembly of the S. cerevisiae Illumina paired-end library

The total N50 for the *S. cerevisiae* Illumina library is highest (77,197 bp – 83,992 bp) for error trimming threshold of 0.020 to 0.01 (17-20 Phred), Figure 1.3(a)). The average read length of the first and second paired end reads (PE1 and PE2) does not change until the quality trimming threshold achieves 0.020 (18 Phred), Figure 1.3(f). The anticipated coverage decreases after the trimming quality threshold achieves the value of 0.025 but does not change before this value (Figure 1.3(e)). The number of INDELs for *S. cerevisiae* data is lower after trimming than with raw data, except when a trimming error threshold greater than 0.001 (Figure 1.3(b)).

Effects of quality and adapter trimming on mapping of the S. cerevisiae Illumina paired-end library

Optimal quality trimming parameters (MAE and MEE) are within the interval 0.01-0.0032 (20-25 Phred) according to results in Figure 1.4(b). This opposite of the *E.coli* pyrosequence test library situation: the number of variance (SNPs) is higher than number of InDels (Figure 1.4(b)). This was expected, since Illumina data have fewer erroneous homopolymers than Roche 454 technology.

Optimal parameters

In summary, a quality trimming parameters (MAE and MEE) should be set between 0.01 (20 Phred) and 0.0032 (25 Phred). In general, when the quality threshold becomes greater than 0.0032 (25 Phred), the preprocessing tends to discard potentially good-quality data therefore setting up MAE and MEE below 0.0025 (26 Phred and higher) in generally not recommended. To summarize this, the recommended settings for MAE and MEE are:

- 0.005 (23 Phred) for Roche 454 data.
- 0.01 (20 Phred) for Illumina MiSeq data.

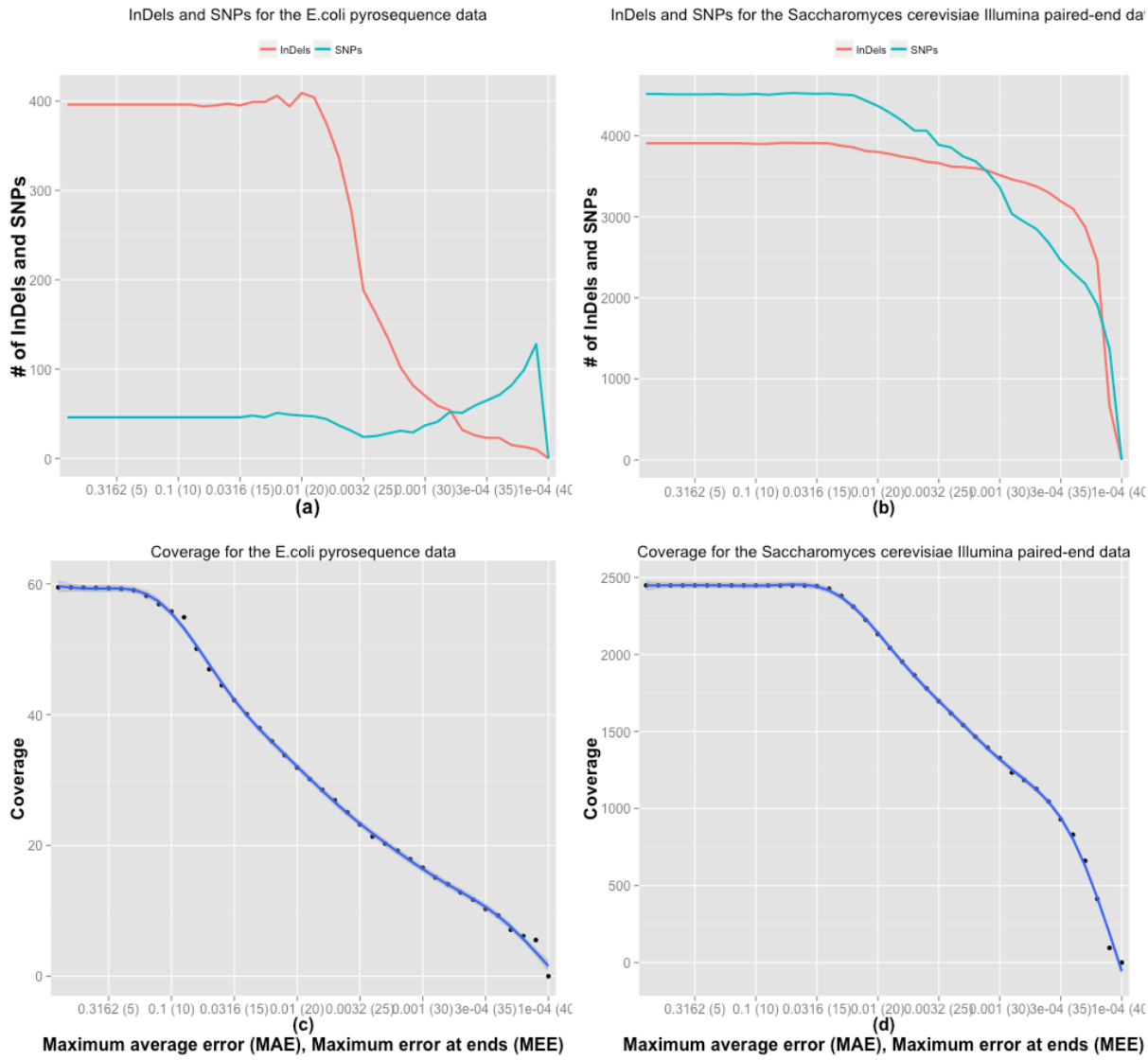

Figure 1.4: Mapping *E.coli* and *S. cerevisiae* data sets. (a) INDELs and SNP for the *E. coli* data. Note the number of INDELs is significantly higher then SNPs indicating that pyrosequence technology produced many unnecessary homopolymers; (b) INDELs and SNPs for the *S. cerevisiae* data set. In this case there is a reverse pattern, in which number of INDELs is lower than SNPs; (c) and (d) represent expected coverage for *E.coli* and *S. cerevisiae* data sets.

Effects of different pre-processing stages on genome assembly and mapping

E. coli Illumina MiSeq data

For the *E. coli* Illumina MiSeq data (2x250bp reads, 400,596 total reads), quality trimming has the greatest impact on assembly and mapping (Table 2G and 3G from Supplementary Materials). De-duplication has little effect on the *E. coli* paired-end library, which represents shotgun sequencing of the whole genome at less than 60x coverage, so only a few duplicated sequences were found (85 duplicates in total). However, this could impact datasets whose coverage is high. Merging paired-end sequences also has a significant effect on assembly, but has little or no effect on mapping.

Adapter trimming has a little impact as it is seen from this particular dataset, where a few (13.45% of reads) contained adapters. The four tests were identical in their ability to detect adapters and retain reads. In the first test (all preprocessing stages were included), 13.45% of TruSeq adapters were found, 85 duplicated reads were found, and ~86.8% of total reads were kept (59.91% paired-end, 13.45% perfectly merging reads and 13.44% were unmerged unpaired reads), 13.2% of total reads were discarded.

The second test (without duplicates screening) shows the same results (due to only 85 duplicates that were found in the previous test, which has a little impact on results). In the third test (excluded stages: duplicates screening, merging paired-end reads by overlap), 13.45% of adapters and 86.8% of data were kept (72.93% paired reads, 13.87% unpaired reads) and 13.2% (total) reads were discarded. The fourth and final test (without duplicates, read overlapping and quality trimming), 13.45% of adapters were found and 100% (paired-end) reads kept (no single unpaired reads).

The best assembly results are from data preprocessed with SeqyClean at all stages of the pipeline (Table 2G from Supplementary Materials). SeqyClean pre-processing results in the largest N50 (82,666), a total assembly length close to reference, the fewest contigs (117 large/170 total), and a relatively low assembling time (4min, 14 sec). Assembling times for non-overlapping reads are slightly higher (5min, 47 sec) than those produced from merged pair-end reads, where the overlapping option was 'on'. This shows that the assembler does a better job, and does it more quickly when paired sequences can be merged. Unsurprisingly, the lowest assembly quality is from raw, unprocessed reads.

The best mapping results were also obtained with data preprocessed with all de-noising stages. Quality trimming made the most difference. By contrast, de-duplication and merging paired-end reads had little effect on mapping. Trimming adapters also had little effect, due to low presence of adapters in the library (13.45%). The worst results were from data with trimmed adapters only and unprocessed data.

E. coli Roche 454 pyrosequence data

For the *E. coli* Roche 454 pyrosequence data (621,578 single-end reads in total), quality trimming has the greatest impact on assembly quality, and quality and adapter trimming has the greatest impact on mapping quality (Table 4G and 5G from Supplementary Materials) as well.

De-duplication significantly influences assembly time (5min 38sec against 7min 33sec without), 8,923 duplicates were found (Table 4G from Supplementary). De-duplication also affects assembly quality, producing better statistics than without de-

duplication. De-duplication reduces complexity for an assembler, allowing it to form fewer and longer contigs with total size close to the reference size.

Adapter trimming (96.2% reads were found with 5' adapters and 99.8% of the reads found with 3' adapters) has a small impact on assembly quality, producing roughly the same N50 (82,861bp against 82,669bp) and other statistics in comparison to assembly from raw data. But assembly time for data, preprocessed with only adapter trimming stage is lower (11min 3 sec) that for assembly performed from unprocessed reads (14min 9 sec).

Mapping statistics is concordant to the assembly one (Table 5G from Supplementary). The best mapping results were obtained from data preprocessed with all stages (de-duplication, quality and adapter trimming), with 88.40% of properly mapped reads. Without de-duplication (only quality and adapter trimming), there are 88.35% of reads aligned. This may indicate that duplicated sequences had low quality thereby decreasing amount of mapped reads.

With adapter trimming only (no de-duplication, no quality trimming) we have 84.02% reads properly mapped, which indicates that quality trimming has significant impact on mapping results. But adapter trimming itself still has significant impact on mapping results because the worth results were obtained from mapping of raw reads: 81.06% (roughly 4% difference).

S. cerevisiae Illumina MiSeq data

For this library (2x250bp paired-end data with 3,875,453 paired reads): the best assembly was seen from data preprocessed with all stages: de-duplication, combining paired reads, quality and adapter trimming, see Table 6G from Supplementary. 3,966

reads were recognized and discarded as duplicates, 19.72% of total reads contained adapters, and the same amount of reads were combined into longer single-end sequences.

De-duplication and combining paired reads provided big impact on assembly speed and quality, providing shortest assembly time: 10 and 11hrs, in comparison to 32 and 47hrs in other tests. As in previous, de-duplication and combining paired reads decrease amount of data to assembly but still preserving enough for producing good quality results.

Quality trimming has also significant impact on assembly time and quality. N50 is 83,571bp against 80,595bp from data preprocessed with adapter trimming flag only. Assembly from data preprocessed with adapter and quality trimming flags also provided relatively low assembly time: 32hrs versus 47hrs in comparison to adapter trimming only.

For the raw data, assembler could not finish the task in reasonable time; thereby we had to stop ongoing assembly process.

The best mapping results were obtained from data preprocessed with all stages (de-duplication, merging overlapping paired reads, quality and adapter trimming), see Table 7G from Supplementary Materials. In this case there are 98.92% of mapped reads.

Without de-duplication (combining paired reads, quality and adapter trimming), there are 98.88% of reads properly mapped. Again, this may indicate that left duplicates had low quality thereby slightly decreasing mapping quality.

Without overlapping and de-duplication, mapping quality is still high in comparison to the two previous tests: 98.90% (which is slightly higher than from data preprocessed with merging overlapping reads option). This can indicate that even if we keep lower

quality duplicates in a library, we can still have good results using the advantage of paired-end paradigm, where one pair support another (using the insert size) even if that pair not properly mapped.

With adapter trimming only we have 98.32% reads properly mapped, which indicates that quality trimming has no such great impact on mapping quality (19.72% reads with adapters were found). Again, in this case we may observe advantage of paired-end method.

Mapping of raw reads indicates only 92.67% of mapped reads. This is significant difference in comparison to the previous test: 5.65%. Thereby, adapter trimming significantly affects mapping quality even if 19.72% of adapters were found.

Comparison of SeqyClean to other preprocessing tools

Among the best HTS data preprocessing tools, SeqyClean is the most comprehensive and precise, developed to serve most user's needs. We performed evaluation of two *E. coli* libraries: Roche 454 pyrosequence library and Illumina MiSeq library and one *S. cerevisiae* Illumina library in order to compare results with tools that work with single and cannot work with several technologies.

De-novo assembling of the E. coli pyrosequence library preprocessed with SeqyClean and other tools

According to the results (see Table 1.1), SeqyClean provides the largest N50 (111,928 bp). Assembly time is significantly lower in comparison to Lucy and AlienTrimmer (4 minutes versus 4 and 13 hours). This indicates that the untrimmed artifacts (adapters) left in a library, introduce additional complexity to the assembler. Assembly from data preprocessed with Btrim and AlienTrimmer have N50 of 97,590 bp

and 88,443 bp respectively, indicating second- and third-largest N50s. Among contig metrics, preprocessing with Btrim provides the smallest number of contigs (large/total) of 176 large/548 total and SeqyClean provides the second smallest number 362 large/1294 total.

Reads assembled from preprocessing with Lucy provide the third smallest number of contigs (549 large/1372 total). Preprocessing with Ea-utils and Btrim tools produced the closest total length to the reference (4,686,639 bp and 4,723,675 bp), and preprocessing with SeqyClean produced the fifth closest result (5,164,984 bp). Pre-processing with SeqyClean also produced the third smallest assembly time of 10 minutes, versus 4 (Btrim) and 6 minutes (Ea-utils). The assembly time for Ea-utils is lower in comparison to SeqyClean, but Ea-utils produced a very fragmented assembly (more than 3,000 contigs). SeqyClean outperforms the other preprocessing applications in terms of N50.

Table 1.1: Assembly comparison: SeqyClean against other applications. Test library - *E. coli* 454 pyrosequence library (*E.coli* K-12/MG1655 454 pyrosequence data, 621,578 reads, 327,471,374 bases).

Application	N50	Large (>1000bp)/Total Contigs	Total Length	Assembly Time (HH:MM:SS)	Reference length
Lucy	84,206	549/1372	5,148,220	04:00:01	4,641,652
AlienTrimmer	88,443	554/1365	5,150,261	13:00:03	4,641,652
Btrim	97,590	**176/548**	4,723,675	00:04:12	4,641,652
Ea-utils	2,214	2561/3439	**4,686,639**	00:06:20	4,641,652
SeqyClean	**111,928**	362/1294	5, 164,984	00:10:11	4,641,652

Table 1.2: Assembly comparison: SeqyClean against other applications. Test library - *E. coli* Illumina library (*E. coli* K-12/MG1655 Illumina data (SRR519926, a 2 x 250 bp run, sequenced on an MiSeq)).

Application	N50	Large (>1000bp)/Total Contigs	Total length	Assembly Time (HH:MM:SS)	Reference length
SeqyClean	**82,666**	**117/170**	4,555,814	00:04:23	4,641,652
AlienTrimmer	53,598	160/218	4,558,242	00:03:35	4,641,652
Ea-utils	59,473	131/183	4,556,869	00:20:21	4,641,652
Skewer	69,502	128/172	4,555,815	00:04:08	4,641,652
Trimmomatic	62,574	130/179	4,557,781	00:05:40	4,641,652
AdapterRemoval	62,382	134/180	4,556,466	00:04:21	4,641,652
Sickle	54,771	139/187	4,556,851	00:04:18	4,641,652

De-novo assembling of the E. coli paired-end library preprocessed with SeqyClean and other tools

SeqyClean provides the largest N50 (82,666 bp, see Table 1.2) and outperforms the other applications among other parameters: the number of contigs having lengths >=1,000bp,/total contigs (117/170); total length of large contigs (4,555,814 bp); total length of the assembly. Skewer produced the second-largest N50 (69,502 bp) and Trimmomatic the third (62,574 bp). Total assembly length is the same for all tools. Also, data preprocessed with SeqyClean produced the smallest number of large/total contigs (117/130), though it led to a relatively large assembly time (04:23 min). Base on these results we can assume that the SeqyClean is at least comparable to these tools in terms of adapter and quality trimming.

Mapping of E. coli pyrosequence and Illumina libraries preprocessed with SeqyClean and other tools

According to Table 1.3, the SeqyClean outperforms the other evaluated applications, except Btrim, providing the second-highest percentage (88.40%) of mapped reads. We assume this happens due to slightly better adapter removal with SSAHA algorithm, implemented in SeqyClean. All alignment tests ran quickly (from 0.5 to 1 min). The best results given by Btrim: 91.02%, Lucy provided 85.89% and AlienTrimmer provided 85.49%; and the worst, Ea-utils: 44.44%.

For Illumina libraries, according to results shown in Table 1.4, SeqyClean provides the best results among all statistics and outperforms other tools in all modes: paired-end reads (PE), unpaired single-end reads (SE) and combined pair- and single-end reads (PE+SE). For quality trimming we used the following parameters: MAE=0.01 (20 Phred),

MEE=0.01. Other parameters were left at their defaults. For other tools we used their recommended parameters. SeqyClean outperforms other tools in PE+SE (combined paired- and single-end reads) mode and provides 99.43% aligned sequences. Trimmomatic and Sickle provide only 92.83% and 92.70% of aligned reads. SeqyClean also slightly outperforms other tools in PE (paired-end) mode, having 99.91% aligned reads against 99.89% from Trimmomatic and 99.65% from Skewer. In SE (single-end reads) mode, 97.25% reads were aligned, versus 66.73% from Trimmomatic and 37.93% from Sickle.

This indicates better removal of TruSeq adapters in single- and, especially, in paired-end modes in comparison to other tools. A high number (more than 99%) of aligned paired-end reads that were preprocessed with other tools in comparison to low rate of alignment of single-end reads (66.73% for Trimmomatic, 37.93% for Sickle, other tools discard single-end reads) makes us to assume these tools do not perform comprehensive de-noising of single-end reads. SeqyClean performs de-noising both paired- and single-end reads (by ‘single-end’ we define a situation when one read has lost its pair due to low quality of bases of it). This potentially makes SeqyClean to be one of the leading applications for sequence preprocessing for subsequent analysis.

Table 1.3: Comparison of Bowtie2-mapping: SeqyClean against other applications. Test library - for E. coli Roche 454 pyrosequence library (*E. coli* K-12/MG1655 454 pyrosequence data, 621,578 reads, 327,471,374 bases).

Application	**Reads mapped/Reads total, %**
SeqyClean	88.40
AlienTrimmer	85.49
Lucy	85.89
Btrim	91.02
Ea-utils	44.44

Table 1.4: Comparison of Bowtie2-mapping: SeqyClean against other tools. Test library: *E. coli* Illumina library *E. coli* K-12/MG1655 Illumina data (SRR519926, a 2 x 250 bp run, sequenced on an MiSeq). PE – paired-end reads; SE – single-end reads; PE+SE – aligned combined PE and SE; PE – only PE, SE – only SE; NA – application does not report SE after cleaning.

Application	**PE+SE, %**	**PE, %**	**SE, %**
SeqyClean	**99.43**	**99.91**	**97.25**
AlienTrimmer	NA	99.44	NA
Ea-utils (fastq-mcf)	NA	99.57	NA
Skewer	NA	99.65	NA
Trimmomatic	92.83	99.89	66.73
AdapterRemoval	NA	99.13	NA
Sickle	92.70	99.30	37.93

De-novo assembling of the S. cerevisiae paired-end library preprocessed with SeqyClean and other tools

SeqyClean provides the largest N50 (85,803 bp), minimum number of large/total contigs (304/644) and lowest assemble time (10hrs, 14 min, 21 sec), see Table 1.5. The second best N50 comes from Skewer (83,993 bp) and the third best N50 from AlienTrimmer (80,620 bp). The total length of the assembly from data processed with SeqyClean is the fourth-largest (11,631,506 bp), where the largest (and most close to the reference) comes from Ea-utils (11,637,616 bp).

Mapping of the S. cerevisiae paired-end library preprocessed with SeqyClean and other tools

Mapping results also suggest that SeqyClean outperforms the other tools (Table 1.6). Data processed with SeqyClean shows the highest percentage of mapped paired-end reads (98.99%), single-end reads (98.53%) and second-highest percentage of combined paired- and single-end reads (98.91%). The highest percentage of mapped of combined paired- and single-end reads comes from data processed with Trimmomatic (98.93%).

These results also indicate that SeqyClean outperforms other applications.

Table 1.5. Assembly comparison: SeqyClean against other applications. Test library - *S. cerevisiae* Illumina paired-end data (a 2 x 250 bp run, sequenced on an MiSeq, 3,875,453 reads).

Application	N50, bp	Large (>1000bp)/Total Contigs	Total Length, bp	Assembly Time (HH:MM:SS)	Reference Length, bp
SeqyClean	**85,803**	**304/644**	11,631,506	**10:14:21**	12,154,788
AlienTrimmer	80,620	328/713	11,553,641	11:31:18	12,154,788
Ea-utils	59,473	323/732	11,637,616	36:58:00	12,154,788
Skewer	83,993	303/718	11,631,846	43:58:31	12,154,788
Trimmomatic	73,759	334/668	11,623,905	10:31:26	12,154,788
AdapterRemoval	74,435	323/760	11,552,722	40:14:47	12,154,788
Sickle	79,319	314/755	**11,639,914**	52:22:47	12,154,788

Table 1.6. Comparison of Bowtie2-mapping: SeqyClean against other tools. Test library: *S. cerevisiae* (a 2 x 250 bp run, sequenced on an MiSeq, 3,875,453 reads). PE – paired-end reads; SE – single-end reads; PE+SE – aligned combined PE and SE; PE – only PE, SE – only SE; NA – application does not report SE after processing.

Application	PE+SE, %	PE mapped, %	SE, %
SeqyClean	98.91	**98.99**	**98.53**
AlienTrimmer	NA	97.39	NA
Ea-utils (fastq-mcf)	NA	96.73	NA
Skewer	NA	98.51	NA
Trimmomatic	**98.93**	98.78	97.83
AdapterRemoval	NA	98.74	NA
Sickle	94.10	94.18.	87.13

Runtime and memory

Memory consumption directly depends on the size of reference genome and sequence library. Table 8G from Supplementary materials shows memory (RAM) footprints needed to store genomes of model organisms into a hash table. Using 2-bit DNA compression can significantly reduce memory consumption. In vector/contaminant trimming mode, the runtime of SeqyClean depends both on the reference and sample library size. In the future we consider applying the Burrows-Wheeler [23] transformation in order to further reduce memory. Runtime depends on the size of the NGS library and on typical machine (4-24 cores, 8-128 GB of RAM) speed is: 1,251 reads/sec (799 sec for 10^6 reads) to process typical pyrosequence library (~600,000 – 1,000,000 reads) and 3,623 reads/sec (276 sec for 10^6 reads) to process typical Illumina library (~10,000,000 (HiSeq) … ~200,000,000 (MiSeq) paired-end reads).

1.5 Conclusions

Sequence quality greatly affects on quality of analysis. With SeqyClean we preprocessed two different NGS libraries: *E. coli* (Roche 454, Illumina) and *S. cerevisiae* (Illumina MiSeq). We have shown that the quality of sequence data is greatly important for both genome assembly and genome mapping. In particular, adapter and quality trimming bring significant improvement of quality of assembly and mapping. As mapping quality increases false-positive INDELs and SNPs are reduced. Such false-positive variants can be mistakenly recognized as good, which leads to poor results. If the sequence library contains contaminants or vector sequence, it is necessary to perform vector and contaminant filtering in order to further improve the quality of analysis. Overall SeqyClean is a valuable improvement over existing preprocessing tools since it

provides the most complete sequence preprocessing solution, offering all sequence preprocessing aspects. We are constantly working on optimization of the algorithms used in SeqyClean.

1.6 Availability and requirements

- Project name: SeqyClean
- Project home page: http://bitbucket.org/izhbannikov/seqyclean
- Operating system(s): OS X, Linux
- Programming language: C++
- Other requirements: none
- License: GPL
- Any restrictions to use by non-academics: none

SeqyClean is available under GPL from the following link: http://bitbucket.org/izhbannikov/seqyclean and requires GNU C/C++ compiler installed. SeqyClean developed for OS X and Linux users only.

1.7 Acknowledgements

This publication was made possible by NIH Grant P20RR16454 from the INBRE Program of the National Center for Research Resources.

1.8 References

[1] Schatz MC, Delcher AL and Salzberg SL, Assembly of large genomes using second-generation sequencing, Genome Res, 2010 20: 1165-1173.

[2] Criscuolo, A., & Brisse, S. (2013). Alientrimmer: A tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. Genomics.

[3] Shizuya, H., Birren, B., Kim, U. J., Mancino, V., Slepak, T., Tachiiri, Y., & Simon, M. (1992). Cloning and stable maintenance of 300-kilobase-pair fragments of human DNA in Escherichia coli using an f-factor-based vector. Proceedings of the National Academy of Sciences, 89(18), 8794-8797.

[4] Kim, U. -J., Shizuya, H., Sainz, J., Garnes, J., Pulst, S. M., de Jong, P., & Simon, M. I. (1995). Construction and utility of a human chromosome 22-specific fosmid library. Genetic Analysis: Biomolecular Engineering, 12(2), 81 - 84.

[5] Magoč T, Salzberg SL, FLASH: fast length adjustment of short reads to improve genome assemblies, Bioinformatics. 2011 Nov 1; 27(21): 2957-63.

[6] 454 Life Science, A. R. C. (2008). Point-and-click tools for assembly, mapping and amplicon variant analysis, http://454.com/products/analysis-software/index.asp.

[7] TrimEST, http://emboss.bioinformatics.nl/cgi-bin/emboss/help/trimest.

[8] VectorStrip, http://emboss.bioinformatics.nl/cgi-bin/emboss/help/vectorstrip.

[9] VecScreen, http://www.ncbi.nlm.nih.gov/tools/vecscreen.

[10] Chou, H., & Holmes, M. H. (2001). DNA sequence quality trimming and vector removal. Bioinformatics/computer Applications in The Biosciences, 17, 1093–1104.

[11] Falgueras, J., Lara, A., Pozo, N. F., Canton, F., Trabado, G. P., & Claros, M. G. (2010, January 20). SeqTrim: a high-throughput pipeline for pre-processing any type of sequence read. BMC Bioinformatics, 11(1), 38.

[12] Erik Aronesty (2013), Comparison of Sequencing Utility Programs.

[13] Lindgreen S, AdapterRemoval: easy cleaning of next-generation sequencing reads, BMC Res Notes. 2012.

[14] Hongshan Jiang, Rong Lei, Shou-Wei Ding and Shuifang Zhu, Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads, BMC Bioinformatics.2014, 15:182.

[15] Bolger, A. M., Lohse, M., & Usadel, B. (2014). Trimmomatic: A flexible trimmer for Illumina Sequence Data. *Bioinformatics*.

[16] Kong, Y (2011) Btrim: A fast, lightweight adapter and quality trimming program for next-generation sequencing technologies, Genomics, 98, 152-153.

[17] Marcel, M (2011), Cutadapt removes adapter sequences from high-throughput sequencing reads, Embnet journal, 12(1); 10-12.

[18] Joshi NA, Fass JN. (2011). Sickle: A sliding-window, adaptive, quality-based trimming tool for FASTQ files (Version 1.29). Available at https://github.com/najoshi/sickle.

[19] Fickett, J. W. (1984). Fast optimal alignment. Nucleic Acids Research, 12, 175–179.

[20] Ning Z, Cox AJ and Mullikin JC, SSAHA: a fast search method for large DNA databases, Genome research 2001; 11; 10; 1725-9.

[21] Langmead, B., & Salzberg, S. L. (2012). Fast gapped-read alignment with bowtie 2. Nat Meth, 9(4).

[22] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Subgroup. G. P. D. P. (2009). The sequence alignment/map (SAM) format and SAMtools. Bioinformatics.

[23] Li H. and Durbin R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics, 25:1754-60.

[24] Gurevich, A., Saveliev, V., Vyahhi, N., & Tesler, G. (2013). Quast: quality assessment tool for genome assemblies. Bioinformatics, 29(8), 1072-1075.

[25] Huse SM, Huber JA, Morrison HG, Sogin ML, Welch DM, Accuracy and quality of massively parallel DNA pyrosequencing, Genome Biol. 2007;8(7): R143.

[26] Ralser, M., Kuhl, H., Ralser, M., Werber, M., Lehrach, H., Breitenbach, M., & Timmermann, B. (2012). The saccharomyces cerevisiae W303-k6001 cross-platform genome sequence: insights into ancestry and physiology of a laboratory mutt. *Open Biology*, *2*(8).

[27] FASTX-Toolkit, http://hannonlab.cshl.edu/fastx_toolkit/.

[28] Using a PhiX Control for HiSeq Sequencing Runs, http://res.illumina.com/documents/products/technotes/technote_phixcontrolv3.pdf

[29] Miller, JR; Koren, S; Sutton, G (2010). "Assembly algorithms for next-generation sequencing data". Genomics 95 (6): 315–327.

Chapter 2

SLOPMAP: A SOFTWARE APPLICATION TOOL FOR QUICK AND FLEXIBLE IDENTIFICATION OF SIMILAR SEQUENCES USING EXACT K-MER MATCHING

Chapter 2 presents SlopMap, an application, written in C++ with the main purpose of searching for the reads originating from some genomic region of interest, such as single gene or a plasmid. SlopMap is user-friendly, command-line application tool, which is able to work with Illumina, Roche 454 and Ion Torrent sequence libraries. SlopMap is open source and available at: http://bitbucket.org/izhbannikov/slopmap.

This chapter was published in a Special Issue on "Bioinformatics for High-throughput Sequencing" of the Journal of Data Mining in Genomics & Proteomics as *SlopMap: a software application tool for quick and flexible identification of similar sequences using exact k-mer matching*, Ilya Y. Zhbannikov, Samuel S. Hunter, Matthew L. Settles, and James A. Foster, 2013.

2.1 Abstract

With the advent of Next-Generation (NG) sequencing, it has become possible to sequence an entire genome quickly and inexpensively. However, in some experiments one only needs to extract and assemble a portion of the sequence reads, for example when performing transcriptome studies, sequencing mitochondrial genomes, or characterizing exomes. With the raw DNA-library of a complete genome it would appear to be a trivial problem to identify reads of interest. But it is not always easy to incorporate well-known tools such as BLAST, BLAT, Bowtie, and SOAP directly into a bioinformatics pipelines before the assembly stage, either due to incompatibility with the assembler's file inputs, or because it is desirable to incorporate information that must be

extracted separately. For example, in order to incorporate flowgrams from a Roche 454 sequencer into the Newbler assembler it is necessary to first extract them from the original SFF files.

We present SlopMap, a bioinformatics software utility, which allows rapid identification similar to provided target sequences from either Roche 454 or Illumnia DNA library. With a simple and intuitive command-line interface along with file output formats compatible with assembly programs, SlopMap can be directly embedded in biological data processing pipeline without any additional programming work. In addition, SlopMap preserves flowgram information needed for Roche 454 assembler.

2.2 Introduction

New methodologies enabled by Next Generation Sequencing (NGS) that are of particular interest to us include transcriptome analysis for RNA research [4] and mitochondrial sequencing from exome data [3]. Such applications include those in which the researcher is interested in assembling only specific content within a genome of interest, using a set of targets to initialize the assembly process. It may seem trivial to identify the reads of interest among those produced by NGS hardware, using well-known general-purpose alignment or mapping tools such as Blat [7], Bowtie2 [8], BWA [9], and SOAP [5]. But even an efficient tool may be difficult to incorporate directly into a bioinformatics pipeline before the assembly stage, since it may be necessary to convert data to a different file format. For example, existing mappers usually use the SAM/BAM [6] file format as output. None use SFF format files [1] as both input and output, and none but Bowtie2 support FASTQ output, and it is supported only in a limited sense.

Moreover, it is difficult to use existing mapping software tools when it necessary to establish a similarity threshold, i.e. when one wants reads that are 50%, 70% or 85% similar to the target (Figure 2.1). Relying only on input parameters such as gap penalties and seed size, which most well-known aligners have, it is difficult to achieve flexible mappings with require percentage of similarity. On the other hand, it often desirable to find reads that are at least 90% similar to the provided target, and to discard the rest.

Another problem arises if there is insufficient data on the edge of the target located within a reference genome (Figure 2.1). In this situation the whole read (marked red) can potentially be discarded due to lack of data on the edge, even if a part of the read has significant similarity to the target.

We present SlopMap, a bioinformatic software utility that quickly and flexibly identifies sequence reads that are within a given percent similarity to a target sequence. SlopMap is not a sequence alignment mapper, but rather identifies reads, which may have been derived from the target region. Unlike traditional alignment software, a SlopMap only report reads that are similar to the provided target. SlopMap selects reads for downstream analysis, such as assembly of sub-genome targets i.e. bacterial plasmids, virae, mitochondria, exome capture data, chloroplasts, transcriptomes, etc. It employs exact k-mer matching, which we call sloppy mapping, without conducting the computationally expensive alignment stage of traditional mappers.

SlopMap can be directly embedded in biological data processing pipelines before an assembly stage, since it maintains file format and preserves the original information such as bases, quality scores, and flowgrams (in the case of SFF files). SlopMap accepts both SFF (Roche 454 or Life Sciences Ion Torrent/Proton) and FASTQ (Illumina) file

for- mats. SlopMap is a simple, easy to use and robust tool that can be used with percent similarity to targets as low as 5% (95% dissimilar).

SlopMap along with its user manual is freely available under GPL from Bitbucket: http://bitbucket.org/izhbannikov/slopmap.

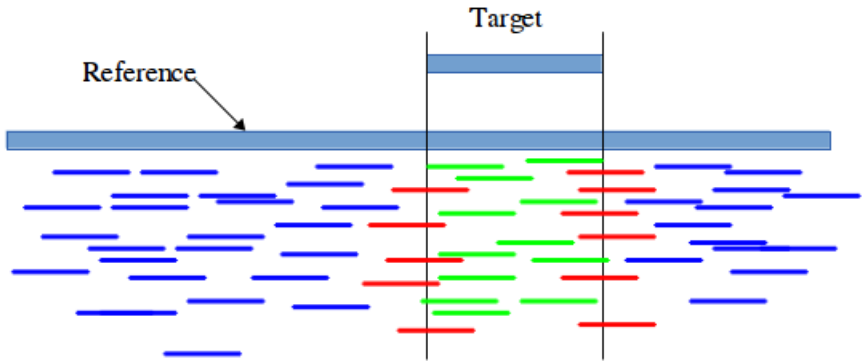

Figure 2.1: Situation when the general-purpose mapper can discard red reads on the edge of the target. Green reads can still be reported.

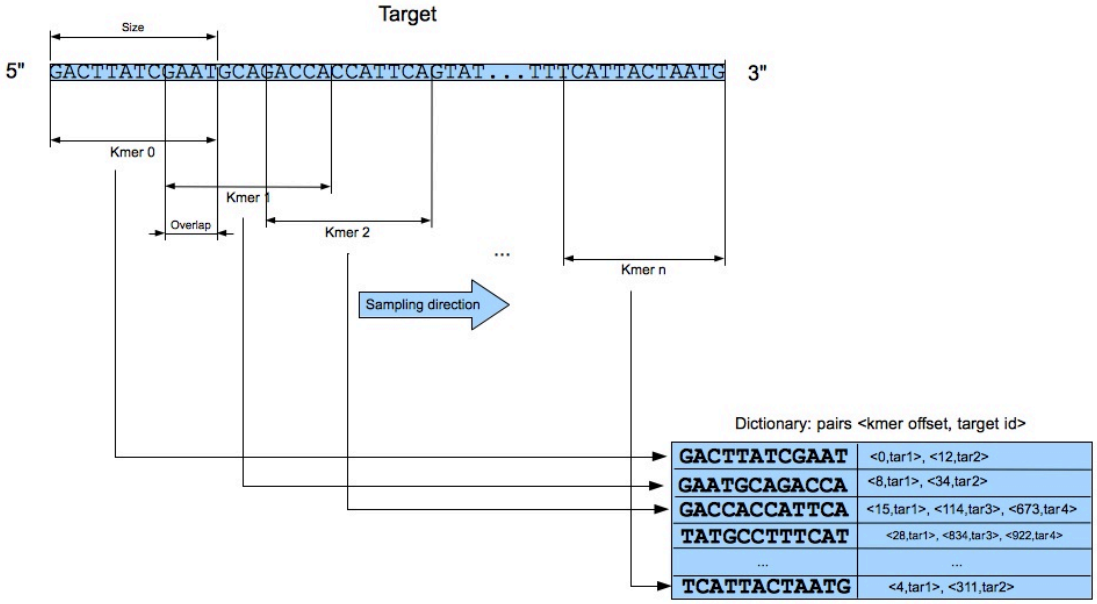

Figure 2.2: Sampling the target with constant pre-defined k-mer size *k* and a distance *d* between two consecutive k-mers.

2.3 Method

SlopMap is fully implemented and optimized in C++ for efficiency. This is a command line application with all the input parameters specified on the command line. SlopMap is tunable via input parameters for k-mer size *k*, percentage of similarity *t* and distance between two consecutive k-mers *d*. It also supports flexible input and output file formats: FASTQ, FASTA, SFF and TXT.

Input Files

Target

The target library file, also known as a "database", is a FASTA formatted file that contains one or multiple records. Each record consists of two parts: a header and a sequence string. The header must contain a name, which is a unique identifier of the record. The sequence string is DNA sequence which specifies the target of interest.

DNA Query Library Files

The DNA query library files are data from the NGS machines. SlopMap can take either Roche 454 SFF or FASTQ formatted files, or Illumina paired- or single-end reads. SlopMap computes the similarity of each read in a query library file to record in a target library.

Search procedure

SlopMap employs a multi-k-mer search approach with single-base overlap (that can be increased) between two consecutive k-mers in order to quickly determine the similarity to the target record. The algorithm is simple and straightforward:

(1) Compute a dictionary from the given target library.

(2) For each query sequence ("read") in the DNA query library:

• Compute a set of consecutive k-mers.

• Calculate the read's similarity to the set of target sequences.

(3) Output summary statistics and files.

Dictionary

The first step is to build a dictionary indexed with k-mers by sampling the given target (Figure 2.2). The k-mers are short (usually 9-15 bases long) substrings representing the contiguous target. The target library sequences are sampled with the following pre-defined parameters: k-mer size and a distance representing the constant overlap between consecutive k-mers. These parameters remains unchanged throughout program execution. All k-mers are hashed and associated with the offset position in the target string and a target record id. By default SlopMap uses k-mer size of 15 bases. Google Dense Hash Map [2] that allows far fast data retrieval and memory efficiency is used as a data structure for the k-mer dictionary.

K-mer matching

Query strings are sequentially sampled, so each query string contained within a given DNA query library is handled individually. K-mer size and a distance between two consecutive k-mers remains the same for all reads in the query library. For each read taken from the query library, a dictionary search is performed and similarity between the read and the target is calculated as follows:

$$S = \frac{number\ of\ shared\ bases}{Min\_length(query\ string, reference\ string)}, 0 \le S \le 1 \qquad (1)$$

Where number shared bases is the total length of k-mers shared between the target and query string, Min length(query string, target string) is the minimum length, either of the query or target. Those reads that meet the pre-specified similarity threshold

(parameter "−*t*" in SlopMap, by default it is set to 0.75) are then saved in output files, others are ignored. The values of *S* range from 40, having no similar k-mers to the target, to 1, having all k-mers shared between the read and target.

Output files

The output files are:

- A report file that contains all information about the reads, including similarity, positions of first and last match within the target, bases and quality scores.
- One or two FASTQ formatted file(s), depending on whether the data are from single- or paired-end library.
- Optionally, an SFF formatted file, which contains only those Roche 454 sequences similar to the provided target.

2.4 Validation

To validate SlopMap we compared it to several alternative DNA mapping tools: Bowtie2, BWA, Blat, on two different query DNA libraries: 621,578 Roche 454 *Esheria coli* K12 W583 reads; and 3,875,453 Illumina Yeast *Saccharomyces cerevisiae* W303-K6001 reads. For the target sets, we randomly chose ten genes with various lengths from both genomes, each of which has over 4000 genes: thrA, thiQ, cydD, ycgB, dhaR, alkA, yfgF, yphE, mscS, parC (E-coli, GenBank accession number: U00096.2); YNL095C, YNL094W, YNL093W, YNL092W, YNL091W, YNL090W, YNL089C, YNL088W, YNL087W, YNL085W (Yeast, GenBank accession number: AF458977.1).

We estimate the number of reads found by SlopMap with various thresholds and k-mers in order to:

- Estimate the effect of threshold to the number of reads found by SlopMap.

- Suggest values of *t* and *k* for optimal read recovery.
- Estimate the effect of various values of *d* (distance between two consequtive kmers) to the number of reads found by SlopMap against various threshold values allowing us to determine the range of optimal values for *d*.
- Compare the number of reads found by SlopMap to the number of reads found by other tools. In particular, to answer the question: what are the threshold values where read sets found by other tools are still subsets of reads found by SlopMap.

All tests were performed on a Linux server with Dual-Core AMD Opteron 8216 2.4 GHz processors (32 processors total) and 1 TB of shared memory and a laptop with single Intel Core i3 processor (four cores) with 4GB of memory.

2.5 Results and discussion

We calculated the number of found reads using various threshold values and k-mer sizes and compared our results to existing read mappers. These results are presented in Figure 2.3, which shows the number of reads against various distances between two consecutive k-mers. Further comparisons are made to compare the overlap between recovered read sets. These results are presented as Venn diagrams in Figures 2.5 and 2.6. We also provided recommendation for optimal values of parameters *k* and *d*.

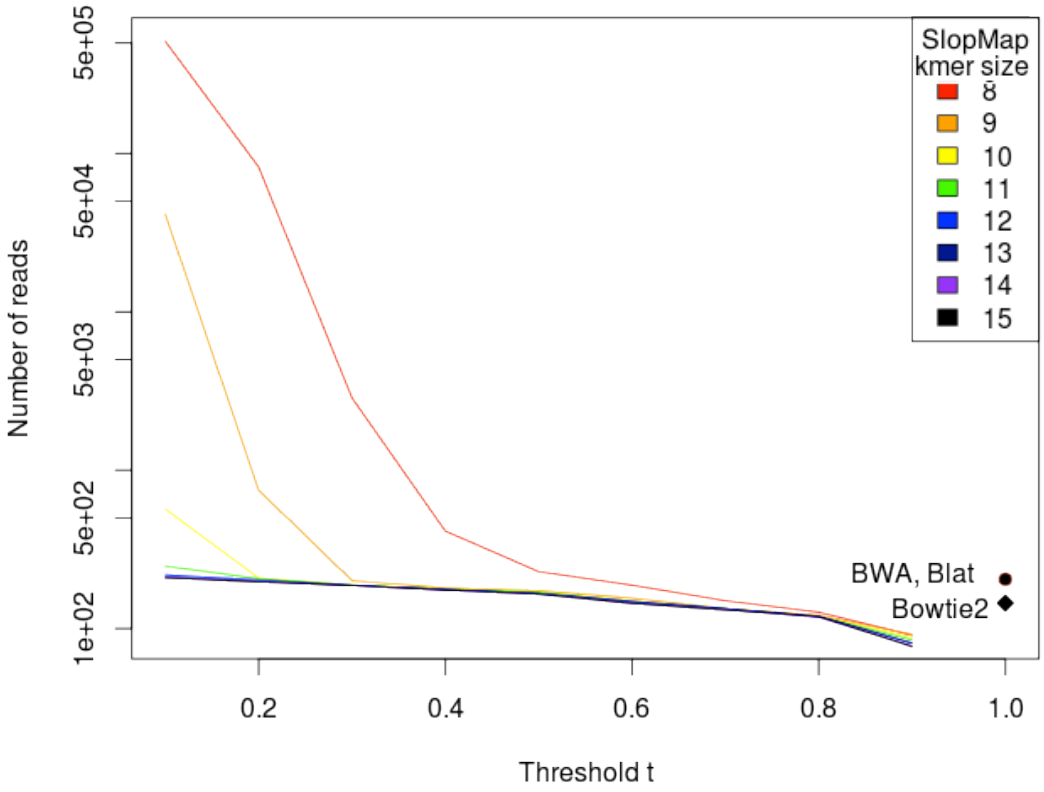

Figure 2.3: Number of reads found by SlopMap versus various percent identity threshold values t and different k-mer lengths *k*. At the left side of the plot there is large amount of reads found for $k \leq 9$ and $t \leq 0.2$ showing that these parameter values may result in many false-positives. Larger values of *k* and t can be used to be more selective in read recruitment, or in situations where the reference is highly similar to the sequenced reads. Roche 454 *E. coli* K12 W583 DNA library (621,578 reads) was used as a query library for this test.

Figure 2.3 shows the number of reads found by SlopMap using different kmer sizes and threshold values. From this plot its easy to see that for kmer values *k*, of 7, 8, 9 along with threshold value less than 0.1 (10% of similarity), large amount of reads were found. These values of *k* should be used only in such situations where the reference sequence is very divergent from the sequenced sample. Low values of *t* and *k* result in high sensitivity at the expense of specificity and should be used carefully to avoid multiple false positive hits.

Values of *k* within the range 10...15 can be used to generate more specific matches and are recommended for general usage. Higher values of *k* will result in less false positive mapping, SlopMap will not match k-mers with mismatches, and will fail match reads at higher values of *k*. This is especially true in situations where reads have errors or adapters, which will generate false k-mers and where there are real variants between the target and sequenced sample.

Number of reads found by SlopMap using various threshold values and distances between two consecutive k-mers

In order to examine how different values of parameter d (the distance between two consecutive k-mers in a read) impacts to the number of reads found by SlopMap and propose the optimal value for *d*, we provided a set of tests with k-mer size $k = 11$ and threshold values $t = 0.1...1.0$ (i.e. from 10% to 100% similarity). This is shown in Figure 4.2. We find that in this data set, *d* has minimal impact on read recruitment. However we observe a higher recruitment rate for lower *d*, suggesting higher sensitivity. Using a small *d* results in slower performance however, so in cases where target and reference are highly divergent, a low d should be used, while a higher *d* can be used for more similar

sequences. With these considerations in mind, we have set the default value of d to 3, and allow the user to change it using the command-line parameter $d = N$. Our recommendations for the parameter d value to be no more than k-mer length. Otherwise there may not be sufficient coverage. The values for the d from 1 to 5 are optimal for k = 11 bases, since they give number of reads significantly higher than other tools within $t \leq 0.5$ ($t = 0.75$ is set by default in SlopMap).

Sensitivity test: comparison the number of reads reported by SlopMap and other tools

Gaps occur when part of the query aligns to one part of the reference and another part aligns close to the first part but with a gap of one or more bases. Such gaps are usually well recognized by some widely used aligners. Another type of gap can occur at the end of a target sequence, when part of the query matches the target, resulting in an end gap. SlopMap can find such reads and thereby identify more similar sequences than some other alignment tools. In order to compare the sensitivity of SlopMap to a set of other mappers (BWA, Blat, Bowtie2), we conducted several tests using Roche 454 reads from *E. coli* mapped against ycgB gene sequence. We are interested what is the cut-off point when reads found by alignment tools are still subsets of reads found by SlopMap. We can roughly say that the set of reads reported by one application is a subset of reads reported by another application if there is more 95% overlap between these two sets. For two k-mer lengths (10 and 15 bases) and threshold values (0.1...1 with step of 0.1) we computed Venn diagrams that show overlap sets reported by SlopMap and other tools. From these diagrams we conclude that for threshold values below 0.3 (30% similarity) and for both k-mer sizes (10 and 15), the reads found by Blat is a subset of number of

reads found by SlopMap. Threshold values when reads reported by Bowtie 2 and BWA are still subsets of number of reads reported by SlopMap are 0.7 and 0.3 (70% and 30% similarity) for k-mer size 10 and 15 respectively. Results are shown in Figures 2.5 and 2.6.

Non-consecutive matches

When we compute the similarity of reads we do not assume that k-mer matches are consecutive. Non-consecutive matching may occur in situations, which are biologically possible such as exon shuffling, inversion, etc. In Figure 2.7, the read and a target are shown along with k-mers shared between them. In this situation, k-mer 1, 2 and 3 match corresponding k-mers in a target but in different order (non-consecutive). The read can be still considered as similar to the target. SlopMap identifies and reports this read as being similar to the target, despite the rearrangement.

Timing considerations

Figure 2.8 displays the execution time required to complete each search. We compare execution times for various threshold values of SlopMap (other parameters were set to default) with other tools.

SlopMap is faster then Bowtie2 and BWA, but slower than Blat, which is the fastest of the mappers we tested. However, Blat requires that the input be in FASTA format, and does not support writing output in FASTA or FASTQ format, making it necessary to perform additional steps, both before and after using the program. Post-mapping conversion work is also required for BWA and Bowtie2. Bowtie2 writes all sequences to the SAM file, and write unmapped or discordantly mapped reads to files using command line parameters to output mapped sequences in FASTQ file format.

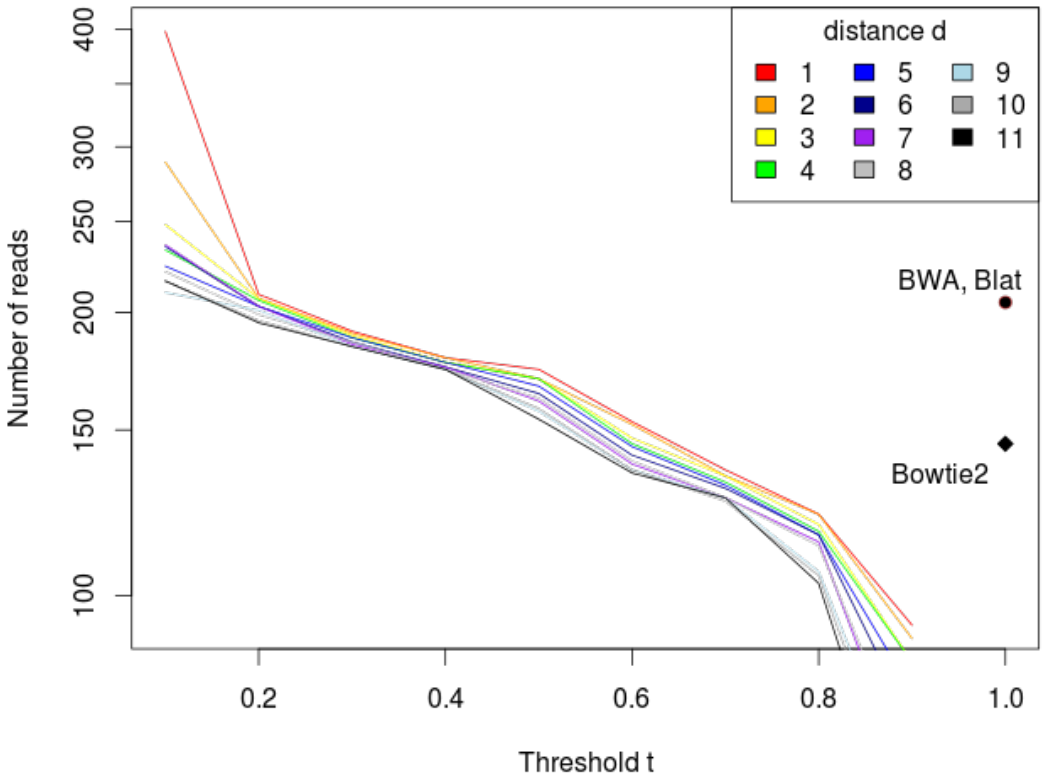

Figure 2.4: Number of reads found by SlopMap against various threshold values and different distances between two consecutive kmers. Roche 454 *E. coli* K12 W583 DNA library (621,578 reads) mapped against a single bacterial gene ycgB(1583 bp) was used as a query library for this test. When threshold value *t* is 100%, SlopMap does not find any reads similar to the target sequence. This is expected because reads can contain base-call errors, homopolymers and other artefacts that introduce noise into sequences.

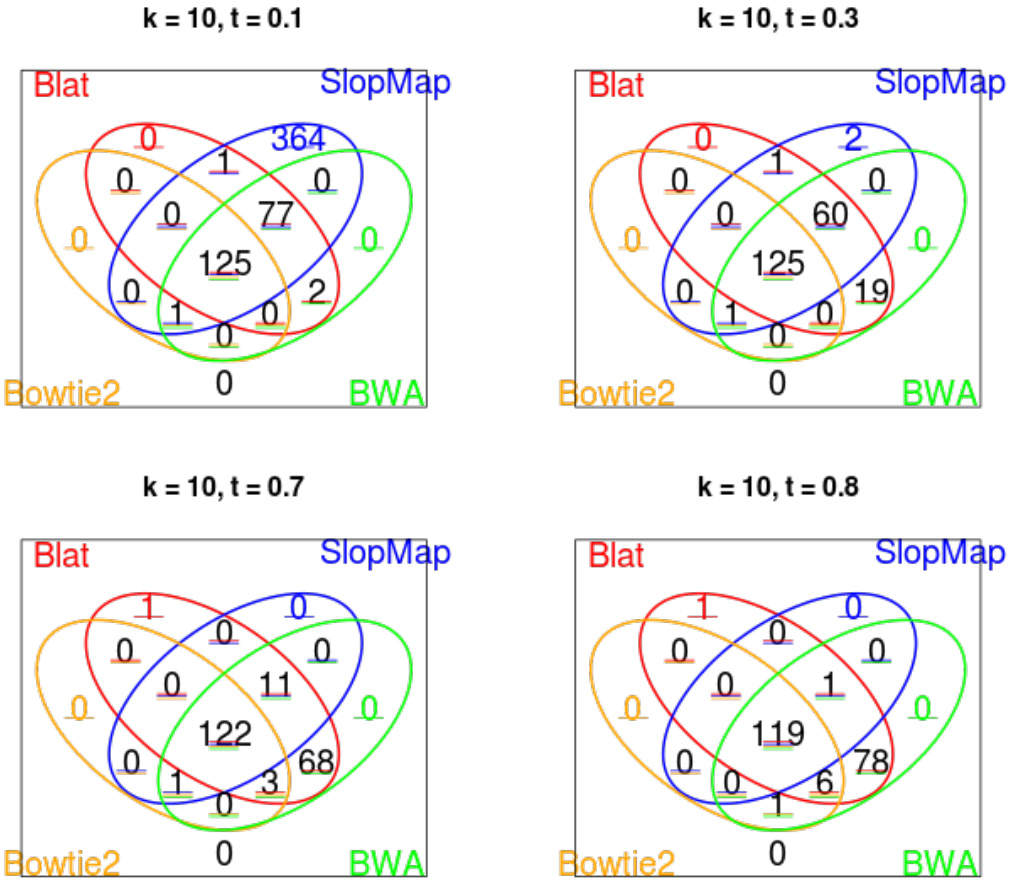

Figure 2.5: Number of reads reported by SlopMap and other tools depending on threshold values t and kmer sizes k = 10 bases. When t = 0.3 (the top-right figure), only 19 reads reported by Blat do not overlap with SlopMap, and those are also reported by BWA; reads reported by BWA and Bowtie2 are subsets of reads reported by SlopMap until $t \geq 0.7$. These tests were conducted on Roche 454 *E. coli* K12 W583 DNA library used as a query set.

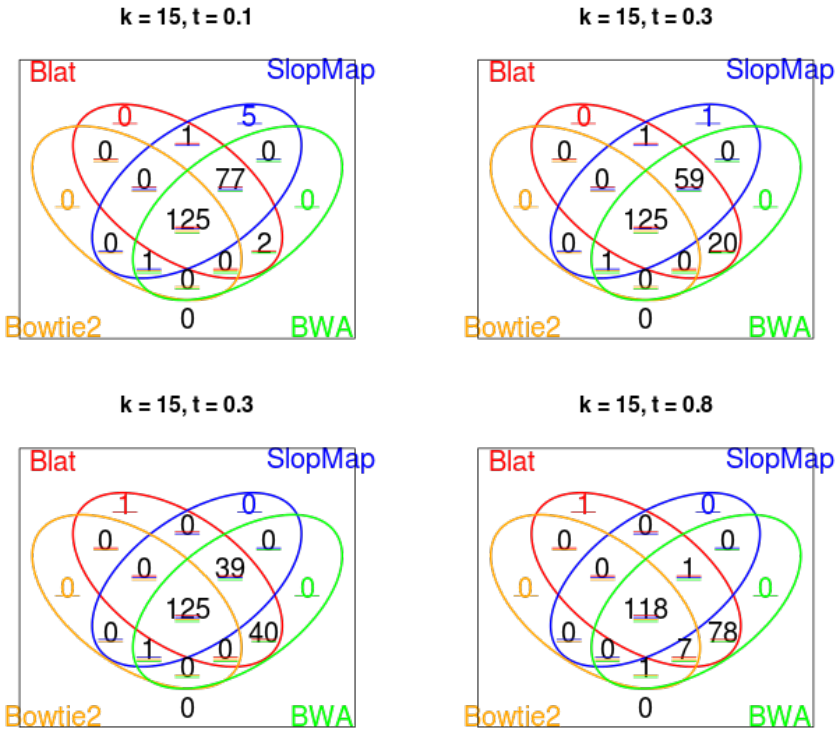

Figure 2.6: Number of reads reported by SlopMap and other tools depending on percent similarity threshold values t and k-mer size k = 15 bases.

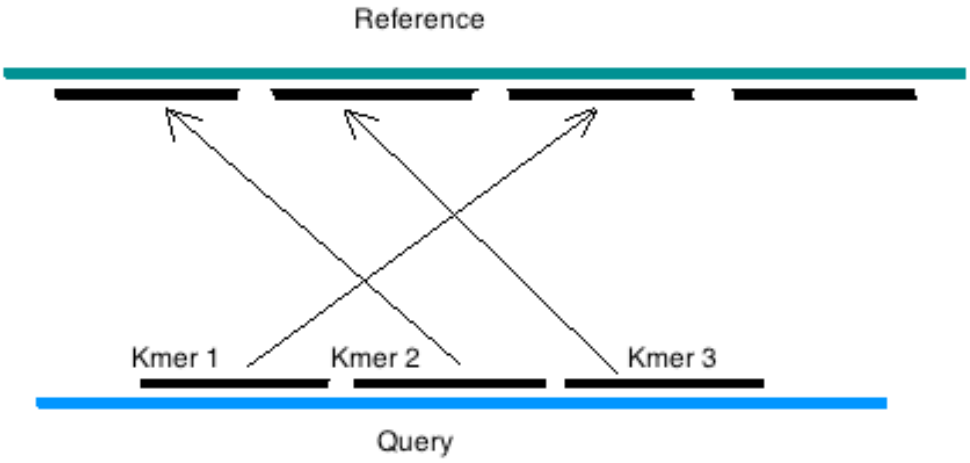

Figure 2.7: Non-consecutive k-mer matching is still when using a target and query where rearrangements may have occurred. SlopMap identifies and reports these matches.

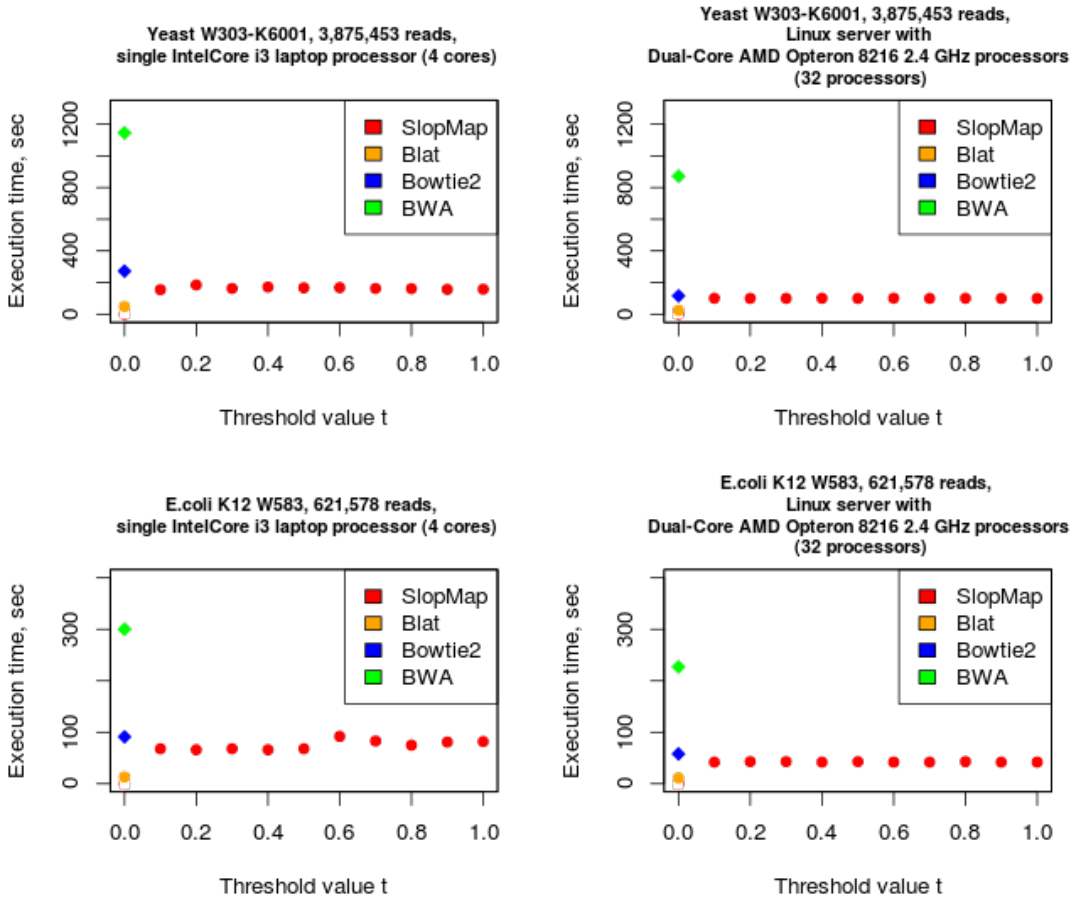

Figure 2.8: Search duration: comparison of execution time between search tools. SlopMap is consistently fast across a range of threshold values (*t*), and perform somewhat faster than Bowtie2, and significantly faster than BWA while using the same amount of CPU resources.

Memory requirements

SlopMap is fast and requires very little memory (2-200MB, depending on target size). The memory consumption of SlopMap during searching grows linearly with the number of sequences and also depends on the k-mer size defined by the user. For example, when the *E. coli* data set containing sequences with a mean length of 450 characters was indexed on 15-mers, 50 kB of memory was utilized for every 20,000 bases.

2.6 Conclusion and future work

SlopMap provides researchers with a high-throughput choice for searching large sets of reads against target sequences. The software presented is faster than some well-known aligners, sensitive to low-similarity matches when desired, and flexible enough to allow similarity comparison for DNA (and potentially RNA and proteins). SlopMap is specifically designed for matching queries against large (more than 500,000 sequences) query sets. Three of SlopMaps beneficial attributes are its speed, flexibility and ease of use. Despite being fast and efficient mapper, we plan to further improve SlopMap by adding support for multicore execution and by exploring more space and time efficient methods for storing and looking up kmers.

We believe that the biological research community will benefit from using SlopMap.

2.7 Acknowledgments

This publication was made possible by NIH Grant P20 RR16454 from the INBRE Program of the National Center for Research Resources.

2.8 References

[1] 454 Life Sciences at al. Standard flowgram format (SFF).

[2] Craig Silverstein at al., Sparsehash: An extremely memory-efficient hash map implementation, 2012.

[3] Guo Y at al., Mitoseek: extracting mitochondria information and perform- ing high-throughput mitochondria sequencing analysis. Oxford Journal of Bioinformatics (Application Note), March 2013.

[4] Huang Lulin at al., The first illumina-based de novo transcriptome sequencing and analysis of safflower flowers. PLOS ONE, 7(6), June 2012.

[5] Li R at al., Soap2: an improved ultrafast tool for short read alignment. Oxford Journal of Bioinformatics, 2009.

[6] The SAM Format Specification Working Group, The sam format specification (v1.4-r985), September 2011.

[7] W James Kent, Blat - the blast-like alignment tool, Genome Research, 12(4):656664, 2002.

[8] Salzberg S. Langmead B., Fast gapped-read alignment with Bowtie 2, Nature Methods, 9:357–359, 2012.

[9] Durbin R Li H., Fast and accurate short read alignment with Burrows-Wheeler transform. Oxford Journal of Bioinformatics, 25:1754–60, 2009.

Chapter 3

ANALYSIS OF HIGH-THROUGHPUT MICROBIAL AMPLICON SEQUENCE DATA USING MULTIPLE MARKERS

Chapter 3 presents a meta-amplicon analysis algorithm developed for multi-marker analysis of 16S amplicon data. This allows researchers to better characterize microbial community composition using multiple genomic markers. The corresponding application, MetAmp is open source and available at http://izhbannikov.github.io/MetAmp.

The manuscript titled "Analysis of High-Throughput Microbial Amplicon Sequence Data Using Multiple Markers" by Ilya Y. Zhbannikov, Janet E. Williams and James A. Foster is currently being prepared for publication.

3.1 Abstract

16S gene amplicon sequencing is widely used in microbial studies to infer community diversity. However, the drawback of this is reduced resolution down to genera or even family level, rather than the species or strain level, due to using PCR primers that are imperfect and unable to produce enough product, and sequence errors. We present a novel approach and corresponding application, MetAmp, to combine amplicon data from multiple genomic markers for clustering next generation sequences into Operational Taxonomic Units (OTUs) for microbial community analysis. This proposed approach calibrates the reads using data from known reference microbial genomes. This represents potential improvements on current amplicon clustering methods used to characterize bacterial community composition and structure. We tested the accuracy of our proposed algorithm using both simulated and real 16S datasets, and compared results to the best available tools, namely UPARSE, QIIME and Mothur.

Results suggest that MetAmp outperforms other tools and opens new possibilities for multi-marker data analysis of 16S empirical amplicon data. Meta-amplicon analyses works best with at least three markers, and is applicable to non-bacterial analyses and to non-16S markers. However, our application and testing have been limited to 16S analysis of microbial communities.

Software, documentation and test datasets are available via the following link: http://izhbannikov.github.io/MetAmp

Contact: ilyaz@uidaho.edu, foster@uidaho.edu, janetw@uidaho.edu

3.2 Introduction

It has become standard practice to characterize microbial community composition and structure with cultivation-independent, high-throughput sequencing of all microbial DNA in a sample. Techniques for analyzing large sequence datasets vary with the type of data and the questions under investigation. In general, the goal is either to approximate entire genomes or gene products, or to characterize phylogenetic diversity or ecological structure. Metagenomic sequencing and analysis addresses the first class of problems. The most common technique for answering the second class of problems is to sequence specific marker regions of the DNA, producing sequences known as amplicons (or fingerprints). Different markers have different, often unknown, biases and limitations. Thereby cluster differently [3]. We hypothesize that it would be worthwhile to be able to combine amplicon sequences from multiple marker regions, reducing the risk of choosing the "wrong" marker for a specific question. However, no method currently exists for combining amplicons from multiple markers, preparatory to downstream analysis. In this

paper, we present a new methodology, “Meta-Amplicon analysis”, and a program, MetAmp, to do just that.

Ideal marker regions are highly conserved and vertically transmitted, so that that all microbes of interest are likely to have homologous sequences. However, ideal markers should also contain regions with sufficient variability to resolve relevant comparative phylogenetic or taxonomic questions. The most commonly used markers have been, and continue to be, so-called hypervariable regions of the 16S rRNA gene, a deeply conserved housekeeping gene with loops in the secondary structure that provide variability. Amplicons form the raw data for downstream bioinformatics analyses, such as rarefaction analysis, using such software as SEED [15], the R-package ‘vegan’ [8], or QIIME [13]. We focus on the analysis of 16S rRNA amplicon data, though our approach can be potentially useful for any amplicon data from any homologous region with sufficient variability.

There are two current approaches to amplicon analysis of microbial communities: phylogenetic- and OTU-based. The phylogenetic approach uses marker sequences to identify taxonomies for each amplicon, using maximum likelihood estimation, most commonly with a Naive Bayes classifier, or by BLAST searches against existing microbial databases such as SILVA, RDB, or GenBank [4,5]. This has the advantage of characterizing the likely evolutionary relationships between populations in each community and, potentially, community functions [14]. But it relies on existing sequence data, which is biased by historical emphasis on populations that can be cultivated or that are relevant to human activities. These are serious limitations, since most populations are still unknown and unsequenced, and often highly diverged from known species. The

taxonomic resolution is also often at the genera/family level [6], which is insufficient for many important questions.

The OTU approach clusters marker sequences by similarity and assigns all sequences that cluster together to a single operational taxonomic unit (OTU) [3, 7], after first cleaning the data by denoising and removing adapters/barcodes, contaminants, and chimeric sequences. The OTU approach is limited by the fact that amplicon sequence similarity is often not a good proxy for population similarity, and by potentially poor congruence between OTUs and taxonomies. OTU-clustering is highly dependent on the clustering algorithm chosen, the metric for sequence similarity and how well it recapitulates evolutionary relationships between populations, and the threshold that defines the operational taxonomic unit (usually 97% sequence identity). Despite these limitations, the OTU approach to amplicon analysis is very common in practice, largely because it can handle unknown and highly diverged populations. MetAmp is tailored to OTU analysis.

It is important to choose the appropriate marker regions for amplicon analysis, whether phylogenetic- or OTU-based. Due to processes such as rate heterogeneity, different marker regions evolve at different rates relative to each other and between species. Moreover, amplicons from some markers may reflect phylogenetic differences in a sample only at the genus, and sometimes the family level [3]. Even perfectly designed primers can miss the majority of species within the sample [16]. Since some important phenotypes, such as virulence, may vary at the strain level, this is a serious limitation.

Most studies use only a single marker region for amplicon based community analysis. The best currently available software tools for amplicon OTU analysis, such as

UPARSE [18], CD-HIT [19], QIIME [13], Mothur [14], assume amplicon are from a single marker, and so cannot analyze data from multiple markers. Consequently, the choice of marker region compels downstream analysis to suffer from the disadvantages of that choice, which we have described above.

We propose a novel amplicon analysis approach that uses known complete genes that include multiple markers to build a reference space in which it is possible to combine amplicons from different markers. Once one can combine amplicons from different markers into a common space, where they are all comparable, one can cluster sequences into OTUs, thereby combining the advantages of some markers in order to overcome the weakness in others. This, in turn, increases taxonomic resolution, which leads to better classifications and assignments.

We call this approach "meta-amplicon analysis", by analogy with metagenomic analysis. We implemented this approach in the MetAmp software package [23], which is freely available. The unique, primary contribution of MetAmp is its ability to combine multiple marker amplicons prior to clustering. MetAmp makes clustering possible, but it not itself a clustering tool. This paper describes the approach, presents data to validate its correctness and utility, and compares it to current alternative software applications for OTU-based amplicon analysis.

3.3. Materials and methods

Meta-amplicon analysis algorithm. The core of meta-amplicon analysis algorithm borrows techniques from image registration algorithms. Commonly, image registration is identification of known reference points ("registration marks") in several images and mapping them onto known features in a reference image, transforming other pixels

accordingly [20]. In meta-amplicon analysis, marker sequences (in this case whole 16S sequences) play the role of registration marks, and amplicon sequences are pixels. The method comprises four consecutive stages.

1. MetAmp builds a 2D *reference topology* of microbial populations (Figure 3.1(a), top plane) in which points correspond to known, full 16S gene sequences, and the distances between the points approximates the distances between the corresponding full 16S sequences regions (*reference points: green on top plane*). Later steps will use this topology as a "gold standard" onto which MetAmp will map similar amplicon-induced topologies for each marker sequence. To do this, MetAmp:
    a. Computes the pairwise distances of a set of known, full-length 16S gene sequences in a global alignment; and
    b. Maps the sequences onto a 2D plane (*reference points, green on top plane*) using Sammon Nonlinear Multi-Dimensional Scaling (SNMDS) [21].
2. MetAmp then builds a separate *guided empirical topology* for each marker sequence. These 2D topologies contain both *anchor points* ("registration marks" in the image processing literature) and the user's empirical amplicon sequences, with distances in the plane approximating distances between the underlying sequences. To do this, for *each marker*, MetAmp:
    a. Extracts the marker sequences from the full 16S sequences used in the reference topology, to use as anchor points (Figure 3.1(a, b, c), hollow green circles). MetAmp extracts the marker sequences bioinformatically using same forward and reverse primers that define the marker regions. In essence, this is a digital PCR of marker regions from known full-length 16S genes, with perfect primer matching.

b. Adds the empirical amplicon sequences (Figure 3.1(b, c), filled blue circles), which the user acquired by from high throughput sequencing. Current sequencing technologies produce so many sequences that MetAmp currently performs a data reduction step, replacing the library of all reads with a smaller library of concensus sequences (details below). In this manuscript, and in the MetAmp documentation, "empirical sequences" refers to this reduced set of consensus sequences rather than to the set of all reads from high throughput sequencing.

c. Builds a distance matrix from global pairwise alignment of both anchor and empirical sequences combined, using the same methodology as in building the reference topology above.

d. Maps both anchor and empirical amplicon sequences onto a 2D plane in a distance-approximating way using SNMDS. This guided empirical topology now contains user sequence data embedded into a topology with anchor points, which will guide the next step (Figure 3.1(c), smaller planes).

3. MetAmp then maps each guided empirical plane, one for each marker, onto the reference plane. In this mapping, the anchor points map onto their corresponding reference points, and the empirical points map into the reference plane with the same convolution as in the anchor-to-reference mapping (Figure 3.1(b, c), arrows). Currently, the mapping is an affine transformation, similar to that used to map pixels in images with registration marks onto a reference image [20]. Empirical points that are mapped into the reference plane are called "*normalized*" empirical points.
4. The resulting plane now contains reference sequences corresponding to known 16S genes and normalized empirical points corresponding to user amplicon sequences (Figure 3.1

(c)). The distances between the empirical points correspond to distances between the user amplicon sequences, using the best available (full length 16S) sequence data to correct for distortions caused by the choice of individual markers. The points in the resulting plane are therefore ready for clustering and downstream OTU analysis.

Below we show each step in details.

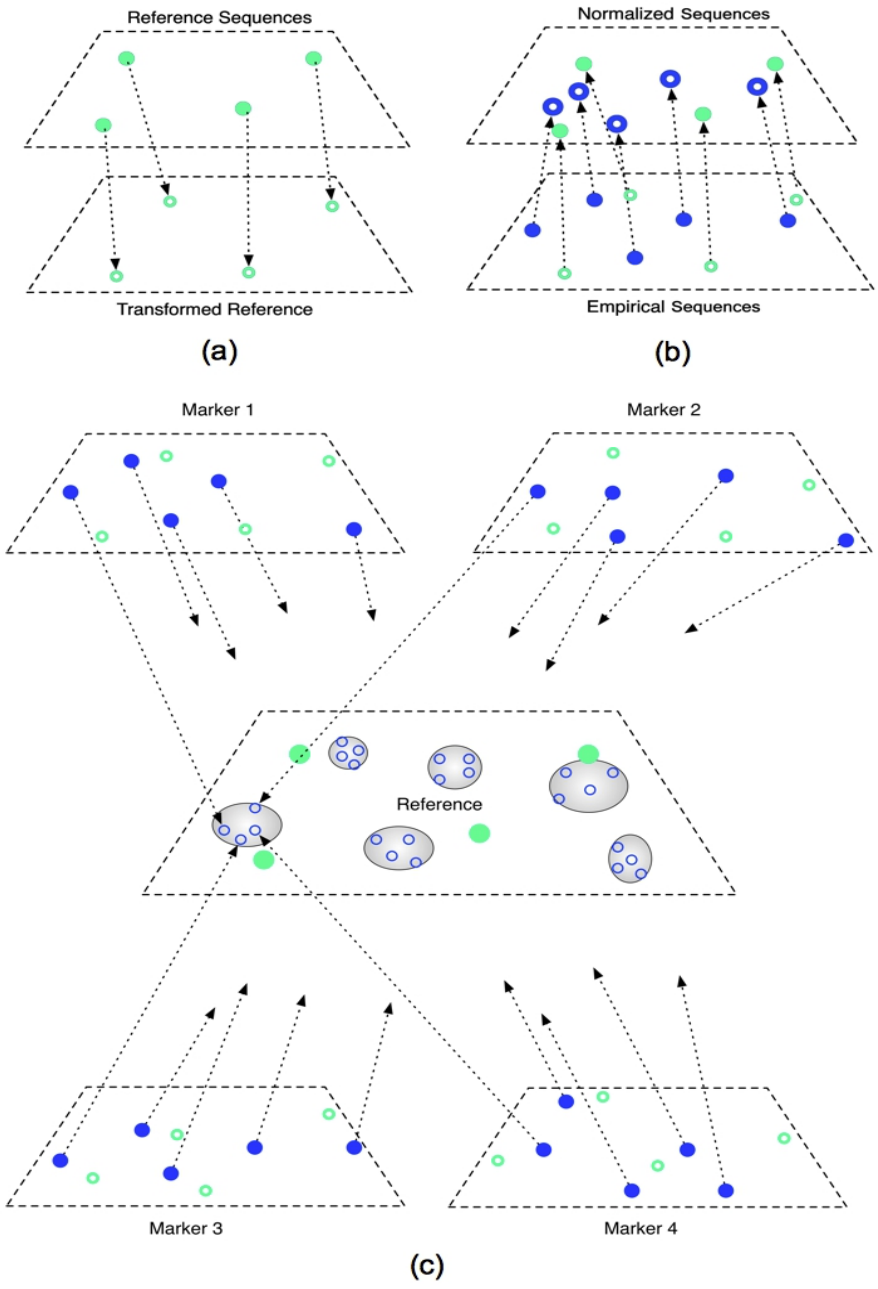

Figure 3.1: Illustration of "meta-amplicon" analysis algorithm. (a) and (b) illustrate the first three steps of the proposed algorithm. In computing the reference topology we use full-length 16S reference sequences. Reference sequences are then pairwise aligned and placed on to the 'reference' 2D plane with multidimensional scaling. To build anchors for empirical reads, marker regions are extracted from corresponding reference 16S sequences and placed on to the plane with the same methodology used above. Next, we add empirical amplicon sequences (solid blue circles), aligned and scaled with the same methodology. Then reference sequences (hollow green circles) are mapped back to the whole 16S space (central plane), carrying empirical sequences (solid blue circles). (c) The whole process described is repeated for each marker region. In the end, we cluster all points from the reference plane into final OTUs.

Step 1, Building reference and guided empirical topologies. Reference sequences come from a reference database (by default we use a dataset containing the ChimeraSlayer reference database in the Broad Microbiome Utilities [24], version microbiomeutil-r20110519. We use this dataset because it contains full forward and reverse-complemented 16S gene sequences and is free of duplicates. At the time of writing this dataset contains approximately 5100 reference 16S bacterial rRNA). We compute dissimilarities with global pairwise alignment via the following equation: $D = 1 - PID$, where *PID* is the pairwise percent sequence identity provided by alignment software USEARCH v7, since it allows the fastest pairwise alignment in comparison to other similar tools. To align sequences, USEARCH employs techniques, similar to BLAST algorithm and uses banded global alignment in order to compute a percent sequence identity. A percent sequence identity is computed using the following parameters (default for USEARCH): Interior gap open = 10 nucleotides; End gap open = 1; Interior gap extend = 1; End gap extend = 0.5. Hence, we have a distance R x R matrix (where R is the number of reference sequences, currently 5,181). To place sequences onto the plane, we use Sammon's Non-linear Multidimensional Scaling (SNMDS), where each reference sequence represents so-called "reference" point (solid green circles in Figure 3.1(a)). Unlike Principal Component Analysis which is commonly used in Multidimensional Scaling, Sammon's mapping minimizes the differences between corresponding inter-point distances in the two spaces, that is it approximately conserves the distance between each pair of points, preserving the topology.

Step 2, Adding empirical amplicon sequences. High-throughput sequencing produces a very large number of reads (an Illumina MiSeq library may contain billions of

reads), so it is necessary to reduce the number of reads before processing. Tools such AbundantOTU [9], USEARCH and CD-HIT allow "compression" of the original sequence library into a set of consensus sequences. By default, MetAmp uses USEARCH v7 with default parameters to replace the full set of sequencing reads with a much smaller set of consensus sequences. MetAmp places these consensus sequences onto the empirical plane along with anchor sequences (Figure 3.1(a), bottom, hollow blue circles).

Mapping between reference and guided empirical topologies. This represents a key step of the meta-amplicon analysis algorithm, which allows "moving" empirical sequences to the reference 16S plane, using “anchor” sequences as a support, which in turn, clusters empirical amplicons together on a reference plane. Such normalized amplicons then clustered together into OTUs. In order to bring empirical sequences back the reference plane, we compute mapping between reference and anchor points and then apply computed mapping to each of empirical amplicon sequence. In the end, amplicon sequences from different marker regions (but obtained from the same species) form dense clusters on reference 16S plane. We use affine transformation between reference points and anchor points. In image registration, affine transformation is used when one needs to map pixels in images with registration marks onto a reference image, therefore we use it. However, we anticipate that some other transformations exist to be used for this purpose. Computing affine transformation between all reference and guide points introduces undesirable distortion; therefore, we perform triangulation of both reference and empirical planes, where reference and corresponding guide points are vertices (see Figure 3.2).

Step 3, Mapping between reference and guided empirical topologies. This is the key step of the meta-amplicon analysis algorithm, which makes it possible to "move" empirical sequences to the reference 16S plane using “anchor” sequences as a support in order to make clustering into OTUs possible. In order to bring empirical sequences back the reference plane, we compute the mapping between reference and anchor points and then apply the same transformation to each of empirical amplicon sequence. In the end, amplicon sequences (reads) from different marker regions that derive from closely related populations form dense clusters on the reference plane. To compute this transformation efficiently, we first form a triangulation of both reference and empirical planes, where reference and corresponding guide points are vertices using the Delaunay algorithm [25]. We then apply affine transformations between regions bounded by triangles of reference points and anchor points (Figure 3.2).

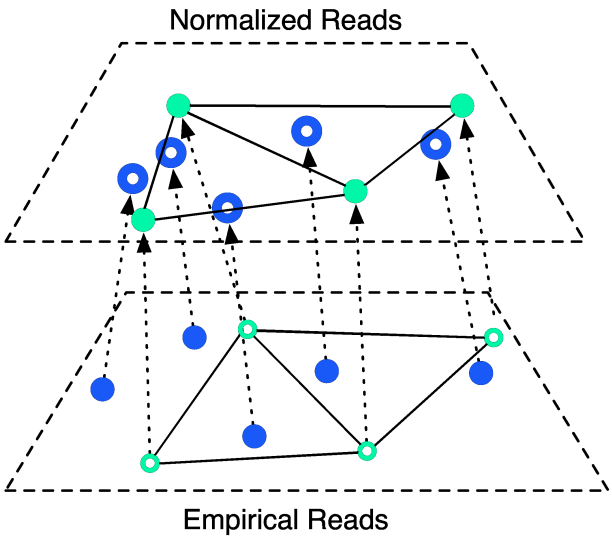

Figure 3.2: An illustration of triangulation where both planes (reference, top plane and empirical, bottom plane) are spitted into triangles and affine transformations are computed between corresponding triangles. We call this a piece-wise mapping between reference and empirical plane.

Step 4, Clustering points. Each empirical point is an empirical amplicon sequence or a consensus sequence bioinformatically formed from several thousands of amplicons (data reduction). After normalization, these points form "clouds", which may or may not be close to reference points. The purpose of clustering is to detect these clouds, which then form the OTUs for downstream analysis.

MetAmp clusters with DBSCAN [22]. Of course, users might prefer to use a different clustering method on the distance information in the normalized plane. DBSCAN automatically determines the optimal number of clusters for correctly chosen parameters. MetAmp supply two parameters to DBSCAN: *Eps*, which is a relative density and *MinPts*, which is the minimum number of points in a cluster that can be density-reachable. By default, *MinPts* = 1. MetAmp estimates *Eps* (Epsilon) as the first percentile of a pairwise distance between reference points, derived from the average density of points in a cluster.

Three types of clusters can be present on the reference plane: (1) clusters that contain both reference and empirical points (upper right hand corner of Figure 3.1(c)); (2) clusters that contain only empirical points (such as lower right hand corner of Figure 3.1(c)); and (3) clusters that contain only reference points (not shown). In the first case, circumscribed reference points are potential identifiers for the empirical reads in the cluster, since they came form known sequences. In the second case, the OTU represents a potentially novel population that can be analyzed with other tools. The third case has no relevance for the empirical data, other than to say that those known populations are not well represented. Note that some “cluster” may be isolated points, or contain very few points. In these cases, it is likely that the corresponding amplicons do not represent the

population well, since amplicons from other markers from the same population have clustered elsewhere. The most conservative strategy is to discard these clusters.

Software. MetAmp is our implementation of meta-amplicon analysis of 16S sequence data. MetAmp is a ready-to-use set of scripts and packages, which requires Python 2.7, R (3.1 or better) and GCC (4.2 or better). MetAmp is available at: http://izhbannikov.github.io/MetAmp/ MetAmp is compatible with Illumina and Roche 454 HTS libraries, though it is no limited to any particular sequencing technology. To run the program, identify sample and reference libraries, and execute the python script *metamp.py* with the appropriate (or default) options. For the majority of tasks most MetAmp parameters can be left to their defaults. Detailed documentation is available from the download site.

Validation methodology. We validated the meta-amplicon approach as implemented in the MetAmp software, and quantified its accuracy relative to other tools. The accuracy of an algorithm in this context is the number of actual OTUs that the tool is able to recover with default parameters. Granted, one could optimize parameters for each tool, but this is a non-trivial task, and we set the defaults for MetAmp to work well in all but the most extreme cases. We tested accuracy on empirical and simulated amplicon sequences, as described below. We also determined the number of reference sequences that led to the best tradeoff between MetAmp performance and accuracy, using these simulation datasets.

Test 1, accuracy on empirical data. We used Human Mock Community (HMC-Mock) empirical data obtained from Human microbiome project (HMP). The HMC-Mock and sequencing protocols are described at HMPDACC (www.**hmpdacc**.org). We used

reads obtained with sequencing V13, V35, and V69 primers (Roche 454 GS FLX Titanium platform) and the following community types: *Even*, where all species have roughly equal concentration (SRR053818, SRR072220, SRR072239), and *Staggered,* where some species had significantly higher concentration that others (SRR072221, SRR072223, SRR072237). These are datasets of 21 known bacterial species that was established by the Human Microbiome Project as a benchmark [18] for sequencing. At the time of this publication, only Roche 454 sequence data were available. We estimated the number of OTUs with MetAmp, UPARSE, Mothur and QIIME. Ideally, the number of OTUs estimated should equal the (known) number of species. Table 3.1 describes these datasets in more detail. This approach was published with UPARSE [18].

Table 3.1: Detailed description of each of the data sets used in this work.

Name	No. of sequences	Sequencing platform	No. of genomes	Gene	Region	Study ID
Mock1-Even	39,088	454 GS-FLX	21	16S rRNA	V35	SRR053818
Mock2-Even	51,313	454 GS-FLX	21	16S rRNA	V13	SRR072220
Mock3-Even	42,227	454 GS-FLX	21	16S rRNA	V35	SRR072220
Mock4-Even	36,031	454 GS-FLX	21	16S rRNA	V69	SRR072239
Mock5-Staggered	57,492	454 GS-FLX	21	16S rRNA	V13	SRR072221
Mock6- Staggered	38,750	454 GS-FLX	21	16S rRNA	V35	SRR072221
Mock7- Staggered	62,860	454 GS-FLX	21	16S rRNA	V13	SRR072223
Mock8- Staggered	42,485	454 GS-FLX	21	16S rRNA	V69	SRR072223
Mock9- Staggered	30,568	454 GS-FLX	21	16S rRNA	V35	SRR072237
Mock10-Staggered	25,871	454 GS-FLX	21	16S rRNA	V69	SRR072237
Simul1	1,010,000	Simulated	100	16S rRNA	V13	-
Simul2	2,000,000	Simulated	100	16S rRNA	V35	-
Simul3	2,000,000	Simulated	100	16S rRNA	V69	-

Test 2, accuracy on simulated sequence data. The empirical datasets of HMC data above were not sufficient for evaluating the ability of multiple markers to discover rare species. HMC has only 21 microbial populations, while typical empirical sample contain hundreds to thousands, with very different abundances. Therefore, we used simulated data to test the accuracy of MetAmp. The general strategy was to randomly select known full length 16S gene sequences from a high quality database (the Gold ChimeraSlayer dataset version microbiomeutil-r20110519 mentioned above. These sequences serve as known reference populations. Next, we extract the relevant marker regions from these known references to serve as ideal amplicon sequences for the known populations. However, actual reads from high throughput sequencers have errors, and the type and distribution of errors varies with the technology. We reproduce this effect by simulating variants of the ideal amplicons above with a bioinformatics tool that captures the appropriate sequence variation for Illumina paired-end reads. The number of reads for each population varied according to a geometric distribution, to approximate the rank abundance curves one would expect from a natural dataset [17]. This produces a large set of sequences that simulate reads from known populations that exhibit the type of variation and copy number one would expect had these known populations been sampled from nature and sequenced using these markers and this technology.

Specifically, we randomly selected 100, 500 and 1000 full 16S sequences from the Gold ChimeraSlayer data set to serve as known microbial populations. We extracted sequences for variable regions V13, V35, and V69 from the full sequences to serve as anchor points (registration marks). For each of these anchor points, we generated up to 5 million (see Table 3.4 for detail) artificial Illumina paired-end reads with lengths of 250

bp, using the sequence simulation tool ART [23] (see Supplementary Materials, Appendix A, for details), with copy numbers drawn from a geometric distribution with p=0.001. We shuffled the 10% most abundant anchor points (10, 50, and 100 anchor points) when determining abundances for the different variable regions, to simulate different markers having different resolution for different abundant species while leaving a common set of rare species. See Figure 3.3 for typical rank abundance curves for simulation data, and Table 5 from Supplementary Materials for species selected.

To summarize, these simulation datasets represent thousands of Illumina amplicon reads that should form 100, 500, and 1000 OTUs with known identities. Using these simulation datasets in place of empirical reads, and the original full length sequences as target "known" populations, we determined the number of OTUs using MetAmp, UPARSE, QIIME, and Mothur. Table 7 shows the percentage of actual OTUs that each tool was able to recover.

Test 3, optimizing number of reference sequences. Clearly, having more reference sequences should improve algorithm accuracy. More reference sequences provide the normalization process with more information, essentially shrinking the size of triangulated regions that must be transformed. However, too many reference sequences will add computational expense without improving results significantly. To quantify this tradeoff, we compare the accuracy of MetAmp as a function of the number of reference sequences. The error induced by MetAmp and the choice of markers is the distance from normalized reads to known, correct positions., which is a function of the number of reference sequences and hence the number of triangles subject to affine transformations.

We defined the accuracy of MetAmp on these datasets as

$$Accuracy = \frac{median\left(\sqrt{(x_e^i - x_t^i)^2 + (y_e^i - y_t^i)^2}\right)}{median(PDT)}$$

Where (x^i_t, y^i_t) is the 2D coordinate of the full length 16S sequence on the reference plane (the "true" position) and (x^i_e, y^i_e) is the 2D coordinate of the corresponding "empirical" sequence after normalization, and PDT is the set of pairwise distance between points on reference plane. Thus, *Accuracy* is the median Euclidean distance from normalized points to their true location, normalized to the median distances between reference points.

For test datasets, we built five test sets by drawing sequences at random from the Gold dataset,with 110, 200, 600, 1100 and 5100 total full-length 16S sequences. We divided these sets into reference and empirical sequences with a ratio of approximately 10 reference sequences to 1 empirical sequence, with one set of empirical sequences for each marker (V13, V35, and V69). See Table 3 ("Accuracy results") from Supplementary Materials for details of these data sets. These datasets were large enough to provide useful data, but small enough to be processed efficiently. See Figure 3.4 for results, which are discussed below.

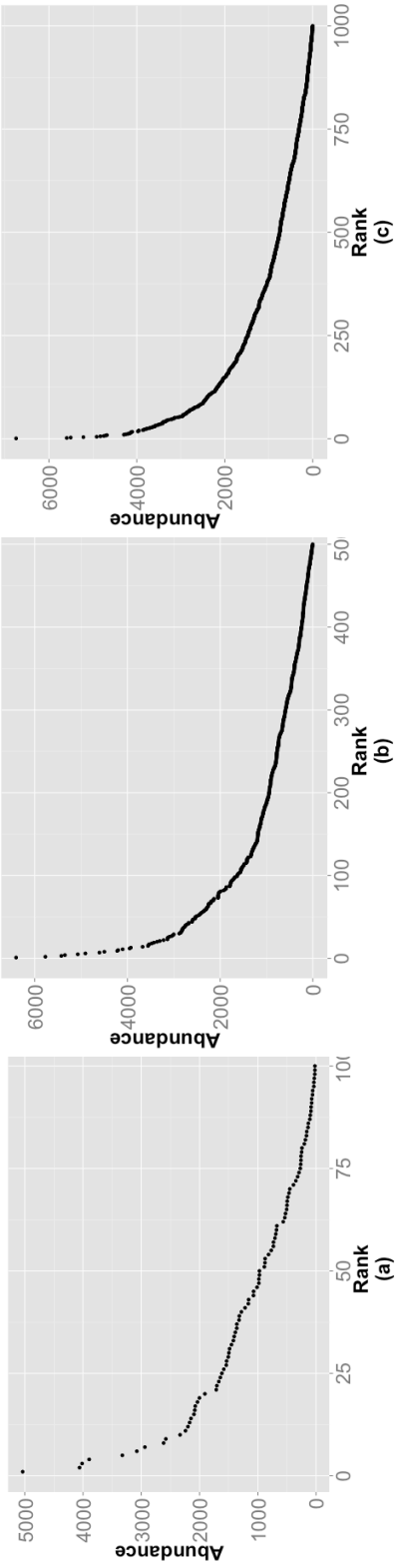

Figure 3.3: Simulated rank abundance curve for 100 (a), 500 (b) and 1000 (c) species. Refer to Supplementary Materials to recover which species were used. To simulate abundance we used geometric distribution with p=0.001.

3.4 Results

The number of OTUs recovered by MetAmp is close to the actual number of species in the community, and compares favorably to other tools, using empirical data. Table 3.2 shows the ratio of average number of OTUs to actual number of populations detected for MetAmp, UPARSE, Mothur, and QIIME using single and multiple 16S rRNA markers on human microbiome mock communities, with equal (Even) and varying population sizes (Staggered), using 97% sequence similarity and average neighbor agglomerative clustering. A perfect score would be 1.0, with larger values ”recognizing” populations that aren’t there, and smaller values failing to recognize some that are. Only MetAmp is capable of using multiple markers, so Table 3.2 shows results for MetAmp with three markers, and with three different single markers, compared with the other tools using single markers. These results for the other tools are comparable with those reported in the literature [18].

We refer to the ratio of recovered OTUs to the actual number of OTUs as the *recovery rate*, which is ideally 1. For the Even community, the recovery rate for is 1.01 when using three markers (V13, V35 and V69), and an average of 0.93 with any single marker. The best results with other tools were produced with UPARSE (1.62). Mothur and QIIME performed worse, with recovery rates of 9.6 and 42.9.

For the Staggered community, the recovery rate for MetAmp was 1.0 with three markers, and an average of 0.85 individual markers. Other tools produced higher recovery rates: 1.39 (UPARSE), 10.18 (Mothur), and 43.3 (QIIME).

Table 3.2: Ratio of average number of OTUs to actual number of populations detected for MetAmp, UPARSE, Mothur, and QIIME using single and multiple 16S rRNA markers on human microbiome mock communities.

Community type	**MetAmp [1]**	**MetAmp [2]**	**UPARSE [2]**	**Mothur [2]**	**QIIME [2]**
Even	1.01	0.93	1.62	9.6	42.9
Staggered	1.0	0.85	1.39	10.18	43.3

[1] Combined markers: V13, V35, V69; [2] Average from each of single marker: V13, V35, V69

Table 3.3: Percentage of detected species for different applications and datasets of 100, 500 and 1000 species. Rank abundances were simulated using geometric distribution. This experiment shows how tested software tools are able to detect rare species. Parameters for each tool are provided in Supplementary Materials.

Marker regions	**Actual No. of species in the community**	**MetAmp**	**UPARSE**	**QIIME**	**Mothur**
		% Of species detected			
V13 + V35 + V69	100	100	-	-	-
V13		-	57	44	49
V35		-	45	47	52
V69		-	38	33	31
V13 + V35 + V69	500	78	-	-	-
V13		-	35	34	31
V35		-	33	37	36
V69		-	31	28	24
V13 + V35 + V69	1000	52	-	-	-
V13		-	15	12	13
V35		-	11	11	8
V69		-	9	7	4

MetAmp finds significantly more OTUs from simulation data sets than the other tools. Table 3.3 shows the average percentage of discovered *OTUs* using MetAmp, UPARSE, QIIME and Mothur. We ran MetAmp with three markers and other tools with a single marker since none of them is capable to use multiple markers.

For data set of 100 OTUs, MetAmp detects 84% of OTUs and outperforms other applications, which use only one marker. Those tools were able to detect no more than 57% (UPARSE) of OTUs. This shows the importance of using multiple markers in microbial studies, even for low-complexity communities. QIIME detected 47% of OTUs and Mothur detected 40%. For data set of 500 OTUs, MetAmp recovers approximately 65%, more than UPARSE (35%), QIIME (37%), and Mothur (36%). For large data sets of 1000 OTUs, MetAmp recovers only approximately 32, more than UPARSE (15%), QIIME (12%), and Mothur (13%).

MetAmp accuracy greatly depends on the number of reference sequences. Figure 3.4 plots the relative error for different numbers of reference sequences (raw data are in Table 6 from Supplementary Materials). This plot shows the accuracy for 10, 100, 500 and 1000 reference points for three markers (V13, V35 and V69) and 100 pseudo-empirical sequences across all tests. The number of pseudo-empirical sequences (100) was the same for all tests. Increasing the number of reference points increases precision. At least 500 reference sequences should be used for meta-amplicon analysis in general, since that is where error starts to reach an asymptote in Figure 3.4.

Computational efficiency significantly depends on the sizes of the reference database and the empirical library. Using the ChimeraSlayer dataset of 5,181 reference bacterial species (forward and reverse complemented whole 16S sequences) and Human

Mock Community data, MetAmp used approximately 7 hours (clock time) on an Intel i5 MacBookPro laptop with 8GB of RAM. This is the maximum time we have observed. There are two potential bottlenecks that reduce algorithm's performance: (1) computing pairwise alignments for sequence similarity; and (2) calculating the mapping between reference and anchor points. The total asymptotic complexity of the meta-amplicon algorithm is $O((N_R + N_A)^2)$, where N_R is the number of reference sequences and N_A is the number of empirical amplicon sequences. Detailed execution time for different reference sets are presented in Table 3.4.

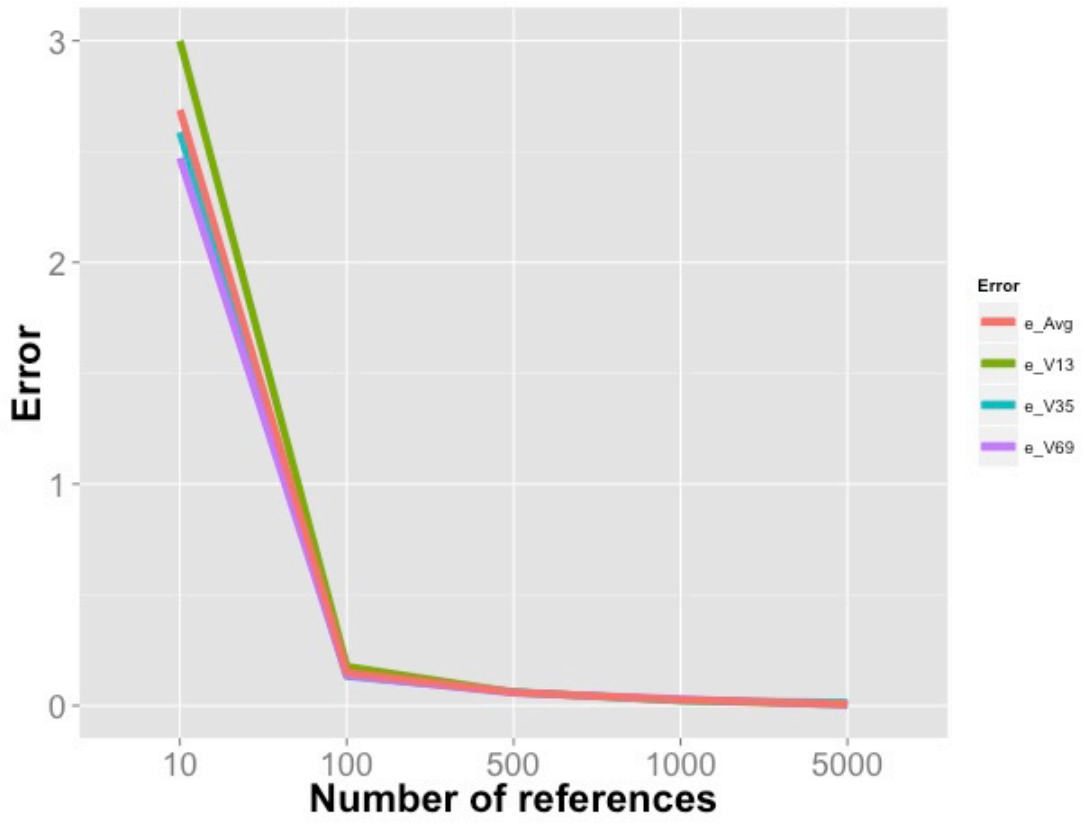

Figure 3.4: Relative difference between 'empirical' and 'true' positions. Here E_V13, E_V35, E_V69 – are ‘errors’, i.e. median difference between ‘empirical’ and ‘true’ positions divided by the median distance between points of the ‘true’ plane. E_Tot – is the average of EV13, EV35 and EV69.

Table 3.4: Execution time of different reference, empirical and simulated data sets tested on 16S rRNA.

Name	No. of sequences	No. of reference sequences	Region	Time (HH:MM:SS)	Memory (Mb)
Mock2-Even+ Mock3-Even+ Mock4-Even	51,313+42,227+ 36,031	100	V35/V13/V69	00:00:38	230
		1500		00:37:15	315
		5181		07:14:25	2079
Mock2-Even+ Mock1-Even+ Mock4-Even	51,313+39,088+ 36,031	100	V35/V13/V69	00:00:37	225
		1500		00:36:41	310
		5181		07:13:45	2058
Mock5-Staggered+ Mock6- Staggered Mock8- Staggered	57,492+38,750+ 42,485	100	V13/V35/69	00:00:38	229
		1500		00:36:24	315
		5181		07:00:31	2100
Mock7- Staggered+ Mock10- Staggered Mock9- Staggered	62,860+30,568+ 25,871	100	V13/V35/V69	0:00:33	220
		1500		0:35:06	310
		5181		06:56:12	2080
Simul1+Simul2+Simul3	1,010,000+2x2, 000,000	100	V35/V13/V69	0:01:33	280
		1500		1:06:36	412
		5181		07:20:04	2161

MetAmp runs slower than other applications. UPARSE was able to complete tests in approximately 15 minutes, on average. QIIME ran for approximately 40 minutes and Mothur approximately 1.5 hours. Detailed execution times for each of the data set are given in Table 3.4.

Analysis of the execution time and memory footprint for different data sets (empirical and simulated) indicates that execution time greatly depends on the size of reference dataset (Table 3.4). For a small reference dataset of 21 sequences, its execution time varies from 33 seconds to 1 minute and 33 seconds. For a middle-size reference data set of 1,500 sequences, its execution time varies between 36 min to 1h 6 min. For large data set of 5,100 species it runs for approximately 7 hours.

3.5 Discussion

This meta-amplicon analysis method makes it possible to use multiple markers for clustering 16S amplicon data into OTUs. However, as with any new bioinformatics technique, using MetAmp requires attention to some technical details, and the resulting analyses depend on specific research questions. We discuss potential limitations and pitfalls in this section.

The relationship between OTUs and taxonomic units is not a simple one. Ideally, one would compare the ability of different OTU clustering techniques to recover underlying taxonomic units by comparing their results on datasets with known microbial community composition. The Human Microbiome Consortium mock community datasets are standards in human microbiome research. However, they represent very small communities, so that they are not stringent tests. It would be preferable to test MetAmp

and other common OTU analysis software on much larger, more complex synthetic communities.

All pipelines that cluster sequences require the user to fix many parameters. But published comparisons often use default parameters, which may explain the poor showing of QIIME and Mothur in Table 3.3 and in published comparisons using the HMC mock communities. In particular, the number of OTUs per detectable species, and in general any estimate of population diversity from OTUs, depends on the chosen sequence similarity threshold chosen, the quality of the data, and the specific clustering algorithm. Depending on the research question, one might choose a similarity threshold other than the commonly used 97% for species level discrimination, with relaxed thresholds leading to fewer OTUs and more strict ones to more OTUs. Sequence errors and PCR artifacts can also significantly affect results. Distance based clustering algorithms are infeasible with modern datasets, since they require unreasonably large distance matrixes. But heuristic clustering algorithms may fail to scale to very large datasets, or may produce clustering artifacts that are difficult to interpret.

Regardless of the OTU analysis algorithms chosen, there are still remain question to be answered regarding the discovery of rare species and species not perfectly targeted by the selected primers. It is not clear how many markers one would need to address these problems, let alone what they would be. Our tests with simulation data show the potential of multi-marker sequencing when one variable region is not a good estimate of species richness. These data also indicate that a minimum of three markers, and therefore at least three guided amplicon topologies, can double the rate at which the algorithm recovers OTUs. One can improve multiple marker cluster calling with a simple majority

vote mechanism, though the current version of MetAmp does not implement this. Suppose two normalized empirical amplicon sequences from different markers that originated from the same 16S rRNA are close to each other on the reference plane, but a third normalized empirical sequence from the same 16S gene is placed far away from the reference (probably due to sequencing error or evolutionary divergence). We can still use these two and discard the outlier. Essentially, two sequences that cluster reinforce the evidence that they belong together, and isolated sequences lack such reinforcement. So, isolated points in the normalized plane are likely to indicate errors, rather than divergent populations.

There are problems with discarding isolated reads, too. Ideally, one would have the same number of reads for each marker region, and the number of reads in an OTU would indicate the diversity of the underlying population. However, discarding reads effectively reduces the number of reads in the remaining OTUs. This may reduce diversity estimates. However, keeping the isolated points would inaccurately inflate diversity estimates by representing a single underlying population twice.

Any OTU based analysis of 16S amplicon data has inherent limitations. For example, estimates of microbial community structure and diversity from OTU data has limited application to estimating community functions. This is a prerogative of metagenomic analysis [1]. PiCrust estimates community functions from OTU analysis using reference genomes and statistical ancestral-state phylogenetic. But for majority of microbial species whole reference genomes are still unsequenced and not. It is also notoriously difficult to estimate population abundance, as opposed to diversity, using 16S data, since 16S genes vary significantly between different bacterial populations and the

number of reads is sensitive to PCR bias. In short, using multiple markers will not be a benefit if OTU analysis itself is inappropriate.

It is not always appropriate to use multiple markers, even when OTU analysis is appropriate. Single-marker amplicon sequencing sometimes allows relatively quick and inexpensive identification to genus or even down to species level, especially for small or simple communities. On the other hand, different markers have different amplification rates even with perfectly designed PCR primers. So, it is possible to miss even highly abundant species, or to have biased sequence sampling, in any PCR based approach. Unfortunately, it is difficult to empirically explore these potential problems, or to validate MetAmp extensively, since there are no existing datasets from single studies with amplicons from multiple markers. We are currently gathering such a dataset, using four markers on human microbiome data.

Of course, sequencing the entire 16S gene will provide more information than any set of markers. This is currently beyond the reach of high throughput sequencing, however. Moreover, sequencing the entire genomes of each population would provide even more information.

Since sequence normalization depends on anchor points, it is important to use an appropriate set of reference sequences. If the set of reference sequences is not representative of one's actual data, there is a risk that normalized amplicon sequences are placed quite away from their "true" positions, producing what we call "distortion". This can reduce the accuracy of OTU clustering, and lead to a situation when a normalized empirical amplicon sequence is assigned to the wrong reference sequence. On the other hand, it requires more computational resources than necessary if many reference

sequences are only distantly related to the empirical data. For this study, we randomly selected reference sequences from all the full length 16S gene sequences that are currently available. But we have also prepared four targeted reference sets: soil, human, marine microbial, and all of these sets. These datasets may provide more accurate clustering or use fewer computational resources for very specific studies.

Combining data from different studies is non-trivial and risky activity that is inappropriate for MetAmp analyses. Two different types of data can potentially be combined: (1) sequence data from different genes, such as 16S, 18S, and ITS; (2) data from different sequencing technologies, such as Roche and Illumina. Our algorithm cannot work on data from different domains. As an example, 16S and 18S are not only different genes but also different molecules. In such cases, analysis should be performed separately for 16S and 18S data. Combining data from different technologies is only useful when the technologies have similar error rates. Technologies with higher error rates introduce bias that will distort OTU determination. However, it may be possible to combine amplicon libraries from different studies for meta-amplicon analysis, if they are from similar technologies.

There are several algorithmic details in MetAmp that deserve further exploration. For example, choosing the correct number of dimensions in multi-dimensional scaling remains an open question. MetAmp uses two dimensions. In our tests (not presented here), the results from 3D and 2D mappings were very similar. So, we did not test higher than 4D mappings. Also, there may be better image transformation methods than affine transformation of triangulated points. Even simple triangulation itself presents difficulties, such as how to triangulate points on the “outside”, the convex hull of all

reference points. We chose the current methodology because it is widely used in the image processing community, and it is relatively efficient. Also, there may be better ways to perform data reduction of raw reads than clustering into consensus sequences. Also, there are more sophisticated methods for computing sequence similarity than simple Hamming distances. Finally, several points in the algorithm, such as piecewise affine transformation of triangles, may benefit from parallelization. Each of these issues provides fodder for future research, and may affect the tradeoff between algorithm efficiency and effectiveness.

It remains unclear what is the best clustering algorithm to use in meta-amplicon analysis. In general, clustering is a non-trivial problem. The number of clusters is not pre-defined, as it needs to be for some supervised clustering methods such as k-means. MetAmp uses the non-supervised clustering algorithm DBSCAN, which requires an epsilon parameter, which can be difficult to determine. MetAmp estimates epsilon with a new heuristic (not described here), since there is no known exact algorithm. Detailed evaluation of clustering methods in order to obtain the most valuable method for meta-amplicon analysis is the topic of future work.

In this manuscript we have described meta-amplicon analysis, which makes it possible for the first time to compare clusters of amplicons from multiple markers. This approach builds a common reference space from known sequences of the gene or genome in which the markers reside, then maps individual amplicons of (potentially unknown) species onto it. Meta-amplicon analysis makes it possible to extend current analysis techniques from single to multiple markers, taking advantage of the strengths of different

markers in different contexts. We also present validation studies for our implementation of meta-amplicon analysis, MetAmp, which is freely available.

3.6 Acknowledgements

This publication was made possible by NIH Grants P20GM016454, P20GM16448 from the INBRE and COBRE programs of the National Center for Research Resources, and by NSF DBI0939454.

3.7 References

[1] Wooley JC, Godzik A, Friedberg I, A Primer on Metagenomics, PLOS Computational Biology, PLoS Comput Biol, 2010, 6(2): e1000667.

[2] David W. Ussery, Tim T. Binnewies, Rodrigo Gouveia-Oliveira, Hanne Jarmer and Peter F. Hallin, "Genome update: DNA repeats in bacterial genomes." Microbiology. 2004 Nov;150(Pt 11):3519-21.

[3] Kuczynski J, Lauber CL, Walters WA, et al., Experimental and analytical tools for studying the human microbiome, Nature Reviews Genetics, 2011, 13:47-58.

[4] Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J, Glöckner FO, The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. 2013, Acids Res. 41 (D1): D590-D596.

[5] Cole JR, Wang Q, Fish JA, Chai B, McGarrell DM, Sun Y, Brown CT, Porras-Alfaro A, Kuske CR, Tiedje JM, Ribosomal Database Project: data and tools for high throughput rRNA analysis, Nucleic Acids Res. 2014;42(1):D633-42.

[6] Kim M, Morrison M, Yu Z, Evaluation of different partial 16S rRNA gene sequence regions for phylogenetic analysis of microbiomes. 2011, J Microb Meth 84(1):81–87.

[7] Foster JA, Bunge J, Gilbert JA, Moore JH, Measuring the microbiome: perspectives on advances in DNA-based techniques for exploring microbial life. 2012. Briefings in Bioinformatics 13(4): 420–429.

[8] Jari Oksanen, F. Guillaume Blanchet, Roeland Kindt, Pierre Legendre, Peter R. Minchin, R. B. O'Hara, Gavin L. Simpson, Peter Solymos,, M. Henry H. Stevens, Helene Wagner, vegan: Community Ecology Package, 2015, https://github.com/vegandevs/vegan.

[9] Ye Y, Identification and Quantification of Abundant Species from Pyrosequences of 16S rRNA by Consensus Alignment, Proceedings (IEEE Int Conf Bioinformatics Biomed). 2011 Feb 4;2010:153-157.

[10] Rabinowitz GB, An introduction to non-metric multidimensional scaling, American Journal of Political Science, Vol. 19, No. 2, 1975.

[11] Yarza P, Richter M, Peplies J, Euzeby J, Amann R, Schleifer KH, Ludwig W, Glöckner FO, Rosselló-Móra R, The All-Species Living Tree project: a 16S rRNA-based phylogenetic tree of all sequenced type strains, Syst Appl Microbiol. 2008 Sep;31(4):241-50.

[12] Schloss PD, Westcott SL, Ryabin T, Hall JR, Hartmann M, Hollister EB, Lesniewski RA, Oakley BB, Parks DH, Robinson CJ, Sahl JW, Stres B, Thallinger GG, Van Horn DJ, Weber CF, Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities, Appl Environ Microbiol. 2009 Dec;75(23):7537-41.

[13] Caporaso JG, Kuczynski J, Stombaugh J et al., QIIME allows analysis of high-throughput community sequencing data, Nat Methods. 2010 May;7(5):335-6.

[14] Langille, M. G.I., Zaneveld, J., Caporaso, J. G.; McDonald, D.; Knights, D.; a Reyes, J.; Clemente, J. C.; Burkepile, D. E.; Vega Thurber, R. L.; Knight, R.; Beiko, R. G.; and Huttenhower, C., Predictive functional profiling of microbial communities using 16S rRNA marker gene sequences, *Nature Biotechnology*, 1-10. 8 2013.

[15] Daniel Beck, Christopher Dennis and James A. Foster, Seed: a user-friendly tool for exploring and visualizing microbial community data, Bioinformatics, 2014.

[16] G.C. Baker, J.J. Smith, D.A. Cowan, Review and re-analysis of domain-specific 16S primers, 2003, Journal of Microbiological Methods.

[17] Harte J, Maximum Entropy and Ecology: A Theory of Abundance, Distribution, and Energetics, 2011, OUP Oxford, pp. 55-61.

[18] Edgar, R. C. (2013) Uparse: highly accurate otu sequences from microbial amplicon reads. *Nat Meth*, 10, 996–998.

[19] Weizhong Li, Adam Godzik, Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences, Weizhong Li & Adam Godzik Bioinformatics, (2006) 22:1658-9.

[20] Zitov, B. and Flusser, J. (2003) Image registration methods: a survey. *Image and Vision Computing*, 21, 977–1000.

[21] Sammon, J. (1969) A nonlinear mapping for data structure analysis. *IEEE Trans on Comp*, 18, 401–409.

[22] Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). Simoudis, Evangelos; Han, Jiawei; Fayyad, Usama M., eds. A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the

Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.

[23] Ilya Y. Zhbannikov, James A. Foster, MetAmp: combining amplicon data from multiple markers for OTU analysis, Bioinformatics 2015; doi: 10.1093/bioinformatics/btv049.

[24] Haas BJ, Gevers D, Earl A, Feldgarden M, Ward DV, Giannokous G, Ciulla D, Tabbaa D, Highlander SK, Sodergren E, Methe B, Desantis TZ, Petrosino JF, Knight R, Birren BW, Chimeric 16S rRNA sequence formation and detection in Sanger and 454-pyrosequenced PCR amplicons, Genome Res. 2011 Mar; 21(3):494-504.

[25] Delaunay, Boris: Sur la sphère vide. A la mémoire de Georges Voronoï, Bulletin de l'Académie des Sciences de l'URSS, Classe des sciences mathématiques et naturelles, No. 6: 793–800, 1934.

Chapter 4

METAMP: COMBINING AMPLICON DATA FROM MULTIPLE MARKERS FOR OTU ANALYSIS

Chapter 4 presents an application MetAmp that was developed for multi-marker analysis of 16S amplicon data. This allows researchers to better characterize microbial community composition using multiple genomic markers. MetAmp is open source and available at http://izhbannikov.github.io/MetAmp.

This MetAmp application note was published in the Oxford Journal of Bioinformatics. The published manuscript is reprinted here. License information and reprinting permission is shown in the Appendix and citation information is shown below.

Ilya Y. Zhbannikov, James A. Foster, "MetAmp: combining amplicon data from multiple markers for OTU analysis", Bioinformatics 2015; doi: 10.1093/bioinformatics/btv049

4.1 Abstract

Motivation

We present a novel method and corresponding application, MetAmp, to combine amplicon data from multiple genomic markers into Operational Taxonomic Units (OTUs) for microbial community analysis, calibrating the markers using data from known microbial genomes. When amplicons for multiple markers such as the 16S rRNA gene hypervariable regions are available, MetAmp improves the accuracy of OTU-based methods for characterizing bacterial composition and community structure. MetAmp works best with at least three markers, and is applicable to non-bacterial analyses and to non-16S markers. Our application and testing have been limited to 16S analysis of microbial communities.

Results

We clustered standard test sequences derived from the Human Microbiome Mock Community (HMMC) test sets and compared MetAmp and other tools with respect to their ability to recover Operational Taxonomic Units (OTUs) for these benchmark bacterial communities. MetAmp compared favorably to QIIME, UPARSE and Mothur using amplicons from one, two, and three markers.

Availability

MetAmp is available at http://izhbannikov.github.io/MetAmp

Contact: ilyaz@uidaho.edu, foster@uidaho.edu

Supplementary information

Available at Bioinformatics online.

4.2 Introduction

High-throughput sequencing technologies allow researchers to characterize microbial community composition and structure without first cultivating the microbes. There are two current techniques for analyzing microbial communities: metagenomic and genomic marker sequencing. Metagenomic analysis fragments and sequences all DNA in a sample, and then optionally assembles and maps genes into annotated genomes. One often uses metagenomic analysis to characterize metabolic potential.

Marker sequencing amplifies and sequences conserved but variable genomic regions, usually hypervariable regions of the essential 16S rRNA gene. One then clusters these sequences, known as amplicons, into Operational Taxonomic Units (OTUs). This approach can handle unknown and highly diverged populations, provided that the OTUs correspond with sufficient accuracy to relevant ecological units. Metagenomic and amplicon analysis therefore answer different research questions. MetAmp targets only amplicon analysis.

Most marker-based studies use amplicons of just one marker, and it is difficult to know which marker to select *a priori*. Amplicons from different markers in a given sample, or even from a single species, can cluster differently, giving very different pictures of the underlying microbial community. Different markers may also lead to very different phylogenies and taxonomies, which can differ substantially from those deduced from known genomes. Unfortunately, current bioinformatics techniques assume a single marker. This is true of the best currently available software for amplicon OTU analysis, such as UPARSE [1], QIIME [2], and Mothur [4]. We introduce the "Meta-amplicon analysis" technique, MetAmp, which makes it possible to clusters and analyze amplicons from multiple markers.

4.3 Methods and algorithms

MetAmp borrows from image registration algorithms in image processing. These identify known reference points, called "registration marks", in several images and map them onto known features in a reference image, transforming other pixels accordingly [5]. In MetAmp, marker sequences play the role of registration marks, and amplicon sequences that of pixels. MetAmp works as follows:

1. MetAmp builds a 2D *reference topology* of microbial populations (Figure 4.1(a), top plane) in which points correspond to known, full 16S gene sequences and the distances between the points approximates the distances between the corresponding full 16S sequences region (*reference points: green on top plane*). Later steps will use this topology as a "gold standard" onto which MetAmp will map similar amplicon-induced topologies for each marker. To do this, MetAmp:
    a. Computes the pairwise distances of a set of known, full-length 16S gene sequences in a global alignment; and

b. Maps the sequences onto a 2D plane (*reference points, green on top plane*) using Sammon Nonlinear Multi-Dimensional Scaling (SNMDS) [3].

2. MetAmp then builds a separate *guided empirical topology* for each marker sequence. These 2D topologies contain both *anchor points* ("registration marks" in the image processing literature) and the user's empirical amplicon sequences, with distances in the plane approximating distances between the underlying sequences. To do this, *for each marker*, MetAmp:

    a. Extracts the marker sequences from the full 16S sequences used in the reference topology, to use as anchor points (Figure 4.1(a, b, c), hollow green circles). MetAmp extracts the marker sequences bioinformatically using same forward and reverse primers that define the marker regions. In essence, this is a digital PCR of marker regions from known full length 16S genes, with perfect primer matching.

    b. Adds the empirical amplicon sequences (Figure 4.1(b, c), filled blue circles), which the user acquired by from high throughput sequencing.

    c. Builds a distance matrix from global pairwise alignment of both anchor and empirical sequences combined, using the same methodology as in building the reference topology above.

    d. Maps both anchor and empirical amplicon sequences onto a 2D plane in a distance-approximating way using SNMDS. This guided empirical topology now contains user sequence data embedded into a topology with anchor points, which will guide the next step (Figure 4.1(c), smaller planes).

3. MetAmp then maps each guided empirical plane, one for each marker, onto the reference plane. In this mapping, the anchor points map onto their corresponding reference points, and the empirical points map into the reference plane with the same convolution as in the anchor-to- reference mapping (Figure 4.1(b,c), arrows). Currently, the mapping is an affine transformation, similar to that used to map pixels in images with registration marks onto a reference image.
4. The resulting plane contains reference sequences corresponding to known 16S genes and empirical points corresponding to user amplicon sequences (Figure 4.1 (c)). The distances between the empirical points correspond to distances between the user amplicon sequences, using the best available (full length 16S) sequence data to correct for distortions caused by the choice of individual markers. The points in the resulting plane are therefore ready for clustering and downstream OTU analysis.

4.4 Validation

We validated MetAmp by performing OTU analysis on two amplicon datasets from the Human Microbiome Mock Community www.hmpdacc.org/HMMC, namely the ”Even” (SRR072220, SRR072239), and the ”Staggered” communities (SRR072221, SRR072223, SRR072237). These two datasets include amplicons from three marker regions (V13, V35, and V69) from Roche 454 GS FLX Titanium sequencing of two communities with 22 known species. Illumina paired end sequences were not available for the HMMC at the time of this writing.

These experiments test the ability of standard OTU analysis pipeline to recover known OTUs, both with and without multiple marker data that has been pre-processed with MetAmp. We computed the average number of OTUs formed by clustering these data using MetAmp,

UPARSE, Mothur, and QIIME and report the ratio of this average to the known number of populations (22). Table 1 reports typical results (see Supplementary Data for full results). A perfect score would be 1.0, with larger values ”recognizing” populations that aren’t there, and smaller values failing to recognize some that are. We tested each tool on V13, V35, V69 individually, and additionally tested MetAmp on pairs and triplets of these markers. MetAmp ran for about 8 hours using the whole reference data set (abut 5.1k genes) on Intel Core i5 Macbook Pro laptop.

4.5 Acknowledgement

This work was made possible by NIH Grants P20GM016454, P20GM16448 from the INBRE and COBRE (NCRR), and by NSF DBI0939454.

4.6 References

[1] Edgar, R. C. (2013) Uparse: highly accurate otu sequences from microbial amplicon reads. *Nat Meth*, 10, 996–998.

[2] Kuczynski, J., Stombaugh, J., Walters, W. A., Gonzalez, A., Caporaso, J. G. and Knight, R. (2011) Using qiime to analyze 16s rrna gene sequences from microbial communities. *Curr Protoc Bioinformatics*, Chapter 10, Unit10.7.

[3] Sammon, J. (1969) A nonlinear mapping for data structure analysis. *IEEE Trans on Comp*, 18, 401–409.

[4] Schloss, P. D., Gevers, D. and Westcott, S. L. (2011) Reducing the effects of pcr amplification and sequencing artifacts on 16s rrna-based studies. *PLoS ONE*, 6.

[5] Zitov, B. and Flusser, J. (2003) Image registration methods: a survey. *Image and Vision Computing*, 21, 977–1000.

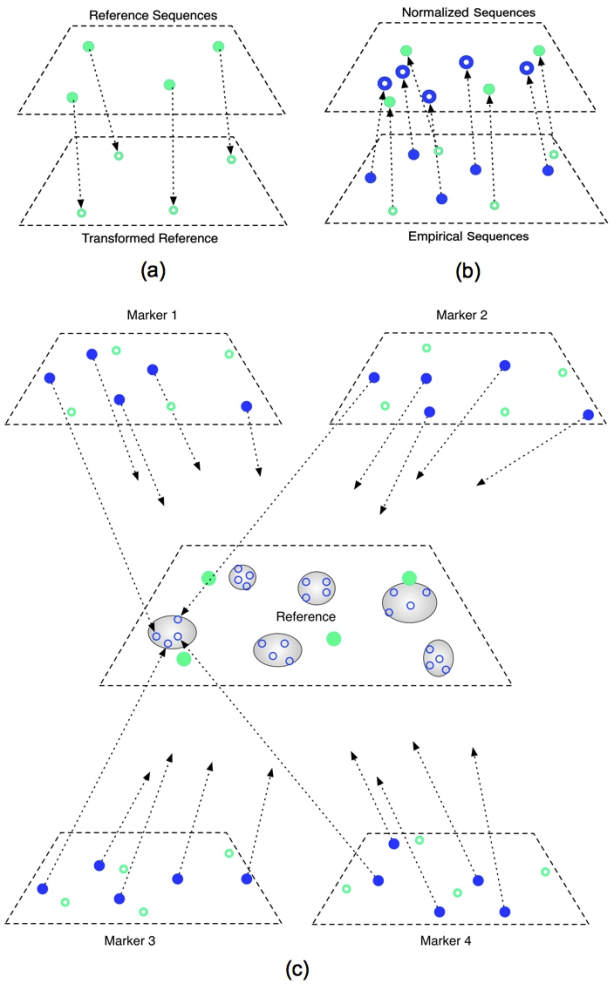

Figure 4.1: Illustration of the MetAmp algorithm for combining amplicons from multiple marker sequences. See the user manual and supplementary materials for more on the analysis workflow, algorithmic complexity, parameters, and examples.

Table 4.1: Ratio of average number of OTUs to actual number of populations detected for MetAmp, UPARSE, Mothur, and QIIME using single and multiple 16S rRNA markers on human microbiome mock communities, using 97% sequence similarity.

Community type	**MetAmp** [1]	**MetAmp** [2]	**UPARSE** [2]	**Mothur** [2]	**QIIME** [2]
Even	1.06	0.93	1.62	9.6	42.9
Staggered	1.0	0.85	1.39	10.18	43.3

[1] Combined markers: V13, V35, V69; [2] Average from each of single marker: V13, V35, V69

CONCLUSIONS AND FUTURE DIRECTIONS

This dissertation describes three projects developing effective preprocessing algorithms and software for preparing HTS data for subsequent bioinformatic analysis.

In Chapter 1, I presented SeqyClean, a HTS data preprocessing software tool for data denoising, which is one of the most important stages in bioinformatic pipelines. SeqyClean is a modular pipeline, which filters most sequencing noise, according to evaluation tests and user feedback. SeqyClean is simple to use, and is one of the most-comprehensive HTS data preprocessing applications.

There are several concerns with SeqyClean that should be addressed when designing future studies. First, the quality of vector and contaminant filtering directly depends on how well the chosen reference vector and contaminant genomes recapitulate those presented in a library of interest. It is possible to miss a majority of vector sequences due to an incorrectly chosen reference; therefore using right reference genomes is critical. Second, potential users should be familiar with command-line interface. This can be alleviated with a convenient GUI (Graphical User Interface), which is a topic for future work. SeqyClean is actively maintained and I am considering adding new options, such as dust score and chimera filtering.

Chapter 2 presents a preprocessing read-recruiter tool, SlopMap. The outcome of SlopMap is a set of reads similar to the provided target (reference). This set of reads can be in two common data formats: FASTQ for paired- and single-end sequences and SFF for Roche 454 libraries. As is often the case with bioinformatics tools, the quality of the output is sensitive to the parameters that the user chooses. For example, the reference genomes for SlopMap must be chosen correctly, to be relevant to the research question.

Due to its simplicity, SlopMap performs (for correctly chosen parameters) at least as quickly and accurately as most existing general-purpose alignment tools. However, several types of optimization are still possible. These may include using more efficient dictionaries and hash-tables; loop unrolling; removing data dependencies; or even low-level programming. It will also be useful to add multi-processing system support, lower memory usage, better detection of similar sequences, user-friendlier interface. SlopMap is actively maintained and the second version is under development.

The last two chapters (Chapter 3 and 4) demonstrate a novel method for analyzing microbial communities with multi-marker 16S data, the meta-Amplicon analysis algorithm, as implemented in the MetAmp tool. The primary contribution of this tool is that it makes it possible to use multiple markers for OTU-based 16S data analysis. However, there are many concerns regarding our meta-amplicon algorithm that should be addressed in the future.

The first potential problem is that the algorithm can still be confounded by rate heterogeneity when different variable regions evolve with different rates. Such cases should be tracked and reported, so that they can be addressed by for future work.

Another potential concern is computational efficiency. In particular, the time required for pairwise sequence alignment (all against all) could be extremely large for reference sets of more than 5,000 OTUs. Therefore, it may be possible to improve performance by applying more efficient pairwise alignment tools. This is the most significant bottleneck of our algorithm and must be addressed in the future.

MetAmp currently uses two-dimensional ordination. But it is unclear how many dimensions best avoids the distortion introduced by the ordination method itself. The

nonmetric multidimensional scaling (NMDS) [37] can potentially be applied but this still requires significant exploratory work.

Finally, MetAmp currently clusters with DBSCAN, which requires knowledge of parameters such as Epsilon (related to density of points) and MinPts (a minimum number of points in a cluster). In this work, I proposed a heuristic method for estimating values for these parameters, but it is not guaranteed to be optimal. Finding optimal values of these parameters still remains an open question.

Sequencing costs less time and money than ever before. Consequently, researchers have significantly more data and analysis options than before. Developing efficient preprocessing tools for analyzing sequence data helps us answer important biological questions. I hope that this thesis makes progress toward that end.

APPENDICES

Appendix A: Application parameters and commands used in validation

MetAmp

Three markers (V13, V35, V69):

```
python metamp.py -r data/gold21/gold21.fasta -r1 data/gold21/gold21_V13V31.fasta -l1
data/even/SRR072220_V13V31_relabeled.fastq -r2 data/gold21/gold21_V35V53.fasta -l2
data/even/SRR072220_V35V53_relabeled.fastq -r3 data/gold21/gold21_V69V96.fasta -l3
data/even/SRR072239_V69V96_relabeled.fastq -o testV13V35V69
```

Single marker (V13):

```
python metamp.py -r data/gold21/gold21.fasta -r1 data/gold21/gold21_V13V31.fasta -l1
data/even/SRR072220_V13V31_relabeled.fastq -o testV13
```

Single marker (V35):

```
python metamp.py -r data/gold21/gold21.fasta -r1 data/gold21/gold21_V35V53.fasta -l1
data/even/SRR072220_V35V53_relabeled.fastq -o testV35
```

Single marker (V69):

```
python metamp.py -r data/gold21/gold21.fasta -r1 data/gold21/gold21_V69V96.fasta -l1
data/even/SRR072239_V69V96_relabeled.fastq -o testV69
```

UPARSE

(markers V13, V35, V69 were clustered separately):

```
# Denoising:
usearch7 -fastq_filter raw_lib.fastq -fastaout denoisedlib -
fastq_truncqual 15 -fastq_trunclen 250
# Dereplication:
usearch7 -derep_fulllength denoised_lib –output drep_lib -sizeout
# Pre-clustering:
usearch7 -cluster_smallmem drep_lib –centroids preclust_lib -sizeout -id
0.99 -maxdiffs 1
# Sorting (remove singletons):
usearch7 –sortbysize preclust_lib -output sort_lib -minsize 2
# Clustering:
usearch7 -cluster_otus sort_lib –otus cluster_lib -minsize 2
# Filtering chimeric sequences:
usearch7 -uchime_ref cluster_lib –db gold.fa -strand plus –nonchimeras
nochimeric_lib
# Final otus:
python python_scripts/fasta_number.py nochimeric_lib OTU_ > finalotus
# Assign reads to OTUS:
usearch7 -usearch_global denoised_lib –db final_otus -strand plus -id
0.97 –uc maptable
```

Mothur

We analyzed 454 reads following the procedure described at http://www.mothur.org/wiki/Schloss_SOP

```
sffinfo(sff=reads.sff, flow=T)
trim.flows(flow=reads.flow, oligos=oligos.txt, pdiffs=2,
bdiffs=1,processors=4)
shhh.flows(file=reads.flow.files, processors=4)
trim.seqs(fasta=reads.v35.shhh.fasta, name=reads.v35.shhh.names,
oligos=oligos.txt, pdiffs=2, bdiffs=1, maxhomop=8, minlength=200,
flip=T, processors=4)
unique.seqs(fasta=reads.v35.shhh.trim.fasta,
name=reads.v35.shhh.trim.names)
align.seqs(fasta=reads.v35.shhh.trim.unique.fasta,
reference=silva.bacteria.fasta, processors=4)
screen.seqs(fasta=reads.v35.shhh.trim.unique.align,
name=reads.v35.shhh.trim.unique.names, group=reads.v35.shhh.groups,
end=27659, optimize=start, criteria=95, processors=4)
filter.seqs(fasta=reads.v35.shhh.trim.unique.good.align, vertical=T,
trump=., processors=4)
unique.seqs(fasta=reads.v35.shhh.trim.unique.good.filter.fasta,
name=reads.v35.shhh.trim.unique.good.names)
pre.cluster(fasta=reads.v35.shhh.trim.unique.good.filter.unique.fasta,
name=reads.v35.shhh.trim.unique.good.filter.names,
group=reads.v35.shhh.good.groups, diffs=2)
chimera.uchime(fasta=reads.v35.shhh.trim.unique.good.filter.unique.precl
uster.fasta,
name=reads.v35.shhh.trim.unique.good.filter.unique.precluster.names,
group=reads.v35.shhh.good.groups, processors=4)
remove.seqs(accnos=reads.v35.shhh.trim.unique.good.filter.unique.preclus
ter.uchime.accnos,
fasta=reads.v35.shhh.trim.unique.good.filter.unique.precluster.fasta,
name=reads.v35.shhh.trim.unique.good.filter.unique.precluster.names,
group=reads.v35.shhh.good.groups)
system(cp
reads.v35.shhh.trim.unique.good.filter.unique.precluster.pick.names
final.names)
system(cp
reads.v35.shhh.trim.unique.good.filter.unique.precluster.pick.fasta
final.fasta)
dist.seqs(fasta=final.fasta, cutoff=0.15, processors=4)
cluster(column=final.dist, name=final.names)
get.oturep(column=final.dist, name=final.names, fasta=final.fasta)
quit()
```

QIIME

I used the methodology described in http://qiime.org/tutorials/tutorial.html "454 Overview Tutorial: de novo OTU picking and diversity analyses using 454 data".

ART (for simulation of artificial Illumina reads)

```
./art_illumina -amp -sam -i ../metamp_data/gold100_V13.fasta -l 50
-c 10000 -o  ../metamp_data/amplicon_pair_dat
```

Appendix B: Complexity of the meta-amplicon algorithm

I use big-O notation, which is asymptotic and considers worst case of time.

1. Pairwise alignment of reference sequences: $O(N_R^2)$, where N_R represents total number of reference species.

2. Pairwise alignment of marker sequences & empirical amplicon reads: $O((N_R + N_A)^2)$, where N_A is total number of empirical amplicon reads.

3. Triangulation: $O(N_T) = O(N_R)$, where N_T represents total number of triangles, $N_T = N_R - 2$.

4. Affine transformation: $O(N_T) = O(N_R)$, (for simplicity we say that it takes constant time), where $N_T = N_R - 2$ is a number of triangles.

5. DBSCAN algorithm complexity: $O(N_P \log(N_P))$, where N_P represents total number of clustering points: $N_P = N_R + N_A$.
6. Total asymptotic complexity will be: $O((N_R + N_A)^2)$