Summer 8-13-2021

StrainIQ: An n-gram-based Method to Identify and Quantify Microbial Communities in Metagenomic Samples

Sanjit Pandey
University of Nebraska Medical Center

StrainIQ: an *n*-gram-based method to identify and quantify microbial communities in metagenomic samples.

By

Sanjit Pandey

A DISSERTATION

Presented to the Faculty of

The Graduate College in the University of Nebraska

In Partial Fulfilment of the Requirements

For the Degree of Doctor of Philosophy

Biomedical Informatics Graduate Program

Department of Genetics, Cell Biology and Anatomy

Under the Supervision of Professor Chittibabu (Babu) Guda

University of Nebraska Medical Center

Omaha, Nebraska

June 2021

StrainIQ: an *n*-gram-based method to identify and quantify microbial communities in metagenomic samples.

Sanjit Pandey, Ph.D.

University of Nebraska Medical Center, 2021

Advisor: Chittibabu (Babu) Guda, Ph.D

Microbes are ubiquitous in nature, and they play vital roles in various processes associated with metabolism in the human body, photosynthesis in plants, or decomposition of waste in the environment. Hence, it is essential to understand how the composition of microbial communities affects the ecosystem of different environments ranging from ocean floors to hot springs to a human body. Microbial communities present in different human body sites are of particular importance due to their implications in the cause and prevention of human diseases. The traditional approaches limit microbial research to exclusively studying species that can be successfully cultured in the lab. With the advent of next-generation sequencing (NGS) technologies, our ability to study microbial communities' composition and function has increased rapidly without having to culture isolated species. More importantly, strain-level diversity is what uniquely identifies an individual's microbiome. In many cases, strain-level variation determines a microbe's ability to cause diseases, resist antibacterial drugs, or be completely harmless. Hence, we must have the ability to identify microbes at a strain-level to effectively design personalized treatment regimens for patients. Many tools have been developed to identify the taxonomic composition using short-read sequencing data from metagenomics samples. They are either alignment-based, longer k-mer based, or SNPs/SNVs based and use more generic databases of genomes containing all the known microbial species. However, most of these methods were designed to predict higher level taxa and hence are not suitable for strain-level prediction. These methods

are also very sensitive to the quality of the reference genomes and the coverage uniformity of the sequencing, while a vast majority of publicly available microbial genomes are incomplete. Due to these limitations, the existing methods do not perform well for the identification of taxa at the strain level.

We developed a tool called StrainIQ (**Strain** **I**dentification and **Q**uantification), to identify and quantify microbial species at the strain-level using the whole-genome sequencing (WGS) data from metagenomic samples. StrainIQ takes advantage of the discriminative nature of unique and weighted common *n*-grams present in complete or draft assemblies of microbial genomes. Additionally, StrainIQ leverages the body site-specific reference genome information to increase the specificity of the prediction. Comparison with popular existing tools shows that StrainIQ is consistently better than other methods at predicting strains with higher sensitivity and specificity. Similarly, StrainIQ is able to estimate the abundance more accurately in comparison to other methods. We also developed a standalone version of the StrainIQ tool and made it available to the public via Github (https://github.com/sanpande/StrainIQ)

ACKNOWLEDGEMENT

I would like to express my special appreciation and thanks to my advisor Professor Dr. Chittibabu (Babu) Guda for guiding and supporting me over the years. Your continuous support and encouragement made this dissertation possible and allowed me to grow into the scientist I am today. I cannot thank you enough for your invaluable advice and guidance in both research as well as on my career. I would also like to thank my supervisory committee members, Dr. Kenneth W Bayles, Dr. Kusum K. Kharbanda, and Dr. Sanjukta Bhowmick for all the invaluable suggestions and critical feedback that positively influenced the project.

Huge thanks to all the past and present members of the Guda lab for your support throughout the project. Your feedback and suggestions during the lab meetings were an integral part of the project. Many thanks to Dr. Neetha N Vellichirammal for your help with the manuscript review and suggestions.

Table of Contents

ACKNOWLEDGEMENT ..... *iv*

List of Figures ..... *vii*

List of Tables ..... *ix*

List of Abbreviations ..... *x*

Chapter 1: INTRODUCTION *1*

1. Introduction to metagenomics **1**

1.1. Amplicon sequencing **2**

1.2. Whole genome sequencing **3**

2. The Human Microbiome **3**

2.1. Significance of human microbiome **4**

2.2. The Human Microbiome Project **7**

3. Metagenomics Sequencing and Data Analysis – Big Picture **9**

3.1 Use of sequencing technologies in metagenomics **11**

4. Summary **12**

Chapter 2: IDENTIFICATION AND QUANTIFICATION OF METAGENOMICS SAMPLES *13*

1. Introduction **13**

2. Current methods **14**

2.1. Alignment-based methods **14**

2.2. Alignment-free methods **16**

2.3. Popular tools and methods **17**

3. StrainIQ method Overview **18**

4. StrainIQ – DNA Signature Element Model Building **20**

4.1. *n*-gram optimization and encoding **23**

4.1.1 Huffman Encoding 26

4.2. Scoring function **31**

5. StrainIQ – I **34**

5.1. Score threshold determination **37**

6. StrainIQ – Q **39**

7. Computational complexity **40**

8. Conclusions **42**

Chapter 3: PERFORMANCE TESTING AND OPTIMIZATION *43*

1. Introduction **43**

2. Materials and Methods **44**

2.1. Datasets 44
2.1.1. Reference datasets 44
2.1.2. Test datasets 44
2.2. Detailed analysis pipelines 45
3. Statistical Measurements 49
4. Results 50
4.1. StrainIQ prediction based on simulated datasets 50
4.2. StrainIQ prediction based on experimental datasets 52
5. Comparison against other popular methods 56
5.1. Identification 56
5.2. Quantification 59
6. Discussion and conclusions 64
Chapter 4: DISTRIBUTION of StrainIQ 66
1. Introduction 66
2. Configuration and installation 66
3. Supporting database and configuration files 67
Chapter 5: PROJECT SUMMARY AND FUTURE DIRECTIONS 68
REFERENCES 70
Appendix 1: GI tract DSEM stats 74
Appendix 2: Mock community genomes 87
Appendix 3: Sensitivity/Specificity comparison of StrainIQ, KrakenUniq, MetaPhlAn, and CLARK 88

List of Figures

FIGURE 1: METAGENOMICS. THE FIGURE SHOWS THE WHOLE GENOME SHOTGUN SEQUENCING OVERVIEW WITH THE GASTROINTESTINAL TRACT AS THE COLLECTION ENVIRONMENT. 2

FIGURE 2: BACTERIAL 16S RDNA. THE FIGURE SHOWS THE HYPER-VARIABLE REGIONS THAT VARY AMONG DIFFERENT BACTERIA. 3

FIGURE 3: COMPLETE HUMAN "GENOME" COMPOSITION 4

FIGURE 4: OVERVIEW OF BACTERIAL INFECTIONS 6

FIGURE 5: HMP SAMPLE COLLECTION BODY SITES 7

FIGURE 6: HMP BODY SITES. THE FIGURE SHOWS THE DISTRIBUTION OF DIFFERENT PROJECTS ACROSS DIFFERENT BODY SITES AS PART OF THE HMP SUB-PROJECTS. 9

FIGURE 7: THE BIG PICTURE. THE FIGURE SHOWS DIFFERENT STEPS INVOLVED IN METAGENOMICS EXPERIMENTS. 10

FIGURE 8: IDENTIFICATION AND QUANTIFICATION 13

FIGURE 9: SEQUENCE ALIGNMENT 15

FIGURE 10: N-GRAMS 16

FIGURE 11: STRAIN IDENTIFICATION AND QUANTIFICATION - OVERVIEW 20

FIGURE 12: DSEM BUILDING 22

FIGURE 13: DSEM EXAMPLE 23

FIGURE 14: UNIQUE AND TOTAL N-GRAMS COUNT COMPARISON FOR DIFFERENT N-SIZES 24

FIGURE 15: N-GRAM SIZE COMPARISON 26

FIGURE 16: HUFFMAN TREE 27

FIGURE 17: NUCLEOTIDE ENCODING 28

FIGURE 18: UNIQUE N-GRAMS DISTRIBUTION FOR GUT GENOMES 32

FIGURE 19: SCORE OPTIMIZATION COMPARISON. 33

FIGURE 20: IDENTIFICATION WORKFLOW 35

FIGURE 21: SCORE THRESHOLD CALCULATION 38

FIGURE 22: READ-GENOME SCORE CALCULATION. THE FINAL ASSIGNED GENOME IS HIGHLIGHTED IN RED. 40

FIGURE 23: SENSITIVITY AND SPECIFICITY FOR SIMULATED DATASETS 52

FIGURE 24: REDUCED REFERENCE COMPARISON 54

FIGURE 25: SENSITIVITY AND SPECIFICITY FOR LOW COVERAGE DATASETS 55

FIGURE 26: COMPARISON OF THE UNIQUENESS OF N-GRAMS IN EACH GROUP. 56

FIGURE 27: SENSITIVITY/SPECIFICITY COMPARISON BETWEEN STRAINIQ AND KRAKENUNIQ AT STRAIN LEVEL AT VARIOUS REFERENCE GENOME QUALITY 58

FIGURE 28: SENSITIVITY/SPECIFICITY COMPARISON BETWEEN STRAINIQ AND KRAKENUNIQ AT VARIOUS COVERAGE. NOTE THAT THE STRAINIQ-SENSITIVITY IS MASKED BY KRAKENUNIQ-SENSITIVITY LINE BECAUSE BOTH ARE AT 100% 59

FIGURE 29: RELATIVE ABUNDANCE COMPARISON. THE FIGURE SHOWS THE DIFFERENCE IN RELATIVE ABUNDANCE PREDICTED BY KRAKENUNIQ AND STRAINIQ AGAINST SIMULATED ABUNDANCE. 61

FIGURE 30: RELATIVE ABUNDANCE COMPARISON BETWEEN STRAINIQ AND KRAKENUNIQ FOR EVEN (A) AND STAGGERED (B) COMMUNITIES 64

List of Tables

TABLE 1: HMP REFERENCE GENOMES 8

TABLE 2: COMPARISON OF DIFFERENT NGS SEQUENCING TECHNOLOGIES USED FOR METAGENOMICS. 11

TABLE 3: POPULAR TOOLS AND METHODS 18

TABLE 4: COMPARISON OF TIME AND MEMORY REQUIREMENT FOR DIFFERENT VALUE OF N 25

TABLE 5: HUFFMAN CODE FOR 2 BASES 29

TABLE 6: HUFFMAN CODE FOR TRIPLETS 29

TABLE 7: SAMPLE SCORES FOR GI TRACT N-GRAMS 34

TABLE 8: IDENTIFICATION SCORE CALCULATION MATRIX 36

TABLE 9: SIMULATED DATASETS 50

TABLE 10: MOCK COMMUNITY SAMPLES 53

TABLE 11: F1 SCORE COMPARISON 57

TABLE 12: NUMBER OF GENOMES WITH BETTER RELATIVE ABUNDANCE 62

TABLE 13: STRAINIQ DEPENDENCIES 67

List of Abbreviations

BLAST	Basic Local Alignment Search Tool
BWA	Burrows-Wheeler Aligner
CPU	Central Processing Unit
DNA	deoxyribonucleic acid
DSEM	DNA Signature Element Model
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GB	Giga bases
GI	Gastrointestinal
HMP	Human Microbiome Project
hrs	Hours
ITS	Internal Transcribed Spacer
LCA	Lowest common ancestor
MG-RAST	Metagenomics Rapid Annotation using Subsystems Technology
MIT	Massachusetts Institute of Technology
NA	Not Applicable
NCBI	National Center for Biotechnology Information
NGS	Next-Generation Sequencing
NIH	National Institute of Health
PGM	Personal Genome Machine
QC	Quality Control
QIIME	Quantitative Insights Into Microbial Ecology
RAST	Rapid Annotation using Subsystems Technology

rDNA	Ribosomal deoxyribonucleic acid
RNA	Ribonucleic acid
SRA	Short Read Archive
StrainIQ	Strain Identification and Quantification
TN	True Negative
TP	True Positive
TPR	True Positive Rate
WGS	Whole Genome Sequencing

Chapter 1: INTRODUCTION

1. Introduction to metagenomics

Microbes are mostly unicellular organisms that are too small to be seen with the naked eye and yet found virtually everywhere in nature. They play extremely important roles in nature as diverse as metabolism in the human body, photosynthesis in plants, or decomposition of waste in the environment. Hence, it is essential to understand how the composition of microbial communities affects the ecosystem of different environments ranging from ocean floors to hot springs to the human body. A great deal of research has shown that a number of human maladies such as obesity [1], gastrointestinal conditions [2], immune deficiency [3] and even mental health [4] are caused by the dysbiosis of microbial communities in the human body. Metagenomics is a growing field focused on the study of these microbial genomes in an environment, such as the human microbiome. It is the study of multiple (Meta) organisms using the genetic material (genomics) obtained directly from the environmental samples. Chen et al. define metagenomics as the application of modern genomics techniques to the study of microbial communities directly in their natural environments, bypassing the need for isolation and lab cultivation of individual organisms [5].

Traditionally, the study of microbes involved culturing individual organisms in labs. Unfortunately, most microbes cannot be cultured in a lab due to our inability to replicate the ideal culture conditions for each species. Metagenomics techniques allow for the sequencing of the entire environmental sample in bulk followed by data analysis to identify and quantify species present in the samples to understand the overall community effect of the microbiome. *Figure 1* shows the steps involved in detail. The samples are collected from an environment (for example gut) followed by DNA

extraction. The DNA is fragmented using shotgun sequencing to obtain numerous small segments of the DNA that can be sequenced using next generation sequencing (NGS) technologies. The sequencer produces reads that need to be partly reassembled at the gene level using bioinformatics data analysis to obtain taxonomic and functional profiles.

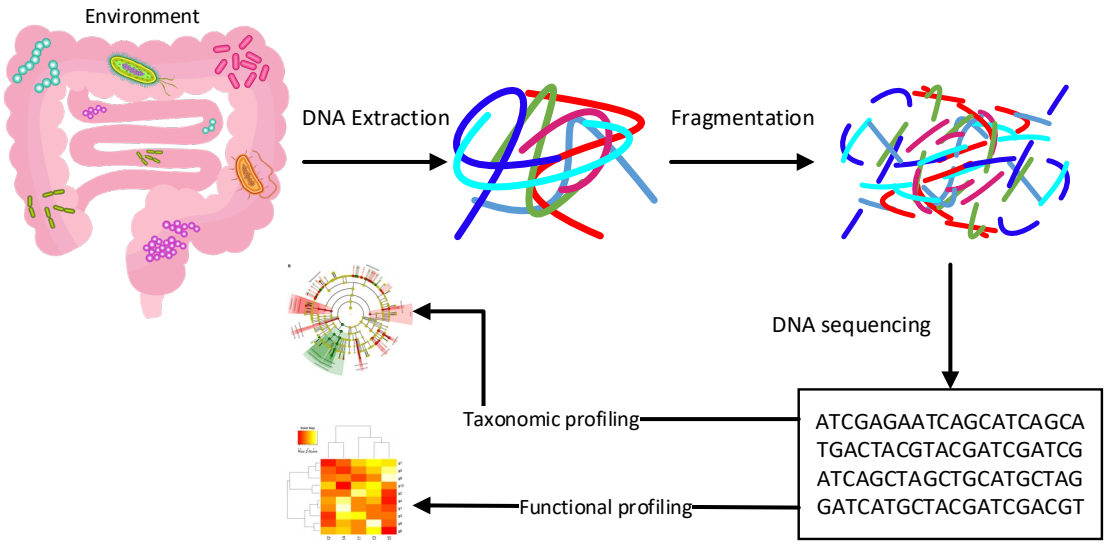

Figure 1: Metagenomics. The figure shows the whole genome shotgun sequencing overview with the gastrointestinal tract as the collection environment.

The microbial community in any environment can be studied by either using WGS or amplicon sequencing of highly variable regions such as the 16S gene.

1.1. Amplicon sequencing

Amplicon sequencing is a highly targeted method that allows researchers to selectively sequence target regions of the genome such as 16S, 18S, ITS, etc. Sequencing hyper-variable regions in bacterial genomes (Figure 2) allows researchers to uniquely identify the taxonomic groups of organisms present in the sample. This approach allows for ultra-deep sequencing of the amplicons which is useful for efficient identification and characterization of taxonomic units in the samples. This method is

useful for sequencing the bacterial 16S rRNA that allows researchers to study phylogeny and taxonomy in an environment. At the same time, this method does not require the investigator to sequence the entire genome to perform functional analysis. Since only the targeted regions are sequenced, amplicon sequencing is much cheaper and quicker.

Figure 2: Bacterial 16S rDNA. The figure shows the hyper-variable regions that vary among different bacteria.

1.2. Whole genome sequencing

Unlike amplicon sequencing, WGS aims to amplify the whole DNA of the metagenome. This method allows us to comprehensively study all genes in all organisms present in the given samples and facilitates taxonomic and functional analyses. Additionally, this method captures viruses and eukaryotes that might be present in the sample. In comparison to amplicon sequencing, this method is more expensive.

2. The Human Microbiome

The human microbiome constitutes all microorganisms living in association with the human body. An average person harbors around 10-100 trillion microbial cells [6]. Different parts of the human body harbor a broad range of environments for microbial communities to grow. Each body site provides a unique ecosystem resulting in a distinct composition of microbes that help drive different biological processes. Alterations in these microbial compositions can perturb biological processes leading to an array of human diseases.

2.1. Significance of human microbiome

The invisible microbes residing on different body parts make up the human microbiome as shown in *Figure 3*. The human microbiome is composed of trillions of microbes that live in and on our bodies. It provides genetic diversity to a host, contributes to the host immunity, impacts the host metabolism and their interaction with drugs.

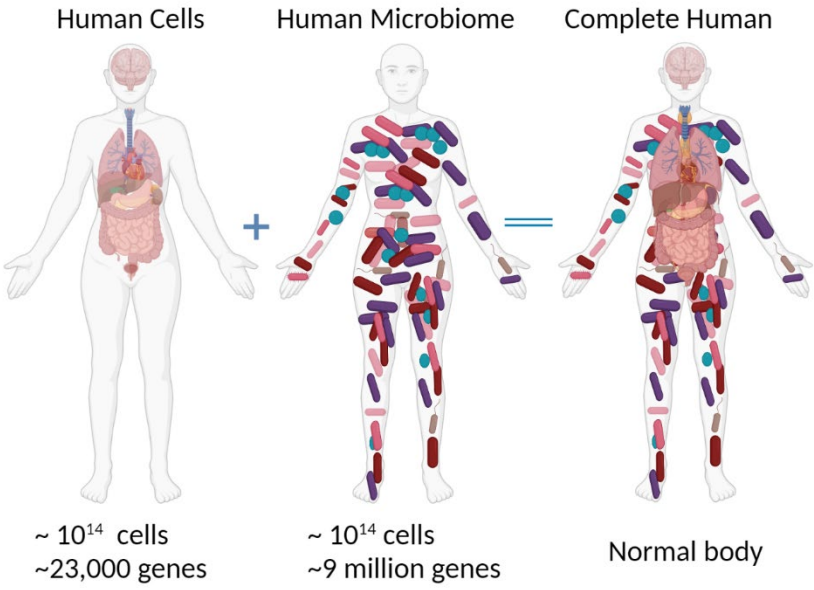

Figure 3: Complete human “genome” composition

Microbial communities present in different sites of the human body are of particular importance due to their implications in the cause and prevention of human diseases [7]. Changes in the composition of microbes are known to have disease-causing effects [8]. Although some individual microbial strains can have drastic effects, it is the community that determines the overall effect on the host’s health [9]. Based on the

effects they have on human health, the microbes in the human body can be categorized as symbiotic or pathogenic.

Symbiotic microbes live in symbiosis with the host. Bifidobacteria, certain strains of *E. coli* and Lactobacilli are some of the microbes that live in symbiosis with human. Bifidobacteria are among the first microbes to colonize the human gut. They have been associated with the production of a number of potentially health promoting metabolites including short chain fatty acids, conjugated linoleic acid and bacteriocins [10]. Some studies have found lower levels of Bifidobacteria linked to higher prevalence of *Staphylococcus aureus* in obese children [11]. Similarly, Bifidobacteria is known to reduce the symptoms of inflammatory bowel diseases [12], maintain remission from ulcerative colitis [12, 13], and to be an effective treatment of diarrhea in infants [14, 15]. Certain strains of *E. coli* are known to prevent *Shigella flexneri* [16] and *Salmonella typhimurium* [17] infection in mice. *E coli* also produces vitamin K [18] and vitamin B12 [19], both of which are beneficial for the host. Lactobacilli are shown to benefit the host by ensuring the lining of the intestines stays intact and producing lactic acid, which may prevent harmful bacteria from colonizing the intestines [20].

Campylobacter jejuni, *Enterococcus faecalis,* and *Clostridium difficile* are among some of the microbes that are pathogenic. Campylobacter species are known to cause foodborne and waterborne infections and are one of the leading cause of bacterial gastrointestinal infections [21]. *Enterococcus faecalis* are responsible for urinary tract infections. Their resistance to most drugs make them incredibly difficult to treat [22]. Enterococci are also responsible for wound and soft tissue infections in hospitals [23]. *C. difficile* is a common cause of nosocomial infection that is responsible for life threatening colitis that can result in death [24]. Figure 4 gives an overview of bacterial infections caused by different microbes in different body parts [25]. *Salmonella typhi* and

Helicobacter spp. are known oncogenic bacteria responsible for gallbladder [26] and liver cancer [27], respectively.

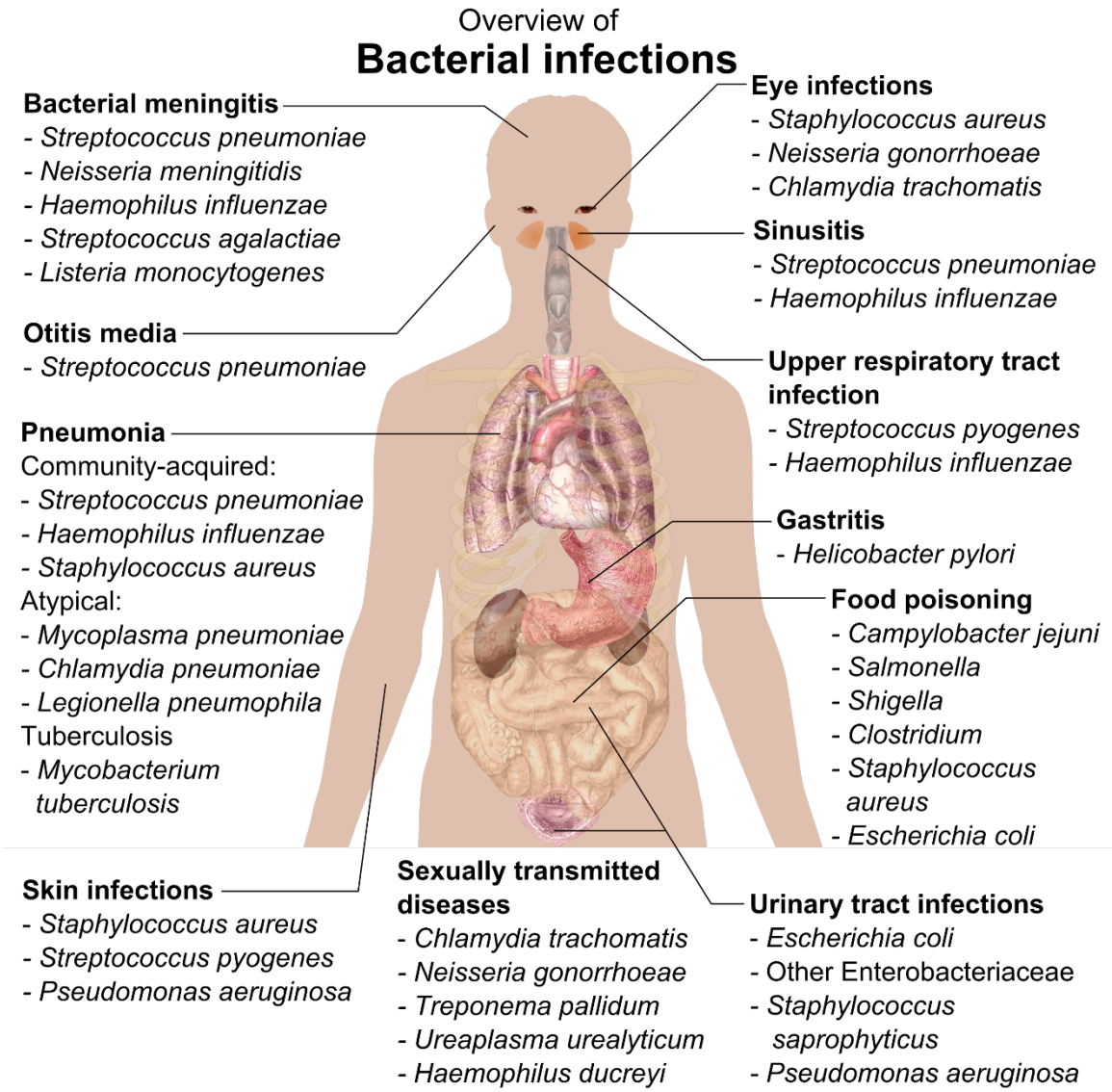

Figure 4: Overview of bacterial infections

A great deal of research has shown that several ailments such as obesity are caused by changes in the diversity of microbial communities in the human gut. Overweight/obese individuals with low fecal diversity are characterized by more marked overall adiposity, impaired glucose homeostasis, dyslipidemia, and more pronounced

inflammatory phenotype when compared to individuals with high bacterial richness [28]. Several mental health illness such as anxiety, depression, and autism have been linked to gut dysbiosis and inflammation [4]. Dysbiosis in gut microbiota has also been linked to carcinogenesis [29, 30].

2.2. The Human Microbiome Project

The human microbiome project (HMP) [31] was funded by the National Institutes of Health (NIH) Common Fund from 2007 through 2016, to characterize the human microbiome and analyze its role in human health and diseases. During the first phase of the project, HMP characterized the microbial communities from 300 healthy individuals across several different sites (*Figure 5*) of the human body: nasal passages, oral cavity, skin, gastrointestinal tract, and urogenital tract.

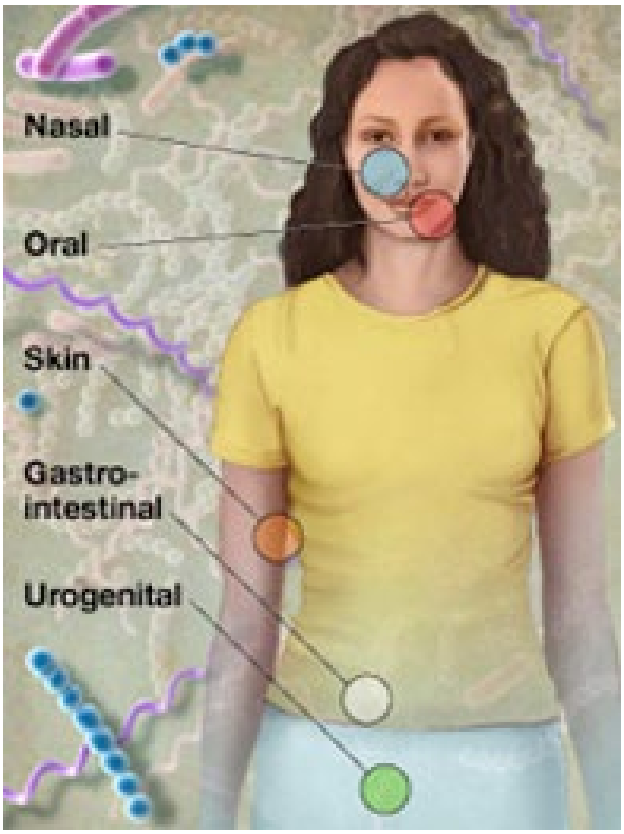

Figure 5: HMP sample collection body sites

The HMP deposits genomic assemblies and other sequences to the NCBI RefSeq database under NCBI Bio Project Accession: PRJNA43021. Reference genomes isolated from the different body sites are stored under the subproject “Human Microbiome Project (HMP) Reference Genomes” with accession PRJNA28331. Table 1 shows the total number of assemblies at different completion levels under different sub-projects. There is a total of 2,947 genomes at different levels of completion, among which only 31 are complete with circular DNA and 14 have partially complete chromosomes. The remaining assemblies are either scaffolds, contigs, or raw reads.

Table 1: HMP reference genomes

Highest level assembly	Number of Projects
Complete genome	31
Chromosomes	14
Scaffolds	1,878
Contigs	386
SRA or Trace	493
No data links	145
Total	2,947

In addition to assemblies, the HMP also provides the body site information for each assembly. *Figure 6* shows different body sites with the total number of assemblies for genomes in each body site.

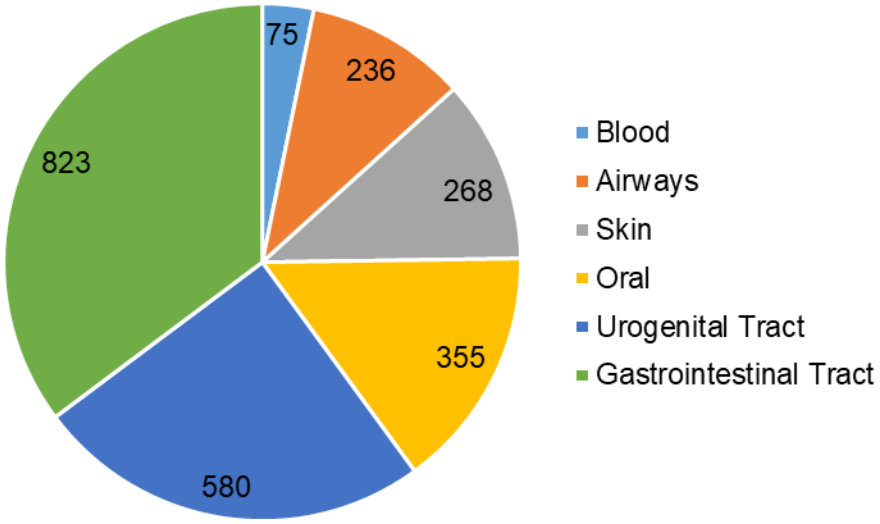

Figure 6: HMP Body sites. The figure shows the distribution of different projects across different body sites as part of the HMP sub-projects.

3. Metagenomics Sequencing and Data Analysis – Big Picture

Metagenomics involves collecting samples from the environment, sequencing them and performing data analysis to identify the taxonomic diversity and estimate the composition (by relative abundance) of the community. *Figure 7* shows the overall workflow of metagenomics sequencing and data analysis [32]. Environmental samples are collected from the environment of interest and the particles are filtered, typically by size. For identifying bacteria in the metagenomic samples, smaller viroid particles and the larger protists are filtered out to enrich the bacterial content in the sample. If needed, computational filtering can be used after sequencing to remove reads belonging to unwanted microbes or host contaminations. The next step is DNA extraction and lysis followed by cloning and library preparation. The library is then sequenced using NGS sequencers such as NextSeq or NovaSeq. The sequencers produce millions of reads

that need to be analyzed using bioinformatics tools to identify and quantify the organisms present in the samples.

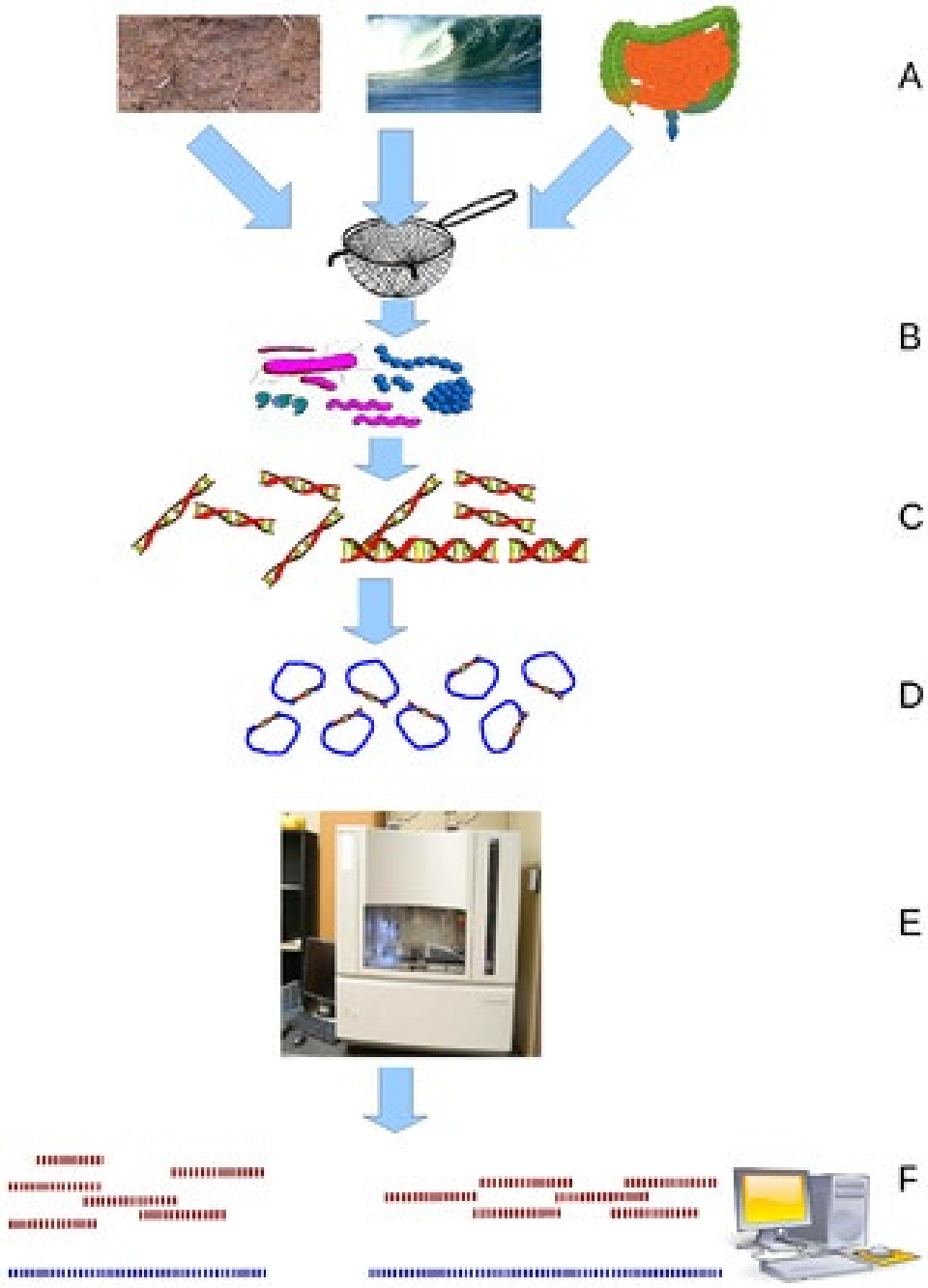

Figure 7: The Big Picture. The figure shows different steps involved in metagenomics experiments. A: Sample collection; B: Filtration; C: DNA extraction and lysis; D: Cloning and library preparation; E: Sequencing

3.1 Use of sequencing technologies in metagenomics

Sanger sequencing played a very important role in the early stages of metagenomics. The concept of using ribosomal RNA genes as molecular markers for the classification proposed by Carl Woese [33] combined with Sanger sequencing allowed researchers to identify individual species present in any metagenomics samples. In recent years, NGS technologies have outperformed Sanger sequencing significantly with its low cost, high yield, and longer sequence read platforms such as Roche 454, IonTorrent PGM, Illumina, and PacBIO RSII. Table 2 shows the comparison of output yield, number of reads, read length, read type, cost per Gb, error type, and error rate among recent NGS technologies which is a big improvement from Sanger sequencing's 96 sequences per run with an average length of 650 bp [34].

Table 2: Comparison of different NGS sequencing technologies used for metagenomics. The cost of sequencing is in USD.

	The Ion Proton	PacBio Sequel System	MiSeq	NextSeq 550 Mid/High-Output	NovaSeq 6000 S1-4
Output per run (Gb)	up-to 15	15 - 100	540Mb - 15 Gb	16.25-120	80-6000
Reads per flowcell	60 to 80 million/Chip	500,000 /SMART Cell	1 - 25 million	130 - 800 million	1.6-10 billion
Maximum read length	200	> 20Kb	2 x 300	2x150	2x250
Read type	SR	SR	SR/PE	SR/PE	SR/PE
Cost/Gb*	~66	~13	~113	~48 - 70	~2 - 48
Error type	indel	indel	substitution	substitution	substitution
Error rate	~1	~13	~0.473	~0.5	~0.1
*The cost is calculated based on the cost of the flowcell/cell/chip and the amount of reads produced by it at maximum capacity.					

The cost calculated in the tables are based on the cost of the flow cell and the amount of reads produced by it at maximum capacity. It does not include the labor cost or other miscellaneous charges. The price will vary significantly depending on the experimental design. Sequencing a 16S sample to generate 300 base pair reads using MiSeq v3 using 600 cycles costs around $88 at University of Nebraska Medical Center NGS Core. Similarly, sequencing a metagenomics sample with paired end 150 base pair

reads using NextSeq mid output, 300 cycle kit costs around $300 per sample. As the sequencing technologies are evolving, the sequencing costs are going down making it more affordable for researchers to use these technologies for metagenomics studies.

4. Summary

Metagenomics is a growing field of study of microbial genomes in an environment, such as the human microbiome. The composition of microbes in different regions of the human body is very distinct, which helps drive different biological processes that are required for human health. Alterations in these microbial compositions could perturb biological processes leading to an array of human diseases. Hence identification and quantification of the microbes present in an environment allow us to better understand the host-pathogen interrelationships and consequent effects on human health. Recent development in NGS technologies has presented us with great opportunities to study and understand the relationship between the host and its microbiome.

Chapter 2: IDENTIFICATION AND QUANTIFICATION OF METAGENOMICS SAMPLES

1. Introduction

Identification and quantification of microbes present in any environment are crucial for investigating natural microbial communities and their significance in the ecosystem. Infections and imbalance of microbial communities in the human body pose a serious health concern. Advances in metagenomics sequencing technologies offer better opportunities to understand the microbial ecology by allowing better identification and quantification of individual microbes.

Next generation sequencers such as NextSeq, MiSeq, and NovaSeq produce a large number of short sequencing reads that need to be processed using bioinformatics tools and algorithms to identify and quantify individual taxa present in the sample. Figure 8 shows the flow chart for the identification and quantification of metagenomics reads obtained from different sequencers.

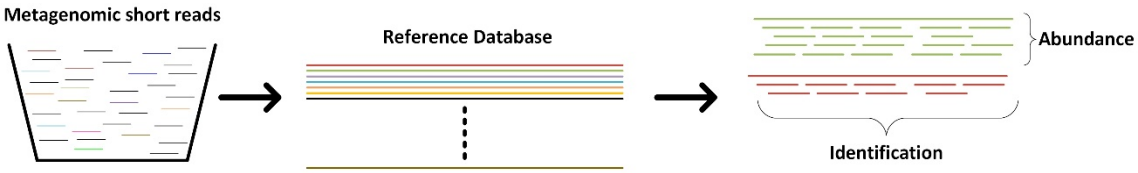

Figure 8: Identification and Quantification

For identifying microbes using the sequencing reads obtained from the sequencers, the reads are compared against a reference database of microbial genomes using various algorithms (such as Bowtie2 [35], BWA [36], BLAST [37], and DIAMOND [38]). The reads are assigned to different organisms based on their match

against the reference genomes. The microbes are then identified based on the reads that are assigned to them. The abundance of each microbe is estimated based on the number of reads assigned to them.

2. Current methods

There are two categories of tools available for taxonomic profiling of metagenomics samples, i.e. “Alignment Based” and “*k*-mer/*n*-gram based”. Alignment based methods use alignment tools such as BLAST and its variants to align the reads obtained from sequencing to the reference genomes followed by statistical analyses to identify the correct assignment of the reads to a taxa. These methods become slow as the volume of the reads increases. On the other hand, alignment-free methods use *n*-gram frequencies or substrings, information theory, graphical representation, or sequence clustering for identification of taxa in a metagenomics sample.

2.1. Alignment-based methods

Alignment-based methods are most widely used across multiple fields including metagenomics. As the name suggests, the reads are aligned directly to the reference genomes using various alignment tools such as BLAST and BWA. The objective of the alignment is to compare two or more sequences to identify similar (homologous) regions. There are two different approaches for sequence alignment: global alignment and local alignment. Global alignment is used to compare the two most similar sequences of approximately the same size, whereas local alignment aligns a short sequence (such as NGS reads) to a substring of a longer sequence (such as the reference genome). Therefore, in metagenomics data analysis, pairwise local alignment is used by the alignment-based methods for determining the most likely origins of the NGS reads by

checking which part of the genome contains the most similar sequences to those reads. *Figure 9*shows a sample alignment of the query sequence against a target sequence. While identifying and quantifying metagenomics samples, the target sequence is one of the reference microbial genomes and the query sequence is a read obtained from a sequencer. The alignment-based methods such as MEGAN[39], MetaPhlAn[40], and MG-RAST[41] process the alignments in different ways to predict the microbes in the sample.

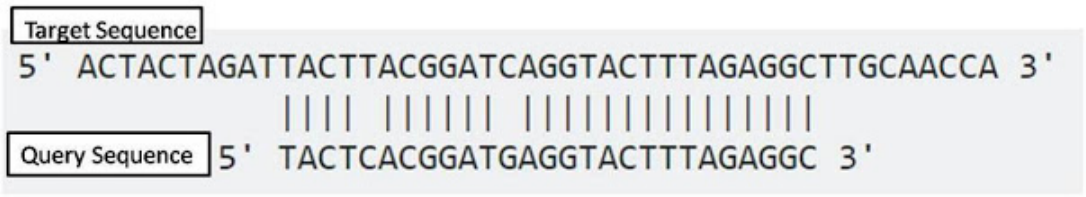

Figure 9: Sequence alignment

The alignment methods compute all possible pairwise comparisons between the query and target sequences. This makes these methods very compute-intensive and very sensitive to gene transfers and sequence lengths. Alignment methods have several drawbacks such as the assumption of collinearity, large memory and computation requirements, and other parameters (e.g., substitution matrices, and gap penalties) used by alignment tools are somewhat arbitrary[42]. In the context of metagenomics, the alignment based methods are also limited by the quality of available draft genomes. Alignment methods have been around for a very long time, so they have well-established algorithms and tools.

2.2. Alignment-free methods

Alignment-free (AF) methods are based on a broad collection of methodologies, including the use of *n*-gram frequencies or substrings, information theory, graphical representation, or sequence clustering. Here we will discuss the methodologies using *n*-grams or *k*-mers. An *n*-gram is a continuous sequence of *n* nucleotides generated from a DNA sequence such as a metagenomic sequencing read. For example “CCGAT” is an *n*-gram of size 5. *Figure 10* shows the generation of overlapping *n*-grams of size 12 from a read of length 31. The first *n*-gram starts at the first base of the read and the second *n*-gram starts from the second base of the read resulting in two overlapping *n*-grams. Each *n*-gram overlaps *n*-1 bases with the preceding and the following *n*-grams.

Figure 10: *n*-grams

The *n*-gram based methods are memory intensive because they require processing of millions of *n*-grams. A read or a sequence of length L generates a total of L - *n* + 1 overlapping *n*-grams. Alignment-free methods work better when the sequences shares low divergence [43]. AF methods are less sensitive to low and moderate frequencies of horizontal gene transfer, and most robust against genome rearrangements[42] because comparisons are made using short pieces of the DNA.

2.3. Popular tools and methods

Over the last several years, many methods have been developed for analyzing metagenomics samples using NGS technologies. *Table 3* lists the most widely used tools for the analysis of metagenomic sequencing data. Tools such as MEGAN [39], MetaPhlAn[40], MG-RAST[41], Qiime[44] and MetaPhyler[45] are alignment based tools. MEGAN is a BLAST-based method that uses the output of BLASTX (translates DNA sequences into amino acid sequences and compares against the protein sequences of the reference genomes) for identifying the taxa and estimating abundance. They recommend using the fast alignment tool, DIAMOND[24] for the alignment to reference genome. MetaPhlAn uses clad-specific markers (from more than 2 million potential markers) as a reference. Similarly, MetaPhyler uses a custom database of 31 phylogenetic marker genes as a taxonomic reference. Similarly, GOTTCHA[46] is a signature based taxonomic profiling method. The reference database is a collection of unique genome segments at multiple taxonomic levels. MG-RAST is an automated pipeline with several steps including quality control, functional and taxonomic annotation, and comparative analysis. The pipeline uses publicly available tools such as FragGeneScan[47], and VSEARCH[48]. MG-RAST is the only tool that provides a web-based interface to perform metagenomic data analysis on a first come first serve basis. However, MG-RAST is able to identify taxa only down to the Genus level and also the wait time for the analysis is unacceptably high. On the other hand, Qiime offers a semi-automatic pipeline that wraps many publicly available software packages together to analyze 16S metagenomic sequences. Kraken[49] is a *k*-mer (or *n*-gram) based alignment-free method that uses LCA mapping to map the k-mers to the reference taxonomy tree. But, Kraken can predict the microbial taxa only down to the Genus level. Due the aforementioned limitations of the current methods such as the dependence on

the quality of reference genomes, the inability to identify taxa beyond Genus or Species level, and the overall accuracy and performance-related issues, we sought to develop a new method to address the majority of these issues.

Table 3: Popular tools and methods

Tool	**Input Data**	**Methodology**	**Highest identification level**
MEGAN	WGS/16S	Alignment-based	Species
MetaPhlAn	WGS	Alignment-based	Species
MG-RAST	WGS/16S	Alignment-based	Genus
QIIME	16S	Alignment-based	Species
MetaPhyler	WGS/16S	Alignment-based	Species
GOTTCHA	WGS	Alignment-based	Species
Kraken	WGS	k-mer-based	Species
KrakenUniq	WGS	k-mer-based	Strain
CLARK	WGS	k-mer-based	Species

3. StrainIQ method Overview

Hypothesis: *N-gram*-based methods are more sensitive for *strain-level identification* even when the complete reference genome is not available for most microbes.

In this project, we developed a novel *n*-gram-based method, StrainIQ (**Strain I**dentification and **Q**uantification), for the identification and quantification of microbial species at different taxonomic levels using whole-genome sequencing (WGS) data from metagenomic samples. StrainIQ takes advantage of the discriminative nature of unique *n*-grams as well as the weighted common *n*-grams present in incomplete and draft

metagenomic assemblies. Additionally, StrainIQ leverages the body site-specific reference genome information to increase the specificity of the prediction.

Figure 11 gives the overview of StrainIQ by depicting different steps involved in the method. The methodology consists of three steps: Model Building (StrainIQ-B), Taxonomic Identification (StrainIQ-I), and Abundance Estimation (StrainIQ-Q). StrainIQ uses the body site-specific reference genomes to create a unique DSEM for each body site. During the DNA Signature Element Model (DSEM) building step, each body site's reference genomes are disassembled to unique overlapping *n*-grams of size *n*. The *n*-grams in each genome are compared against all other genomes in the body site to identify the unique (present only in one genome) and shared (present in multiple genomes) list of *n*-grams. Each *n*-gram is assigned a weight based on the scoring function S_n described in section 4. For testing the method, we selected the body site, gastrointestinal (GI) tract because it contained the highest number of organisms. Similarly, the models can be built for any body-site with at least 50 genomes to obtain a relevant model. For the GI tract, we used 459 draft and complete genomes downloaded from NCBI and 29 mostly complete genomes downloaded from atcc.org mock communities (ATCC® MSA-1006™, ATCC® MSA-1003™) to build the DSEM. During the Identification step, reads from the test datasets were compared against the DSEM to obtain an ordered list of matching genomes in the descending order of the scores. The predicted list of genomes was obtained by selecting the genomes with a score above the cut-off threshold as described in the Methods and Materials section. During the abundance estimation step, the reads in the datasets are assigned to the predicted genomes using the unique and common *n*-grams present in the reads. Each of the Model Building, Identification, and Quantification steps are described in detail in sections 4, 5, and 6.

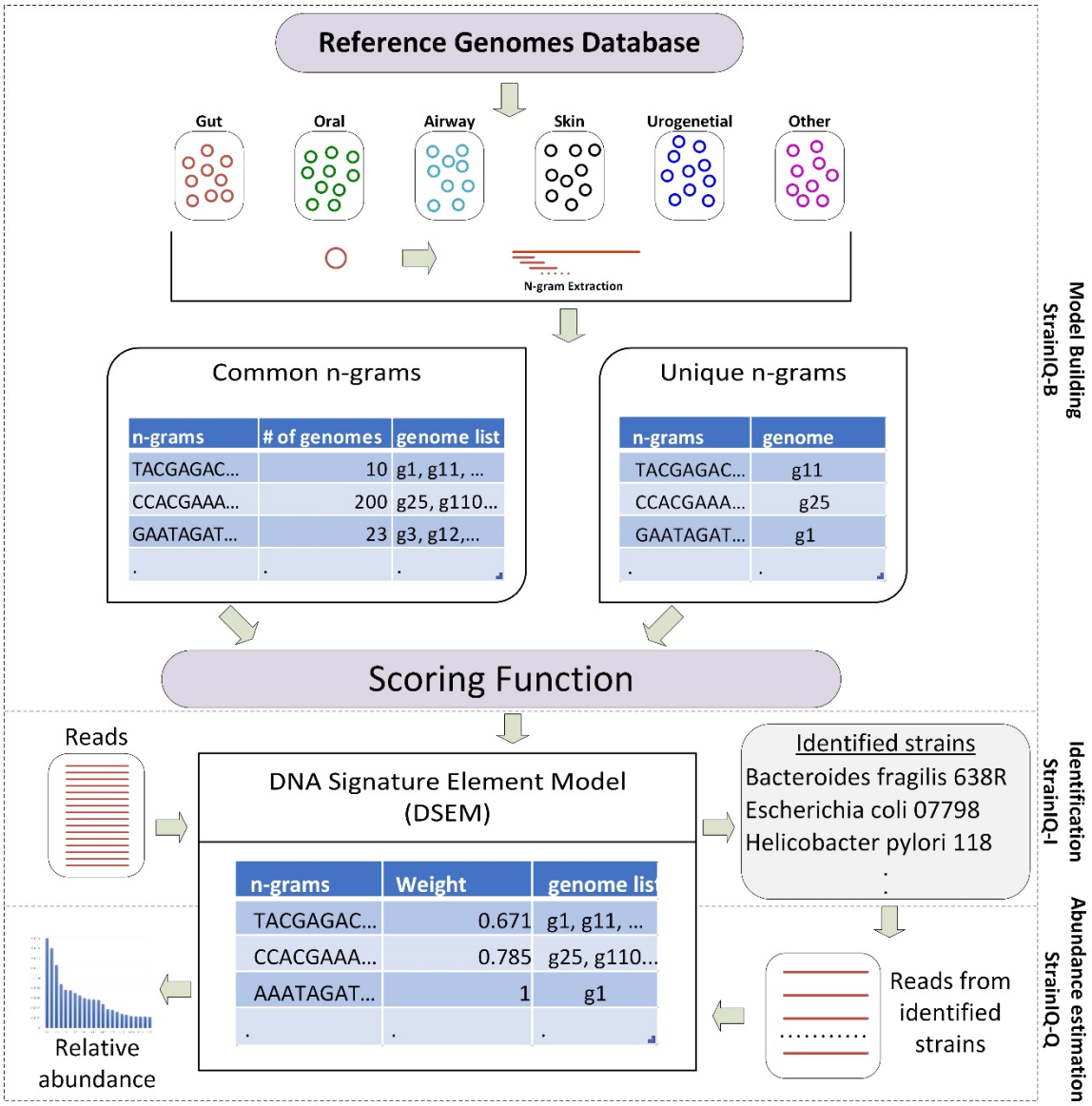

Figure 11: Strain Identification and Quantification - Overview

4. StrainIQ – DNA Signature Element Model Building

Our method uses unique (occurring in only one genome) and common (occurring in more than one genome) *n*-grams as signature elements for identifying taxa at the strain level in the metagenomic samples. We use these *n*-grams as features to build the model that we call DSEM. DSEM building involves generating *n*-grams from the reference genomes and scoring each *n*-gram using a scoring function. The score

represents the discriminatory value of each *n*-gram in the genome. For each genome belonging to a particular body site, we extract unique and common *n*-grams and assign weights to each *n*-gram using the scoring function described below (section 4.2) in such a way that those occurring in fewer genomes have higher weightage and those occurring in more genomes have lower weightage. *Figure 12* shows the different steps involved in building a DSEM for different body sites. The first step involves collecting and separating genomes to different body sites such as Gut, Oral, Airways, Skin, and Urogenital tract. The genomes in each body site are then processed to generate *n*-grams. The figure uses Gut as an example to show the following steps in the DSEM building, which is also applicable to other body sites. Each genome sequence in the GI tract category is disassembled to get common and unique *n*-grams. Appendix 1: GI tract DSEM stats shows the *n*-gram statistics for the genomes in the GI tract. For the gut body site with 471 genomes, we obtained 988,966,457 *n*-grams of which 809,679,392 were unique and 179,187,065 were present in two or more genomes in the gut. The *n*-grams are then scored using the scoring function as described in section 4.2. The bar chart shows the distribution of top scoring *n*-grams in the gut. The larges bar with the height of 809,679,392 and score 1 represents the unique *n*-grams.

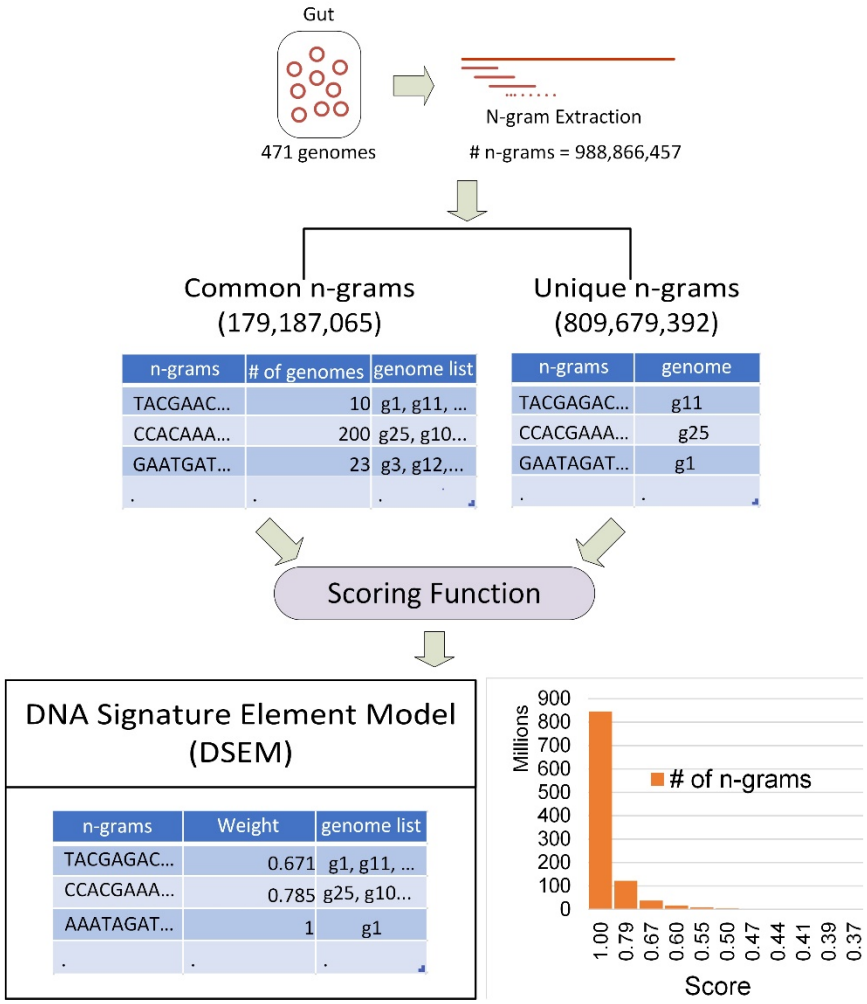

Figure 12: DSEM Building

We can build a separate DSEM for each body site that has at least 50 identified genomes to confer enough discriminatory power to the model. *Figure 12* shows a section of the DSEM. The first column contains encoded *n*-grams, the second column is the list of genomes in which the *n*-gram is present and the third column is the weight for the *n*-gram calculated using the scoring function. We can see that the *n*-grams that are present in only one genome are weighted at 1 (the highest), while those present in multiple genomes (for example, the first *n*-gram in the table that occurs in genomes- 1, 46, 80 and 244- is weighted at 0.6) have lower scores.

n-gram	genome_list	weight
000101101110100001001101100000110001000010	1,46,80,244	0.60
100100001000001100011110101111100011010 01	1	1.00
100101001110111110010110111101111010000000	1,46,245	0.67
000001010010001101100101101001110001100011	1,244,245	0.67
110101100000000010110110100010100011000 1110	1,244	0.79
101101001000001100010110001001100010100010	2	1.00
001100000111101111111011110100000000000001	1,244	0.79
111010000010000100101111011011010001100101	1,245	0.79
111011110100001011110110100101001011011010	1,46,245	0.67
100111100101101001100111011001110011110101	1	1.00

Figure 13: DSEM example

4.1. *n*-gram optimization and encoding

Determining the optimal size of *n*-grams is one of the most important parts of the algorithm. A larger *n*-size generates more unique *n*-grams but also significantly increases the memory and processing time for the tool, whereas a smaller *n* risks losing the discriminatory power to identify the strain-level differences in a metagenomic sample. For standardizing the *n*-gram size, we considered two factors: the number of unique *n*-grams and the total number of *n*-grams in the DSEM. A larger *n*-gram size increases the discriminative power at the expense of memory and execution time. For a nucleotide sequence of length x, the generation of overlapping *n*-grams yields x-*n*+1 *n*-grams, where $n < x$. Only four bases (A, C, G, and T) are allowed to be present in an *n*-gram; *n*-grams containing any other characters are ignored.

Recent computational advancements in processing power and affordable memory allow us to process a large number of *n*-grams more efficiently. We generated all *n*-grams for n=12, 15, 18, 21, 24, and 27 from reference genomes and separated the unique and common *n*-grams. We tried different sizes of *n* with increments of three because the genetic code is a triplet code made of a series of three nucleotides [50]. *Figure 14* shows the comparison of *n*-gram count for various sizes of *n* between 12 and

27 for gastrointestinal genomes. The x-axis shows the size of *n* ranging from 12 to 27. The y-axis shows the total number of *n*-grams. The total number of *n*-grams and the unique *n*-grams increase with the increase in the size of *n* until *n*=21 and start to plateau after that. At *n*=21, approximately 82% of the total *n*-grams were unique, while at n=27, the percentage of unique n-grams increase only to 83%.

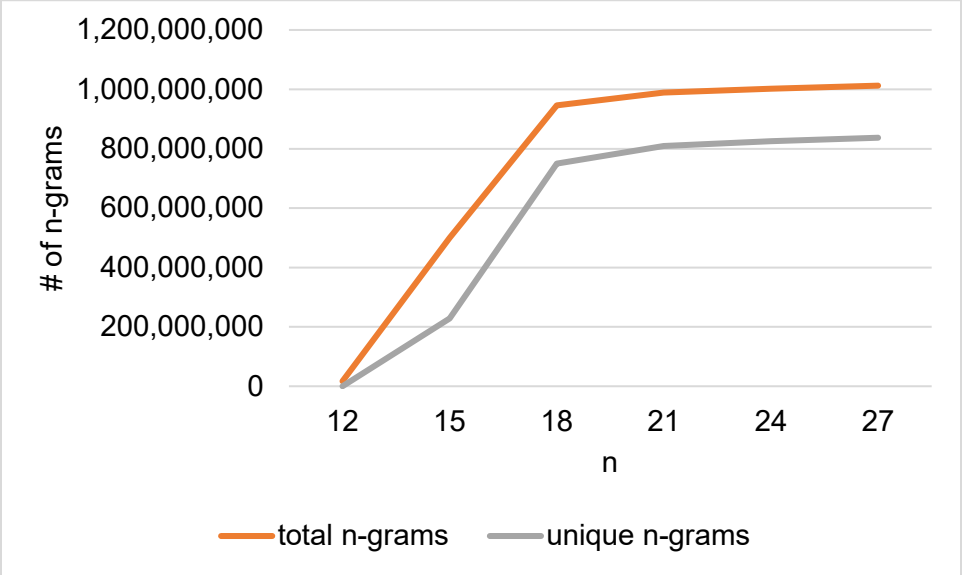

Figure 14: Unique and total *n*-grams count comparison for different n-sizes

Table 4 shows the number of unique or common *n*-grams, memory, and the time required for generating the *n*-grams of different sizes. The "# *n*-grams" column shows the total number of *n*-grams generated for the GI tract for various sizes of *n*. The "# unique *n*-grams" shows the number of *n*-grams that belong to only one genome in the

body site. The “MaxMemory” column shows the maximum memory required. “Runtime(hrs)’ shows the time required for *n*-gram generation for each *n*-gram size.

Table 4: Comparison of time and memory requirement for different value of n

n	# *n*-grams	# unique *n*-grams	Max Memory (GB)	Runtime (hrs)
12	16,768,845	14,933	4.8	1.54
15	500,202,450	227,602,558	32	2.63
18	946,341,199	750,230,389	52	3.08
21	988,866,457	809,679,392	59	3.41
24	1,002,173,023	825,425,923	65	4.07
27	1,012,411,275	837,006,517	71	4.4

The number of unique or common *n*-grams increases with the size of *n*, as shown in *Figure 15* . For *n*=12 (number of unique *n*-grams=21,656), the number of unique *n*-grams is in thousands whereas for higher *n*’s the number of unique *n*-grams is in millions (for *n*= 27, number of unique *n*-grams = 1,012,411,275). The height of the bars is almost constant for *n* ≥21. The line in the figure shows the memory and processing time required to generate the *n*-grams. The memory requirement increases with the size of *n*. The time required to generate the *n*-grams increases with the size of the *n*-grams. For n= 12, 15, 18, 21, 24, and 27 it took 1.54, 2.63, 3.08, 3.41, 4.07, and 4.4 hours, respectively to generate *n*-grams for GI containing 471 genomes. Based on the number of unique *n*-grams, and the memory requirements for different *n*-grams, we choose *n*=21 for generating models for gut genomes.

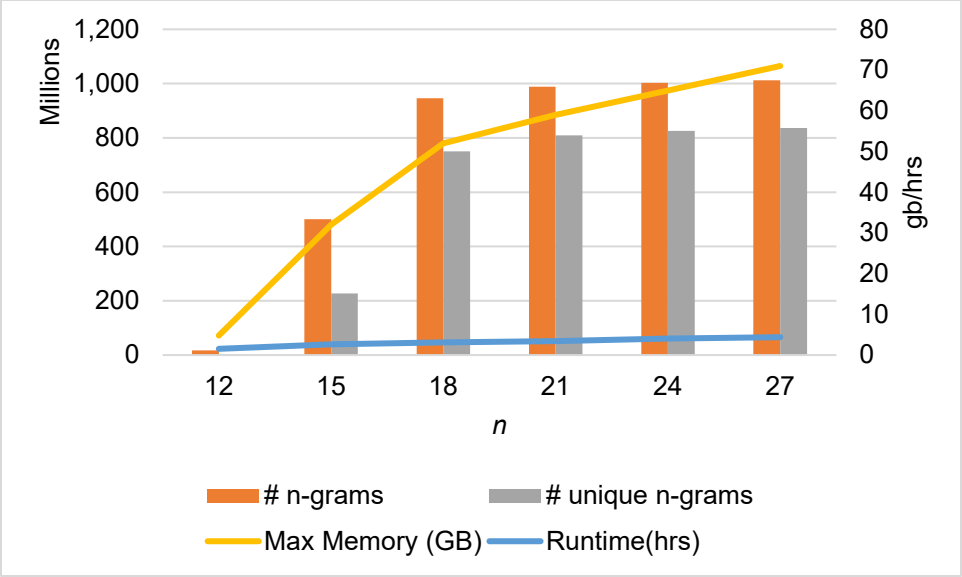

Figure 15: *n*-gram size comparison

4.1.1 **Huffman Encoding**

N-gram based methods need to process billions of *n*-grams depending on the size and number of genomes. Hence, implementation of a data compression algorithm makes the *n*-gram processing more efficient. We encoded the *n*-grams using Huffman encoding [51] to increase efficiency and reduce memory and storage requirements. Huffman encoding is a lossless data compression method. The nucleotide bases in the sequences are assigned variable-length codes based on the frequencies of the corresponding characters. The least frequent nucleotide gets the largest code, and the most frequent nucleotide gets the smallest code. For determining the optimal coding, we counted the frequencies of the four bases (A, C, G, and T) in the reference genomes and calculated the Huffman codes as shown in *Figure 16*. The frequency for A, C, G, and T were 403,456,764, 338915995, 339,021,716, and 404,821,377, respectively. We started with each base and its frequencies as leaf nodes. We then combined the leaf nodes with the least frequencies to create a new internal node of the tree with a new frequency (677,937,711) as the sum of the lowest frequencies. From the remaining two leaf nodes and the new internal node, we selected two nodes with the least frequencies

and added another new internal node with frequency 808,278,141. At this stage, two internal nodes with frequencies 677,937,711 and 808,278,141 remain which we used to create the root node with frequency 1,486,215,852. This completed the Huffman tree.

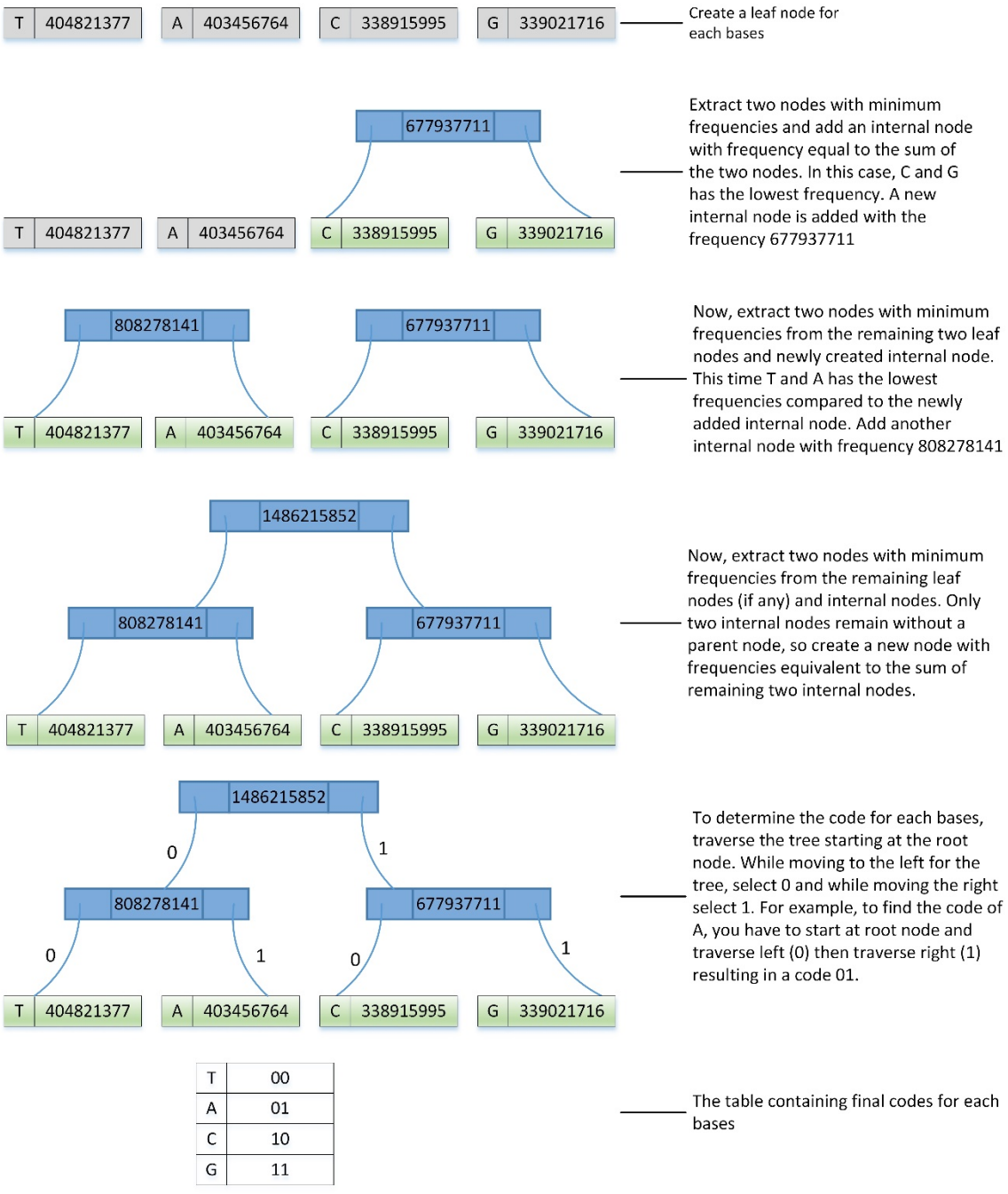

Figure 16: Huffman Tree

To calculate the codes for each of the bases, we traversed the Huffman tree starting at the root node. We built the code by selecting 0 for the left branch and 1 for the right branch of each node. The final table in the figure shows the codes for each of the bases.

For our method with *n*=21, encoding a single base was the best option with each base encoded as A:00. C:01, G:10, and T:11. *Figure 17* shows the advantage of encoding nucleotide bases to save memory and storage requirements. Each nucleotide base represented as A, C, G, and T are stored as equivalent binary in memory occupying 8 bits each. As shown in the binary table in *Figure 17*, A is represented and stored as 01000001 in the memory. By coding A into 00, we only require 2 bits to store A in the memory reducing the storage requirement by 6 bits. In our case, each *n*-gram of size 21 now can be store using just 42 bits instead of 168 bits.

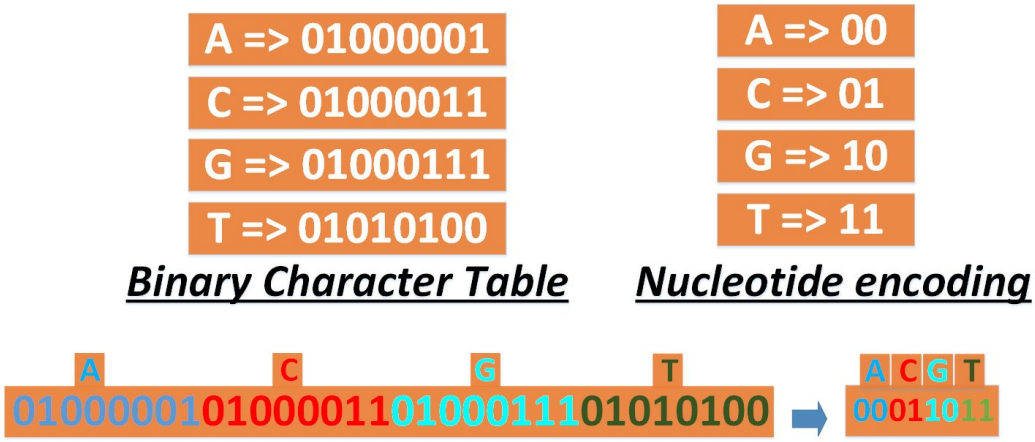

Figure 17: Nucleotide encoding

We created similar Huffman trees using frequencies for duplet and triplet nucleotide codes in the reference genome datasets and generated variable length Huffman codes as shown in Tables 5 and 6, respectively. The encoding generated in

Figure 16 generated the best compression for the *n*-grams, so we choose to encode single nucleotide bases.

Table 5: Huffman code for 2 bases

Symbol	Weight	Huffman Code	Code Length
AA	95648961	1101	4
AC	72689691	0010	4
AG	79766452	0100	4
AT	115439944	1111	4
CA	93731629	1011	4
CC	66365657	0001	4
CG	80955742	0110	4
CT	79970488	0101	4
GA	89330698	1000	4
GC	92344450	1010	4
GG	66314334	0000	4
GT	73115795	0011	4
TA	84824744	0111	4
TC	89626593	1001	4
TG	94075115	1100	4
TT	96132751	1110	4

Table 6: Huffman code for triplets

Symbol	Weight	Huffman Code	code length
AAT	32816556	00100	5
ATT	32837057	00101	5
TTT	30466124	00000	5
AAA	30260519	111111	6
AAC	22170514	011111	6
AAG	26571870	111000	6
ACA	19347662	010110	6
ACC	18225923	010100	6
ACG	17153254	001110	6
ACT	15976252	000100	6
AGA	21378366	011010	6

AGC	22159702	011110	6
AGG	17698902	010000	6
AGT	16020004	000101	6
ATA	26469874	110110	6
ATC	26779370	111010	6
ATG	25472731	101111	6
CAA	26303578	110100	6
CAC	16188636	000110	6
CAG	23541206	101000	6
CAT	25442352	101110	6
CCA	22326053	100001	6
CCG	22409578	100101	6
CCT	17738970	010001	6
CGA	17961312	010010	6
CGC	20629214	011000	6
CGG	22373923	100100	6
CGT	17255399	001111	6
CTG	23574868	101001	6
CTT	26752957	111001	6
GAA	29614444	111100	6
GAC	15509236	000010	6
GAT	26870667	111011	6
GCA	24105720	101011	6
GCC	22437582	100110	6
GCG	20644530	011001	6
GCT	22226570	100000	6
GGA	21646378	011100	6
GGC	22446905	100111	6
GGT	18321113	010101	6
GTA	16983645	001101	6
GTC	15639841	000011	6
GTG	16236010	000111	6
GTT	22347793	100011	6
TAA	25638208	110010	6
TAC	16920238	001100	6
TAT	26435413	110101	6
TCA	25607207	110001	6
TCC	21814778	011101	6
TCG	18013770	010011	6
TCT	21506687	011011	6
TGA	25677018	110011	6

TGC	24181606	101100	6
TGG	22341466	100010	6
TGT	19519162	010111	6
TTA	25599969	110000	6
TTC	29810719	111101	6
TTG	26528795	110111	6
CCC	11793440	1010100	7
CTA	12409649	1011010	7
CTC	15127319	1111101	7
GAG	15072430	1111100	7
GGG	11807483	1010101	7
TAG	12478810	1011011	7

4.2. Scoring function

StrainIQ uses both unique and common *n*-grams to accurately identify and quantify microbes in a metagenomic sample. With *n* = 21, most genomes have at least one unique *n*-gram for most body sites. Figure *18* shows the distribution of unique *n*-grams across all genomes in the GI tract. The number ranges between 85 and 9.15 million with an average of 1,796,099 as represented by the orange line. Unique *n*-grams directly serve as signature sequences for the identification of microbes in a metagenomic sample. However, some organisms have very few unique *n*-grams that can be easily missed during sequencing steps or the differences can be very subtle between different strains of the same species. In addition, abundance estimation requires the ability to assign the short reads to each identified microbes and a number of those reads do not contain unique *n*-grams due to their shorter size. Hence, common *n*-grams are also employed in the scoring algorithm to address these issues. We designed our scoring function to assign weights to both unique and common *n*-grams based on their uniqueness in the genomes of a specific body site.

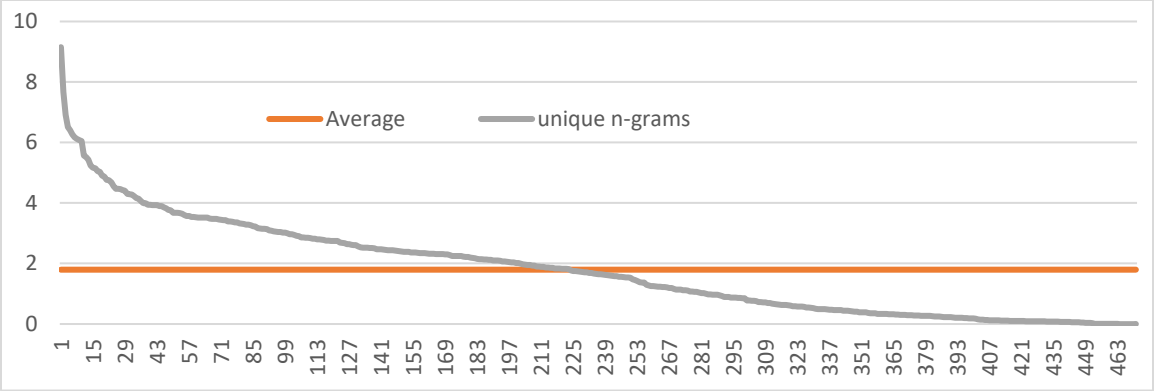

Figure 18: Unique n-grams distribution for gut genomes

The purpose of the scoring function is to assign weights to the *n*-grams based on their discriminatory nature in the corresponding reference genome set. The unique *n*-grams that are specific to only one genome in a body site are separated from the common *n*-grams that occur in more than one genome. The scoring function considers the number of genomes that contain an *n*-gram and assigns appropriate weight to the *n*-gram to reflect its discriminatory nature. The scoring function is similar to the term "weighting" as discussed in our previous study [52]. For any *n*-gram x, the score S_x is given by the expression:

$$Sx = \quad (\log\_e(\frac{|c|}{|c:x\epsilon c|})/\log\_e|c|)^2$$

where c denotes the total number of reference genomes in the DSEM and $c: x\epsilon c$ denotes the total number genomes in which x is present. For a unique *n*-gram, $c: x\epsilon c = 1$; hence $S_x = 1$.

The scoring function can be further optimized to amplify the difference between the scores as the uniqueness of the *n*-grams decreases. This is useful when trying to determine the optimal score for different body sites or environments. As the composition

of the microbiome changes between the body sites, the scoring function needs to be optimized accordingly. *Figure 19* shows the top section of the line chart plotting the scores when the difference in weight between the *n*-grams is amplified either by square or cube as the uniqueness of the *n*-gram decreases. The figure shows the comparison of the speed of score decay when using original score versus squared or cubed scores. The y-axis is the score of the n-grams and the x-axis is the number of genomes in which the n-gram is present. The prediction based on the squared and cubed score were similar. We calculated an average sensitivity and specificity of 86.5%, and 77.3%, respectively, for the squared score and sensitivity and specificity of 85% and 75.4%, respectively, for the cubed scores. We selected square to optimize the original score. For unique *n*-grams, the score is always one and for an *n*-gram that is present in all the genomes in the body site, the score is always zero.

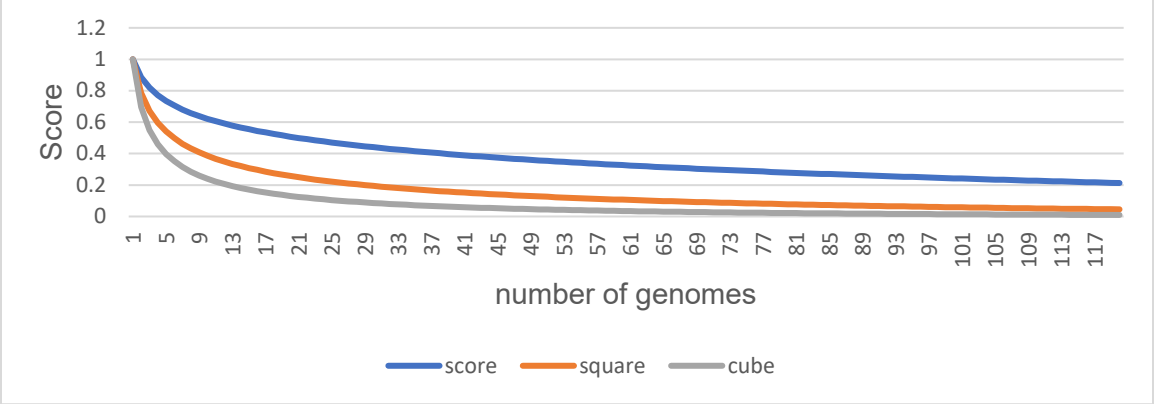

Figure 19: Score optimization comparison.

Table 7 shows the scores for *n*-grams (Sn) based on the degree of uniqueness ($c{:}n\epsilon c$) of the *n*-gram. The score for *n*-gram ranges between 0 and 1, where all unique *n*-grams will receive a score of 1 and those present in all the genomes will receive a score of 0. The square power rapidly dampens the score for *n*-grams that are commonly present in multiple genomes; hence, *n*-grams that occur in fewer genomes receive better

discriminatory scores closer to 1 and vice versa. A genome is predicted to be present in a sample based on the sum of the scores of all the *n*-grams; hence *n*-grams even with smaller scores can still contribute to the decision-making process.

Table 7: Sample scores for GI tract *n*-grams

For GI tract with 438 genomes		
n-gram	weight (Sn)	# of genomes ($c{:}n \epsilon c$)
n1	1	1
n2	0.5407931402	5
n3	0.05897467409	100
n4	0	438

5. StrainIQ – I

The identification step involves performing QC on the metagenomic samples, converting reverse reads if any into a forward direction, generating *n*-grams from the metagenomic reads, comparing them to DSEMs, and determining which taxa they belong to. *Figure 20* gives the overall workflow of the identification step.

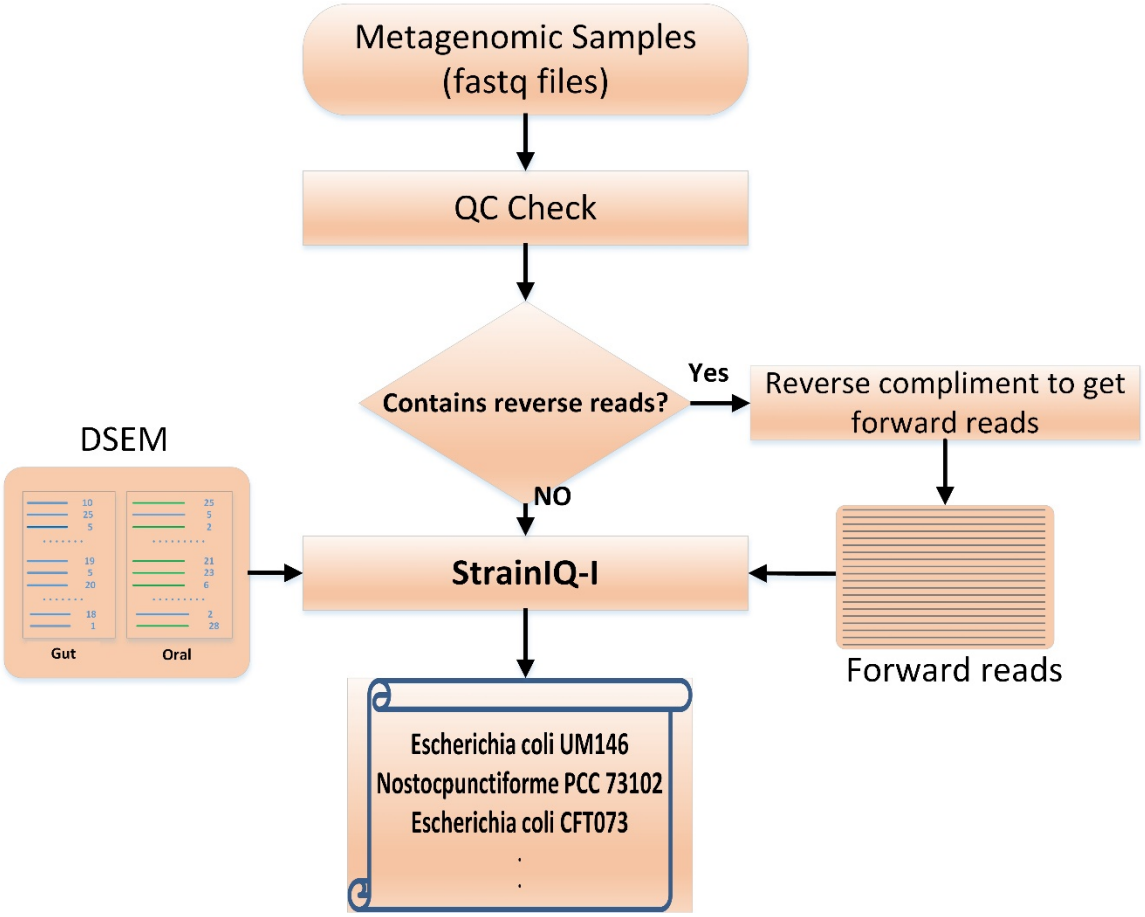

Figure 20: Identification workflow

After preprocessing the reads, StrainIQ-I deconstructs the reads into unique overlapping *n*-grams and identifies their scores based on the DSEM. We build a matrix with genomes as columns and the *n*-grams as rows and fill each cell in the matrix with the scores of the *n*-grams. For N = {n_1, n_2, n_3 … , n_x} where N is a set of *n*-grams generated from a metagenomic sequencing read and G = {g_1, g_2, g_3 … , g_y} where G is a set of reference genomes in a body site, $W_{x,y} = \begin{bmatrix} w_{\mathrm{n1,g1}} & \cdots & w_{\mathrm{n1,gy}} \\ \vdots & \ddots & \vdots \\ w_{\mathrm{nx,g1}} & \cdots & w_{\mathrm{nx,gy}} \end{bmatrix}$. The summed column score gives the initial probability of the presence of a taxon in the metagenomic sample for each reference genome in the matrix: $S_{gi} = \sum_{j=1}^{x} w_{nj,gi}$, where x is the total number of *n*-grams in the metagenome, and i is the total number of genomes in the body

site. *Table 8* shows a sample of the preliminary identification score calculation matrix. The rows are unique *n*-grams generated from the input reads. The columns are the genomes in the DSEM. Each cell is populated with either the *n*-gram weight if the *n*-gram for the row is present in the corresponding genome in the column or 0 if the *n*-gram is not present in the genome. The last row shows the column sum (S_{gi}) for each genome in the DSEM which will be further normalized to calculate the actual identification score.

Table 8: Identification score calculation matrix

genomes → *n*-grams↓	$\mathbf{g_1}$	$\mathbf{g_2}$	$\mathbf{g_3}$	..	$\mathbf{g_j}$
$\mathbf{n_1}$	$W_{n1,g1}$ = 0.67	0	0.67	..	0.67
$\mathbf{n_2}$	1	0	0	..	0
$\mathbf{n_3}$	0.78	0.78	0	..	0
..	..	..	..	..	..
$\mathbf{n_i}$	0.6	0.6	0	..	0
$\mathbf{S_{gi}}$=	3098056.571	2540015.916	1796622.858	1804972.262	1565461.202

The genome scores need to be normalized to minimize the bias caused by the advantages larger genomes have over smaller genomes due to the overwhelmingly large difference in the number of score contributing *n*-grams. Appendix 1: GI tract DSEM stats have the *n*-gram counts for each of the genome in the GI tract. We can see that the largest genome has almost sixty times the number of *n*-gram in them compared to the smallest one. So, we normalized the genome scores using a parameter called '*n*Factor' that considers the size of the genome and the number of *n*-grams contributing to the prediction score. *n*Factor is defined as: $n\text{Factor}_g = n_c/n_t$, where n_c is the number of score-contributing *n*-grams and n_t is the total number of *n*-grams in the genome. The *n*Factor adjusts the raw genome scores in such a way that genomes with more discriminatory *n*-

grams - irrespective of their genome size – score higher than those with less discriminatory *n*-grams. The final probability score can be calculated as $fS_{gi} = S_{gi} * nFactor_g$. A score threshold (cutoff value) is calculated for each body site based on the genomes present as described in section 5.1. Any genome with a score equal to the cutoff or above is considered as present in the metagenomic sample.

5.1. Score threshold determination

Because the number and diversity of microbial taxa vary for each body site, the distribution of unique and common *n*-grams follows suit warranting the need to determine a site-specific score threshold to distinguish true positives from false positives and true negatives from false negatives. We calculate the threshold by determining a value below which any genome can score simply by a random chance. For this, we calculated the scores for all genomes in a body site (positive sets) and compared them against the scores of different sets of genomes belonging to other body sites (negative sets). We used the scores of negative datasets to determine the score cut-off value for predicting genomes present in the metagenome, as described in our previous study [53].

While our methodology is generic and can be optimized to work with all body sites with at least 50 genomes, we used the example of the GI tract for testing purposes as it contains the highest number and most diverse set of taxa. For this experiment, we generated *n*-grams for all the genomes in the gut as the positive dataset. We randomly selected three sets of genomes from other body sites as negative datasets. We plotted the normalized genome score distribution of positive and negative datasets to determine the cut-off value for genomes that could be used for identifying a taxon in the metagenome. In an ideal scenario, we expect the maximum score for any genome in the negative datasets to be less than the minimum score of the genomes in the positive

dataset. But there are plenty of *n*-grams of size 21 that can occur in both negative and positive datasets resulting in the cases where the genomes in the negative datasets get significant scores, some exceeding those of the genomes in the positive dataset. We plotted the score distributions of genomes from positive and negative datasets in descending order for the positive dataset and ascending order for the negative datasets as shown in *Figure 21*.

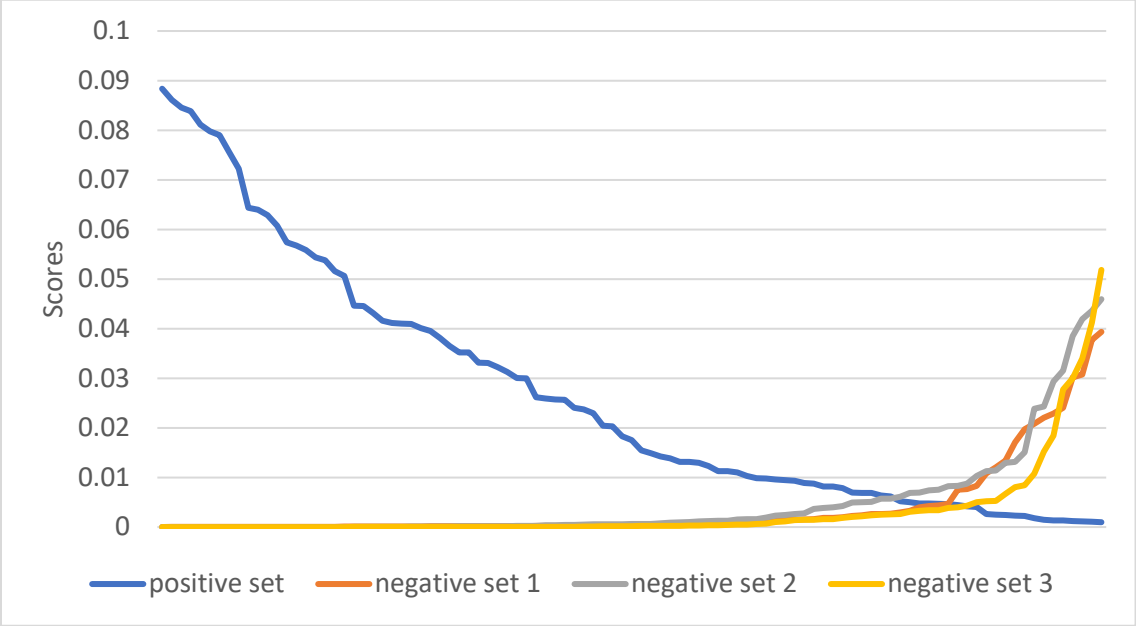

Figure 21: Score threshold calculation

The figure shows the section of the plot where the scores for the positive and negative datasets intersect. The score distribution in the negative datasets before the intersection is represented by *n*-grams that are less discriminatory and those beyond the intersection are more discriminatory than that of the positive dataset. In other words, the intersection is the score threshold where *n*-grams from the positive dataset have higher discriminatory power than those in the negative dataset to accurately identify the taxa. The values beyond the intersection indicate the scores that any random genome can have because of the common *n*-grams. We optimized the cut-off value further to obtain

optimal sensitivity and specificity using ten sets of simulated datasets. We determined the optimal cut-off to be $3.16E^{-9}$ for the GI tract.

6. StrainIQ – Q

Relative abundance is calculated by assigning the reads to identified genomes based on the *n*-grams present in the reads. With *n*=21, a significant number of *n*-grams were unique to single genomes making the read-to-genome assignment process rather simple. First, we assigned all the reads containing at least one unique *n*-gram to corresponding genomes. Then, we calculated the read-genome score for reads containing only non-unique *n*-grams using a read-genome matrix with reads as rows and genomes as columns. The read-genome score for a read and the genome is the sum of the weights of the *n*-grams that are common between a genome and a read. For N = {n_1, n_2, n_3 … , n_x} where N is a set of *n*-grams present in reads R = {R_1, R_2, ….R_i} containing only non-unique *n*-grams, the probability of R_i belonging to genome g_y is calculate as: $R_{ig} = \sum_{j=1}^{x} S_{njgy}$. The read R_i is assigned to the genome g_y with a maximum R_{ig} score. Each read is assigned to only one genome that has the highest read-genome score. *Figure 22* gives an overview of the read assignment process. In the figure, A is the DSEM and B is the matrix storing the probability score for a read and genome. The assignment column is populated at the end of the read processing step. C is the read (R1) that is being processed. Even though we selected all overlapping *n*-grams from each read, the algorithm also allows for selecting *n*-grams based on a window size to make the quantification process run faster. Selecting the larger window size produces fewer n-grams, which speeds up the process but it may result in a situation where the read scores are not distinguishable between two or more genomes.

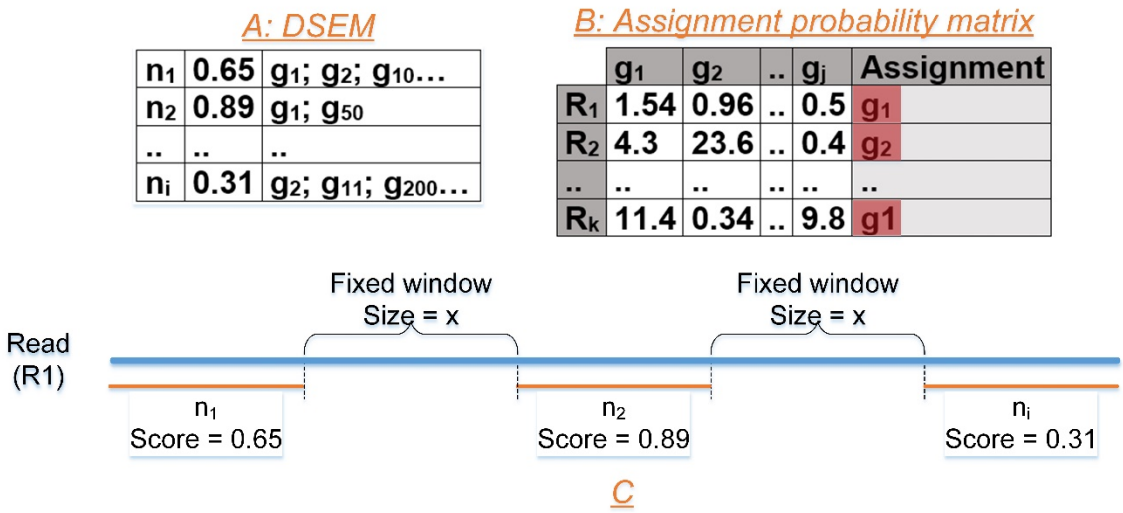

Figure 22: Read-genome score calculation. The final assigned genome is highlighted in Red.

For each *n*-gram (n1, n2,...ni) in the read R, we calculate the probability score S_{Rg_j} as

$$S_{Rg_j} = \sum_{n \in g_j} S_n$$

where S_n is the score for the *n*-gram in the DSEM (A). After calculating the score for each of the $R_i g_j$ pair in the assignment probability matrix (B), the read is assigned to the genome with the highest score (S_{Rg_j}).

7. Computational complexity

The execution of StrainIQ involves DSEM building, identification, and quantification. The most time-consuming part of the entire process is DSEM building, which is done only once for each body site unless there is a change in the known composition of the body site. During DSEM building, StrainIQ calculates scores for all the *n*-grams present in the genomes in two major steps. First, it generates the entire set

of overlapping unique *n*-grams of length 21 (default) from the nucleotide sequences across different genomes. Secondly, it compares the *n*-grams in a genome against those from all the other genomes in the body site to generate a comprehensive list of unique and common *n*-grams for the body site. For *n*-gram generation from *k* number of genomes with *l* as the length of the longest genome, the worst run-time complexity is *O(knl).* For comparing the *n*-grams across genomes in the body site, if the largest genome has α number of non-repeatable *n*-grams then the worst time complexity for model building can be given as O(nk(l + α)).

For identifying the genomes in the metagenomic reads, StrainIQ compares the *n*-grams obtained from the reads against the common and unique *n*-grams in the DSEM and creates a matrix of size δxN where δ is the number of genomes and N is the total number of unique *n*-grams in the reads. For generating the *n*-grams from τ number of reads with l as the length of the longest read, the worst run-time complexity can be given as O(τnl). The time complexity for comparing β number of *n*-grams is O(βn). Once a matching *n*-gram is found in the DSEM, the row entries are updated across all the δ number of genomes either with the weight of the *n*-gram or with a 0. The worst time-complexity for updating the row and column entries and for obtaining the entire column sum is O(2βδ). At the same time, the worst time complexity for determining the largest column sum out of all the δ columns is O(δ). Therefore, the worst time-complexity for this step is O(τnl) +O(δβ((n+2)+ 1/β)).

For estimating the abundances StrainIQ filters the DSEM to remove genomes not identified by StrainIQ-Identifier. For this, StrainIQ creates a subset of DSEM containing only the genomes identified by StrainIQ by comparing the identified genomes against the DSEM. For x identified genomes and η entries in the DSEM, the worst run-time complexity to create a subset of the DSEM is O(xη). Each read is then processed to

generate *n*-grams and compare it against the smaller DSEM resulting in the same complexity as the identification step.

8. Conclusions

In this chapter, we discussed the advantages and disadvantages of alignment-based methods for the identification and quantification of metagenomic samples. We also reviewed alignment-based and alignment-free metagenomics tools publicly available for research. We discussed in detail, the advantage of *n*-gram based methods in the context of StrainIQ. We also discussed in detail the algorithm used by StrainIQ for accurately identifying and quantifying metagenomic samples. We optimized the *n*-gram size based on the optimal use of memory and runtime and the scoring function for the best performance of the method using the data from the gut body site, which is described in the next section. We selected $n = 21$ as default size for n and a threshold score of $3.16E^{-9}$ for prediction cut-off.

Chapter 3: PERFORMANCE TESTING AND OPTIMIZATION

1. Introduction

To test the accuracy of StrainIQ, we tested the method using simulated and mock communities. The best way to verify the accuracy of a prediction method is to use the method to predict the taxa for a known set of communities and compare the prediction against the expected community/composition.

To test the performance of StrainIQ, we used several simulated datasets. Simulation allows us to create metagenomic samples with known microbes. We can also control the abundance of the microbes in a simulated dataset. This is the best way to test a tool during preliminary evaluation and iteratively optimize its performance. Unfortunately, simulation does not perfectly represent real-world environments. During experiments (library preparation, sequencing) different artifacts are introduced to the samples that need to be addressed by any prediction method. We used mock communities sequenced locally to address this issue.

Mock community samples are a genomic mix of known microbes with known composition. ATCC [54] sells mock communities from various body sites that can be processed locally and sequenced to obtain samples containing known microbes with known abundances. These mock communities, when prepared and sequenced locally, contain regular experimental artifacts that we would expect in real-world samples.

We also ran other popular methods MetaPhlAn [40], CLARK [55], and KrakenUniq [56] using these simulated and mock communities. We compared the StrainIQ prediction against these methods using Sensitivity, Specificity, and F1 Score.

2. Materials and Methods

2.1. Datasets

2.1.1. Reference datasets

For building body site specific DSEMs we used the body site information from Human Microbiome Project (HMP). We downloaded the reference genomes for the Human Microbiome Project (HMP) BioProject (NCBI BioProject Accession: PRJNA43021) from the NCBI website. We downloaded 2,234 genome assemblies from the NCBI BioProject database in September 2020. These genomes were isolated from various body sites, including the gastrointestinal tract (GI), airways, oral cavity, skin, and urogenital tracts. We downloaded the body site information for each assembly from the HMP portal.

Around 50% of bacteria found in the wild contain one or more plasmids [57]. Plasmids are independent, mostly circular self-replicating DNA (occasionally RNA) molecules a bacterial genome. Most of the time plasmids are removed from the bacterial genome but found that sixteen of the downloaded genome assemblies had fasta sequences of plasmids in them. We parsed the assembly files to remove the plasmids before generating models.

2.1.2. Test datasets

We used simulated datasets to represent diverse communities and compositional variances for testing and comparing different tools. We used InSilicoSeq [58] to simulate metagenomes with 20 million 150 bp paired-end reads from 200 - 300 randomly selected reference genomes from the GI tract using the NovaSeq error model. InSilicoSeq provides a flag to use draft genomes for simulation, which allowed us to use all draft genomes for creating test samples. We used InSilicoSeq to simulate the datasets using the following parameters:

```
> iss generate –draft <list of genomes> --cpus 50 --n_reads 20M --model
novaseq –output <output_file_name>
```

We also used Gut Microbiome Genomic Mix (ATCC® MSA-1006™) containing 12 evenly mixed gut genomes, and Staggered Mix Genomic Material (ATCC® MSA-1003) containing 20 staggered mix genomes from ATCC (https://www.atcc.org/) for experimental validation. We prepared three replicates for even and staggered mix mock communities to avoid obvious variations caused during the experiment. Since most of the strains included in these mock samples are complete and not present in NCBI databases, we obtained the complete genomes from ATCC and updated our DSEM before testing.

2.2. Detailed analysis pipelines

We compared StrainIQ results against MetaPhlAn, CLARK, and KrakenUniq. In the section, we will describe the analysis steps and pipelines for each of the methods in detail.

StrainIQ: Here we describe StrainIQ pipeline in detail. StrainIQ has three main parts: Builder, Identifier, and Quantifier.

Step 0: StrainIQ Builder: StrainIQ-Builder generates a DSEM for a body site. This is run only once for each body site at the front end of the process and repeated as and when the genomes need to be updated in a DSEM. It takes *n*-size and the list of genomes in a body site as input. The –glist is a tab-delimited file with user-assigned genome-id and genome file location. Along with the output DSEM, the builder also creates a configuration file for use with identification and quantification steps.

```
#Build DSEM for a list of genomes in the genome list file (glist)
python StrainIQ.py
    -p builder # Use builder program to build DSEM
    -n 21 # Default n-gram size for GI.
    -glist <genome list> # genome file and location.
```

Step 1: StrainIQ Identifier: StrainIQ-Identifier takes –dsem and sample name as input to identify the microbes in the given sample. The identifier refers to the configuration shown below for the additional parameters. The configuration file is generated as a part of the DSEM building and has the same name as DSEM with the .conf extension. I addition to DSEM and configuration file, the identifier also refers to the Map file and Taxonomy file shown below.

```
#Configuration file
n=21 #n size
number of bacteria=488 #number of microbes in the DSEM
cutoff=3.16E-90 #Default cutoff value for the body site
```

```
 #Map file: Tab delimited file with gid, strain and unique n-grams count
genomeID        RefSeqassemblyaccession uNgrams strain_tax_id
1       GCF_000146285.1 2366754 585198
2       GCF_000243215.1 2366754 742817
3       GCF_000144025.1 2366754 765115
4       GCF_000160135.1 2366754 608534
5       GCF_000148285.1 2366754 749521
```

```

#Taxonomy file: Tab delimited file with gid and taxonomy mapping
gid     RefSeqassemblyaccession strain  phylum  class   order   family  genus   species Organism.Name
1       GCF_000146285.1 585198  1       1       1       1       1       1       Enterococcus faecalis TX4244 1
2       GCF_000243215.1 742817  1       1       1       1       1       1       Enterococcus faecalis TX4244 2
3       GCF_000144025.1 765115  1       1       1       1       1       1       Enterococcus faecalis TX4244 3
4       GCF_000160135.1 608534  1       1       1       1       1       1       Enterococcus faecalis TX4244 4
5       GCF_000148285.1 608534  1       1       1       1       1       1       Enterococcus faecalis TX4244 5
```

```
#Identify taxa using the DSEM for the body site
python StrainIQ.py
    -p identifier # Use identifier program for identification
    -dsem gi.dsem # Choose appropriate DSEM for the body site
    -sample sample1.fastq #Provide sample in fastq format
```

Step 2: StrainIQ Quantifier: StrainIQ-Quantifier takes –dsem, -sample and –prediction file as input to calculate the abundance of microbes in the metagenomic sample.

```
#Quantify taxa based on the identified genomes
python StrainIQ.py
    -p quantifier #Use quantifier program for quantification
    -dsem gi.dsem #Choose appropriate DSEM for the body site
    -sample sample1.fastq # Provide sample in fastq format
    -prediction sample1.prediction # provide identified genomes
```

MetaPhlAn: We ran MetaPhlAn (3.0) with default parameters. We used the reference database mpa_v30_CHOCOPhlAn_201901 supplied as a part of the tool. We installed the database using --install parameter. We ran MetaPhlAn using the same parameters and database for both simulated and experimental datasets.

```
#MetaPhlAn database install
metaphlan
    --install
    --index mpa_v30_CHOCOPhlAn_201901
    --bowtie2db <database folder>
#MetaphlAn identification and quantification
metaphlan
    set1.fa_R1.fastq, set1.fa_R2.fastq
    --input_type fastq
 -s sams/set1.sam.bz2
 --bowtie2db metaphlan_databases
 --bowtie2out metagenome.bowtie2.bz2
 --nproc 10
 -o set1_profiled_metagenome.oOption.out
```

CLARK: We ran CLARK (v1.2.6.1) with default parameters for each sample. We created a custom database for CLARK with genomes from the GI tract. We used

set_targets.sh script to create a custom database. We ran CLARK against all simulated and experimental datasets using the custom database.

```
#Create custom database
set_targets.sh DIR_DB custom
#Run identification/prediction
CLARK
    -P set2.fa_R1.fastq set2.fa_R2.fastq
    -R set2_results.txt
    -n 50
    -D DIR_DB
    -T targets.txt
#Run abundance
getAbundance
    -D DIR_DB
    -F set1_results.txt.csv > set1_abundance.log
```

KrakenUniq: We ran KrakenUniq (v 0.5.8) using default parameters. We built a custom database for reference using the genomes from the GI tract. For building a custom database, we formatted the reference genomes to add taxid to the fasta header in the genome files. We added the reformatted genomes to the library using –add-to-library option. We created a new database using the new library. We used the same parameters and database for both simulated and experimental datasets.

```
#Format new genome files to add taxid to fasta header
>kraken:taxid|7000787823 f5bcb58692924cb7_1
    f5bcb58692924cb7_1 assembly_id="f5bcb58692924cb7"
    genome_id="d1ef0271f5b14846" atcc_catalog_number="ATCC
    12228" species="Staphylococcus epidermidis"
    contig_number="1" topology="circular"
ATGTCAGAGAAAGAAATTTGGGATAAAGTTTTAGAAATTGCCCAGGAAAGAATTTCAA
    AC
ACTAGTTATCAAACGTTCATAAAAGATACGCAACTCTACTCACTTAAAAATGACGAAG
    CC
```

```
#Add custom file list to library, run –add-to-library for
          each new genome.
krakenuniq-build
      --add-to-library
          kraken_formated_genomes/GCF_000010385.1_ASM1038v1
          _genomic.fa
```

```
#Add the genomes in the library to the database
krakenuniq-build
    --db gi
    --kmer-len 31
    --threads 50
    --taxids-for-genomes
          --taxids-for-sequences
       #Run classification using the custom database.
       krakenuniq
           --threads 10
           --db db/gi
           --paired
           --report set1_kraken_report.txt
           --unclassified-out unclassified_seqs_set1#.fa
       set1/set1.fa_R1.fastq set1/set1.fa_R2.fastq >
       set1ss_READCLASSIFICATION.tsv &
```

3. Statistical Measurements

We used several statistical measurements to evaluate the performance of StrainIQ and used other methods for comparison. As we know the composition of both the simulated and mock communities, we were able to calculate the sensitivity and specificity metrics. We also used the F1 score where possible to understand the real difference when sensitivity and specificity were not enough. Each parameter is described in the context of the project below.

TP: True Positive. Number of microbes correctly identified as present in a sample.

TN: True Negative. Number of microbes correctly identified as not present in a sample.

FP/ type I error: Number of microbes incorrectly identified as present in a sample.

FN/ type II error: Number of microbes incorrectly identified as not present in a sample.

Sensitivity/recall/TPR: Sensitivity refers to the tool's ability to correctly identify the microbes present in a sample. It is calculated as follows:

$$Sensitivity = \frac{TP}{TP + FN}$$

Specificity/TNR: Specificity refers to the method's ability to correctly identify the microbes not present in the sample. It is calculated as follows:

$$Specificity = \frac{TN}{TN + FP}$$

F1 Score/F-measure: F1 score is the harmonic mean of precision and recall. It measures the overall accuracy of the method which makes it ideal for the cases where sensitivity and specificity are not enough to correctly distinguish the merits of the methods. It is calculated as follows:

$$F1\ Score = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

4. Results

Here we present the results of StrainIQ predication on simulated and mock community. We will also show the comparison of these results against MetaPhlAn, CLARK, and KrakenUniq.

4.1. StrainIQ prediction based on simulated datasets

We simulated ten sets of test datasets generated from InSilicoSeq[58] using randomly selected genomes from reference genomes in the GI tract. InSilicoSeq takes complete, draft or incomplete draft genomes and simulates reads like those from Illumina sequencing. We selected the NovaSeq error model and the draft genomes option to generate 150bp paired-end reads for each set. The detailed statistics for the simulated sets are shown in Table 9.

Table 9: Simulated datasets

Datasets	# of reads	# of genomes	error model	base pairs
set1	19,991,006	300	NovaSeq	150
set2	19,990,266	300	NovaSeq	150
set3	19,991,472	300	NovaSeq	150

set4	9,993,568	200	NovaSeq	150
set5	9,993,308	200	NovaSeq	150
set6	9,993,846	200	NovaSeq	150
set7	9,994,270	200	NovaSeq	150
set8	9,992,948	200	NovaSeq	150
set9	9,993,326	200	NovaSeq	150
set10	9,994,538	200	NovaSeq	150

We used StrainIQ to identify the taxa present in each of the ten sets. Our method was able to identify taxa at the strain-level in the simulated datasets at an average of 86.72% sensitivity and 75.15% specificity. *Figure 23* shows the sensitivity and specificity for each of the ten sets. Set1, Set2, and Set3 were simulated using 300 genomes from the GI tract and Sets 4 through 10 were simulated using 200 genomes from the GI tract. We see that the specificity for the samples containing a larger number of reads (20 million) is lower than the specificity for the samples containing fewer reads (10 million). At the same time, samples with a larger number of reads are more sensitive compared to the samples containing fewer reads. The average sensitivity for samples containing 20 million reads is 90.75% whereas the average sensitivity for samples containing 10 million reads is 84.98%. Similarly, the average specificity for samples containing 20 million reads is 67.64% whereas the average specificity for samples containing 10 million reads is 78.36%. We calculated the F1 score for all samples to better understand the results for varying coverage. The average F1 score for samples with 20 million reads is 0.873 whereas the average F1 score for the samples with 10 million reads is 0.798.

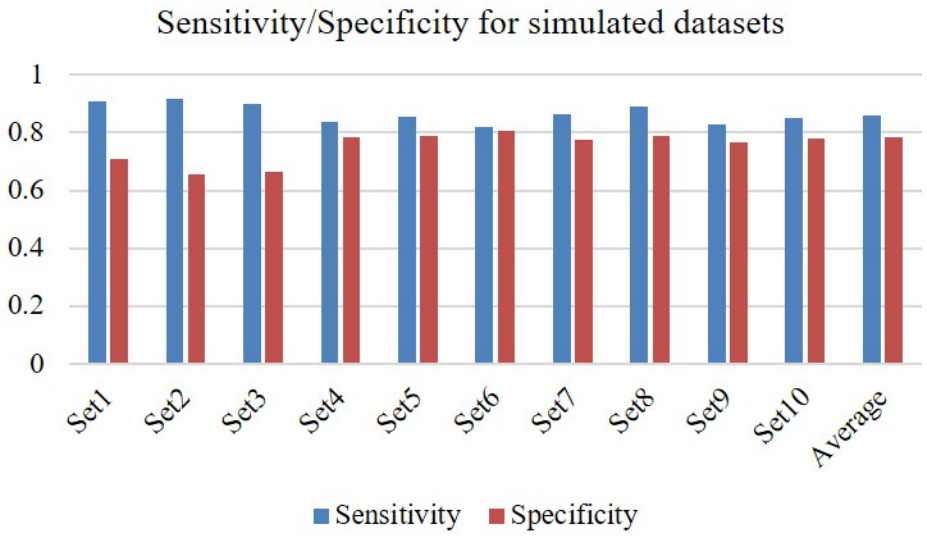

Figure 23: Sensitivity and Specificity for simulated datasets

4.2. StrainIQ prediction based on experimental datasets

Simulation cannot artificially create artifacts that are normally introduced during different phases of experiments such as library preparation and sequencing. We acquired mock communities containing known quantities of microbes from ATCC [54]. We sequenced the mock communities containing even and staggered mix genomes (ATCC® MSA-1006™, ATCC® MSA-1003™) at the Genomics Core facility at University of Nebraska Medical Center. This allows us to test the strength of the tool with known standards to identify the taxa in the samples with even (all organisms have equal relative abundance) and staggered (unequal relative abundance) communities. The staggered mix allows us to explore the performance of the tool in identifying both high and low abundant genomes in the samples. Three replicates for each of the mock communities were sequenced on the NextSeq550 to generate 150bp paired-end reads. Table 10 show the three replicates sequenced for even and staggered mock communities with the total number of reads in each replicate. The read count ranges between 34.9 million to 42.9 million.

Table 10: Mock community samples

Sample Name	Sample Type	Read Count
MSA-1003_1	Staggered	34,981,088
MSA-1003_2	Staggered	42,954,950
MSA-1003_3	Staggered	39,670,082
MSA-1006_1	Even	37,831,270
MSA-1006_2	Even	38,208,832
MSA-1006_3	Even	36,099,148

Since the genomes in the mock communities are not available on the NCBI database, we downloaded the reference genomes of the microbes in mock community from the ATCC website and updated the DSEM to include these new genomes. Appendix 1: GI tract DSEM stats shows the *n*-gram statistics after adding the new mock genomes. We noticed that the addition of the new genomes significantly reduces the number of unique *n*-grams in the DSEM.

Gut microbial reference genomes are mostly incomplete and low quality because of which reference-based analyses can be tricky and susceptible to errors. On the other hand, both the mock communities we used contain genomes that are complete and are of high quality. This allowed us to artificially reduce the quality of the reference genomes to simulate incomplete genomes. We tested StrainIQ using 100%, 75%, 50%, and 25% of the randomly selected regions from the reference genomes to test the ability of the tool to correctly identify the taxa at the strain-level even when the reference genomes are incomplete. We built additional DSEMs using 75%, 50%, and 25% of the reference genomes for the reduced reference tests. *Figure 24* shows the specificity and sensitivity of StrainIQ for all four cases for even and staggered mock communities. The performance of StrainIQ didn't change significantly with incomplete reference genomes. Staggered samples have the lowest sensitivity of 80% for reference genome reduced to 25% but the specificity is steady at 90%. Unlike even mix samples which have even

relative abundance of 0.083 per genome, staggered mix samples are a mixture of genomes with variable relative abundance ranging from 0.0002 to 0.179712. This possibly contributed to erratic change in sensitivity for the staggered samples.

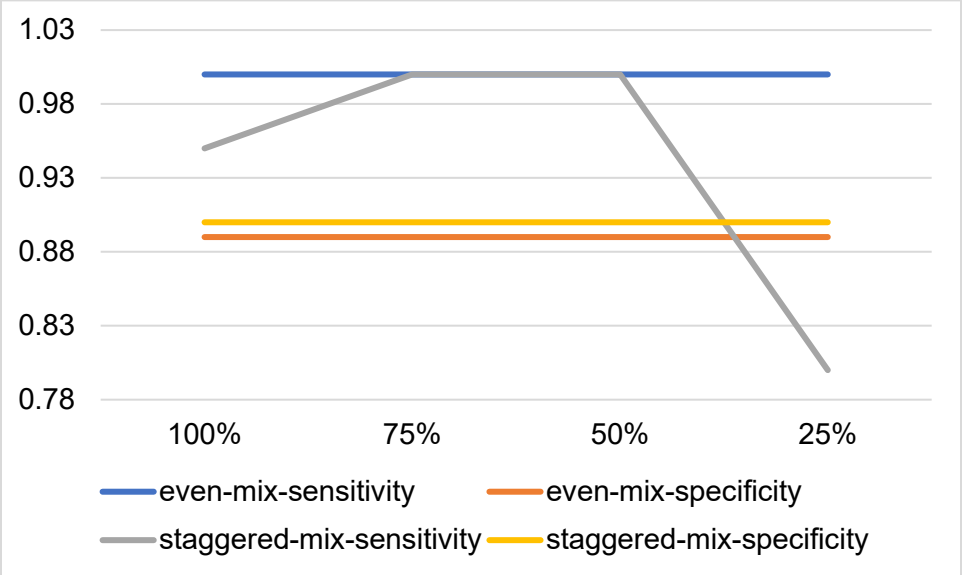

Figure 24: Reduced reference comparison

This shows the strength of our method to accurately identify strains in a metagenomics sample even when the reference genomes are draft assemblies. We also noted that the identification algorithm could accurately identify strains with similar specificity for both even and staggered mixed samples. Real-world microbial samples usually contain some genomes at a much lower abundance level. Based on our results, StrainIQ was able to predict genomes in those samples accurately.

We also tested StrainIQ against datasets with varying coverage. The even mock samples have 12 genomes, and the staggered mock samples have 20 genomes in the genomic mix. There are fewer genomes in each sample than what we normally expect in GI tract samples. This resulted in the generation of sequencing reads equivalent to nearly 120X coverage from the sequencer. This allowed us to test our method at varying coverage ranging from 120X to 1X. We created three sets of data using only 25%, 50%, and 75% of the total reads resulting in new datasets corresponding to 30X, 60X, and

90X coverage, respectively. To test for the lower-than-normal coverage, we created additional three sets of data to represent only 1x, 3x, and 5x coverage. *Figure 25* shows the sensitivity and specificity of StrainIQ for samples with varying coverage. The sensitivity does not change between different coverage levels. The specificity also remains constant across 120x, 90X, 60X, and 30X samples. But the specificity starts increasing at 5X coverage and continues to increase as the coverage decreases. As the coverage decreases, the false positives caused by repeating common *n*-grams at higher coverage levels start to decrease. This helps improve the specificity at lower coverages. Although not seen in this case, this can also reduce the sensitivity when the genomes present in the samples rely mostly on common *n*-grams for identification.

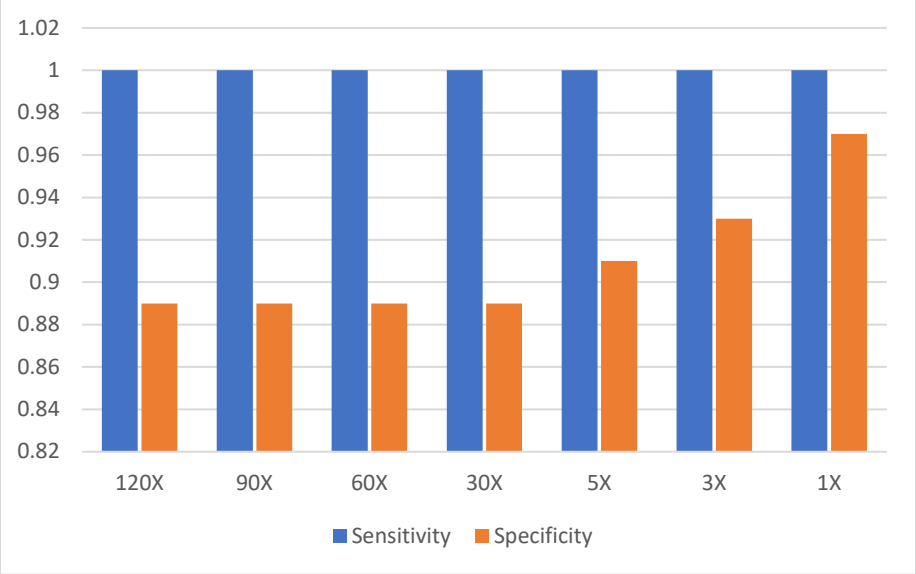

Figure 25: Sensitivity and Specificity for low coverage datasets

We analyzed the *n*-grams present in all the groups to calculate the ratio of common *n*-grams to unique *n*-grams to determine how the uniqueness of the *n*-grams in the samples varies when the number of reads decreases. As expected, we observed that the ratio of the number of common *n*-grams to the number of unique *n*-grams

increases from 5X coverage to 1X coverage. This is reflected in the specificity increase shown in *Figure 25*. In a situation where reads don't contain at least one unique *n*-gram, StrainIQ relies on the common *n*-grams to identify the genomes present in the sample. During these situations, these common *n*-grams help StrainIQ maintain high sensitivity. At the same time, common *n*-grams are also responsible for false positives. There can be situations where a microbe is not present in the sample but simply has enough common *n*-grams to get falsely identified. As the number of reads decreased in the samples with only 5X and lower coverage, the effect of common *n*-grams decreases as well hence increasing the specificity of the tool.

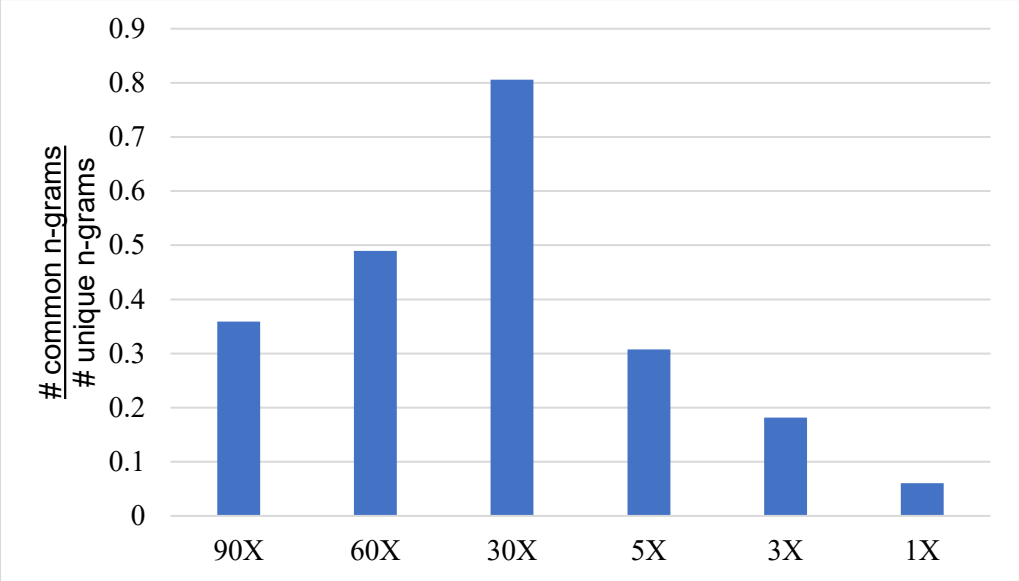

Figure 26: Comparison of the uniqueness of *n*-grams in each group.

5. Comparison against other popular methods

5.1. Identification

We compared StrainIQ against other popular tools for metagenomics analysis using simulated datasets. We used three simulated samples containing 300 genomes and 20 million 150bp NovaSeq reads. We used MetaPhlAn, CLARK, and KrakenUniq to

identify taxa at different levels using the pipeline detailed in section 2.2 in Chapter 3. We calculated the average F1 score from the three samples for each tool as shown in Table 11. The table lists F1 scores for different methods at strain, species, and genus levels.

Table 11: F1 Score comparison

	Genus	Species	Strain
StrainIQ	0.977053995	0.8861729	0.820636267
KrakenUniq	0.982779232	0.9420496	0.639081251
MetaPhlAn	0.913947366	0.7190503	NA
CLARK	0.887157548	0.7193917	NA

StrainIQ's performance was superior to both CLARK[55] and MetaPhlAn[40] at the genus level and on par with KrakenUniq with the F1 score of 0.977. CLARK's F1 score is 0.887. CLARK produces a significant number of false positives compared to other methods resulting in a very low specificity of a mere 3.5% as shown in Appendix 3. At the species level, StrainIQ performs better than CLARK and MetaPhlAn but underperforms compared to KrakenUniq. StrainIQ has better specificity than CLARK and is more sensitive than MetaPhlan at the genus and species level as shown in Appendix 3. KrakenUniq [56] performs the best at the genus level compared to all other methods with an F1 score of 0.942. Similarly, the species-level F1 score for KrakenUniq is 0.942. Most notably, StrainIQ outperforms KrakenUniq at the strain level with an F1 score of 0.82 as against 0.639 for KrakenUniq. StrainIQ makes use of complete overlapping *n*-grams from the reference genomes allowing it to accurately identify taxa at higher resolution. This is probably because KrakenUniq uses the classification of Kraken [49] at higher resolution (Species) to identify strains, while StrainIQ focuses on identifying strains first and builds up to calculate higher taxa making it more accurate at strain level predictions.

We also used mock microbial communities to compare StrainIQ against KrakenUniq at the strain level. To investigate the effects of incomplete reference

genomes, we also ran KrakenUniq with reduced sets of reference genomes (75%, 50%, and 25%). *Figure 27* shows the comparison of specificity and sensitivity between StrainIQ and KrakenUniq with the original as well as the reduced reference genomes. The performance of StrainIQ was unaffected by the reduced reference genomes. This algorithm maintained the specificity at 90% for all sets of reduced reference genomes. On the other hand, the specificity of KrakenUniq decreased for incomplete reference genomes. KrakenUniq has the highest specificity of 74% for 100% for the reference genome and the lowest specificity of 61% for 25% of the reference genome. The sensitivity and specificity for StrainIQ were 100% and 89%, respectively for the test run with 25% of the reference genome. The sensitivity and specificity for KrakenUniq were 100% and 61%, respectively for the test run with 25% of the reference genome. Our experiments confirmed that StrainIQ does better than other popular algorithms for incomplete draft genomes.

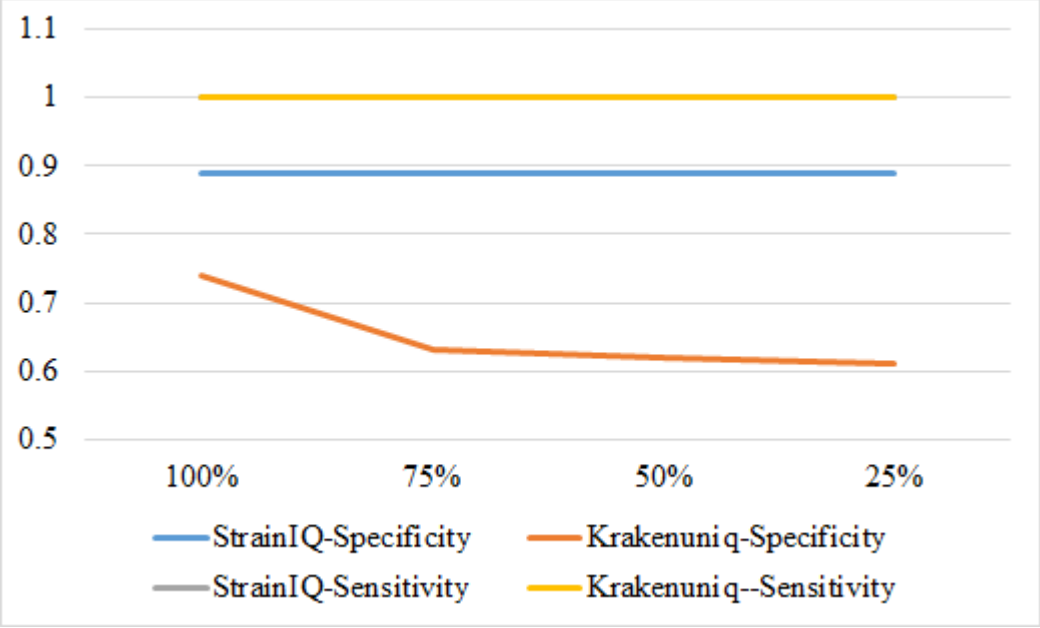

Figure 27: Sensitivity/Specificity comparison between StrainIQ and KrakenUniq at strain level at various reference genome quality

We also tested the performance of KrakenUniq using samples with variable coverage and compared it with StrainIQ. We ran KrakenUniq on samples with coverage ranging from 120X to 1X and compared the sensitivity and specificity against StrainIQ as shown in *Figure 28*. We found that the sensitivity remains consistent for both StrainIQ and KrakenUniq, but the specificity decreases for KrakenUniq to 61% for 30X datasets. StrainIQ has better specificity at every coverage level in comparison to KrakenUniq. Specificity for KrakenUniq follows the same trend as StrainIQ for the coverage ranging from 5x to 1x. As described earlier, this is probably caused by the decrease in common *n*-grams responsible for higher false-positive rates KrakenUniq is also an *n*-gram based method that makes it susceptible to the effects of common *n*-grams.

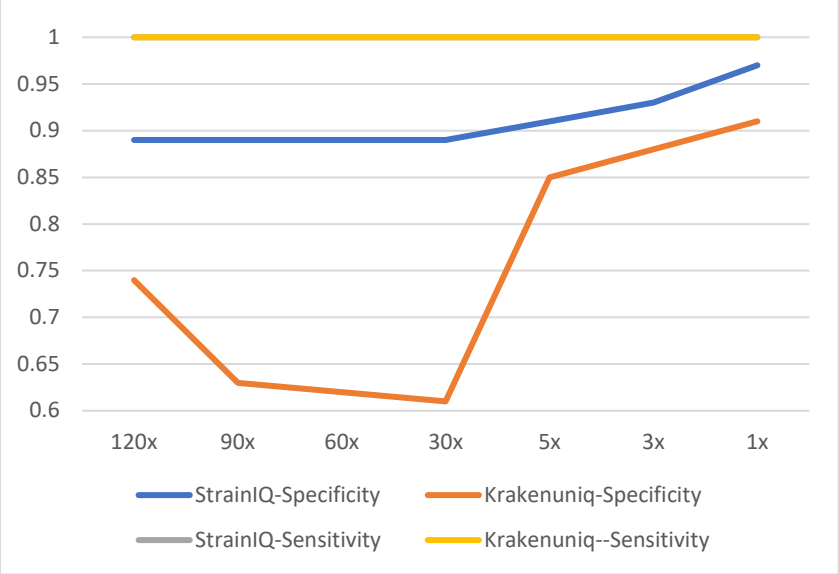

Figure 28: Sensitivity/Specificity comparison between StrainIQ and KrakenUniq at various coverage. Note that the StrainIQ-sensitivity is masked by Krakenuniq-sensitivity line because both are at 100%

5.2. Quantification

After identification of taxa, StrainIQ also calculates the relative abundance of the organisms present in the metagenomic sample by assigning the reads to corresponding

taxa. We tested the performance of StrainIQ using simulated as well as experimental samples.

We calculated the relative abundance for all the ten datasets used for validation as described before. We compared the relative abundance performance for StrainIQ and KrakenUniq. To make sense of the comparison, we calculated the difference in the relative abundance predicted by each of the methods against simulated abundance for all of the predicted genomes.

$$diff - StrainIQ = abs(SimAbundance - StrainIQAbundance)$$

$$diff - KrakenUniq = abs(SimAbundance - KrakenUniqAbundance)$$

We ignored the genomes that were not identified by either of the methods. *Figure 29* shows a section of the plot comparing the difference between StrainIQ prediction and simulated abundance, and the difference between KrakenUniq prediction and simulated abundance represented by the orange and the grey lines, respectively. The closer the lines are to x-axis (0), the better is the prediction for the method.

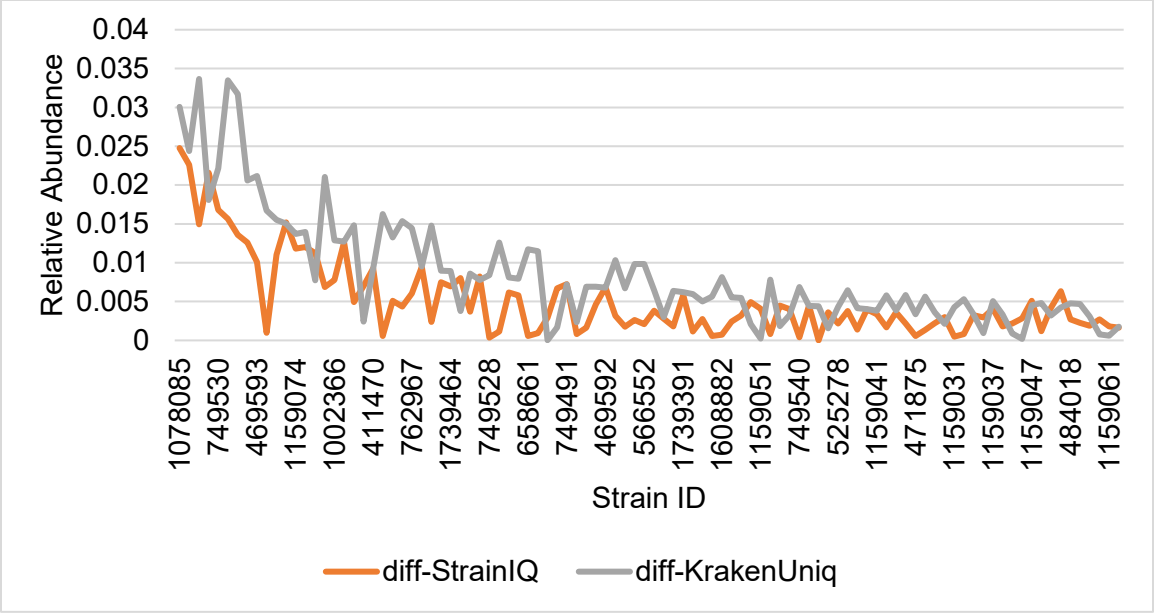

Figure 29: Relative abundance comparison. The figure shows the difference in relative abundance predicted by KrakenUniq and StrainIQ against simulated abundance.

Based on the difference between the predicted and simulated relative abundance values, we determined the number of genomes each method predicted better for all ten sets. Table 12 lists the number of genomes with better relative abundance for StrainIQ and KrakenUniq. The first column shows the ten datasets tested. The second and third columns are the number of genomes predicted by the methods that have better relative abundance than the other method. The last column “StrainIQ’s lead (%)” shows the percentage of the total number of genomes doing better for StrainIQ in comparison to KrakenUniq. Overall, StrainIQ’s relative abundance performance is much better than KrakenUniq’s, while KrakenUniq performs slightly better than StrainIQ with Set3 and Set6.

Table 12: Number of genomes with better relative abundance

Samples	# of Genomes	StrainIQ	KrakenUniq	StrainIQ's lead (%)
Set1	300	211	176	16.6
Set2	300	196	190	3.06
Set3	300	190	198	-4.21
Set4	200	183	140	23.5
Set5	200	175	143	18.3
Set6	200	147	151	-2.72
Set7	200	203	127	37.4
Set8	200	187	145	22.5
Set9	200	179	147	17.9
Set10	200	173	142	17.9

We tested the performance of StrainIQ using both even and staggered mock communities. Even communities have 12 genomes with an even relative abundance of 8.33%, and the staggered communities have 20 genomes with varying relative abundance ranging from 0.02% to 18%. The actual relative abundance is shown in Appendix 2. *Figure 30*shows the comparison of estimated relative abundance between StrainIQ and KrakenUniq for the twelve genomes present in the even mock microbial community and twenty genomes present in the staggered mock community. Panel A in the figure shows the relative comparison for the even samples. The y-axis on the left represents the relative abundance and the y-axis on the right is the number of unique *n*-grams represented by the green line. Panel B in the figure shows the relative comparison of the staggered samples. While generating the figure, we ignored the genomes falsely predicted by either of the tools. For even mock community, KrakenUniq and StrainIQ produced 162 and 56 false positives, respectively. For the staggered mock community, KrakenUniq and StrainIQ produced 89 and 49 false positives, respectively. Apart from producing a significantly large number of false positives, KrakenUniq follows the same trend as SrainIQ as shown by the red and orange lines. Prediction from both the tools seems to follow the same trend above and below the expected relative

abundance. To understand the reason behind the trend, we added a line corresponding to the number of unique *n*-grams in each genome in the even (green line) and staggered (orange line) mix samples. We observed that the relative abundance accuracy is dependent on the number of unique *n*-grams in the genome for both tools. For *Enterobacter cloacae subsp cloacae ATCC 13047*, which has 4,763,541 unique *n*-grams, both KrakenUniq and StrainIQ overestimated the relative abundance. Similarly, for *Escherichia coli ATCC 700926*, which has the least number of unique *n*-grams (180,987), both tools underestimated the relative abundance. The same trend follows for the staggered samples as well. For *Rhodobacter sphaeroides ATCC 17029,* which has 4,403,102 unique *n*-grams, both tools overestimate the relative abundance and for *Staphylococcus epidermidis ATCC 12228* which has 173,174 unique *n*-grams, both tools underestimate the relative abundance.

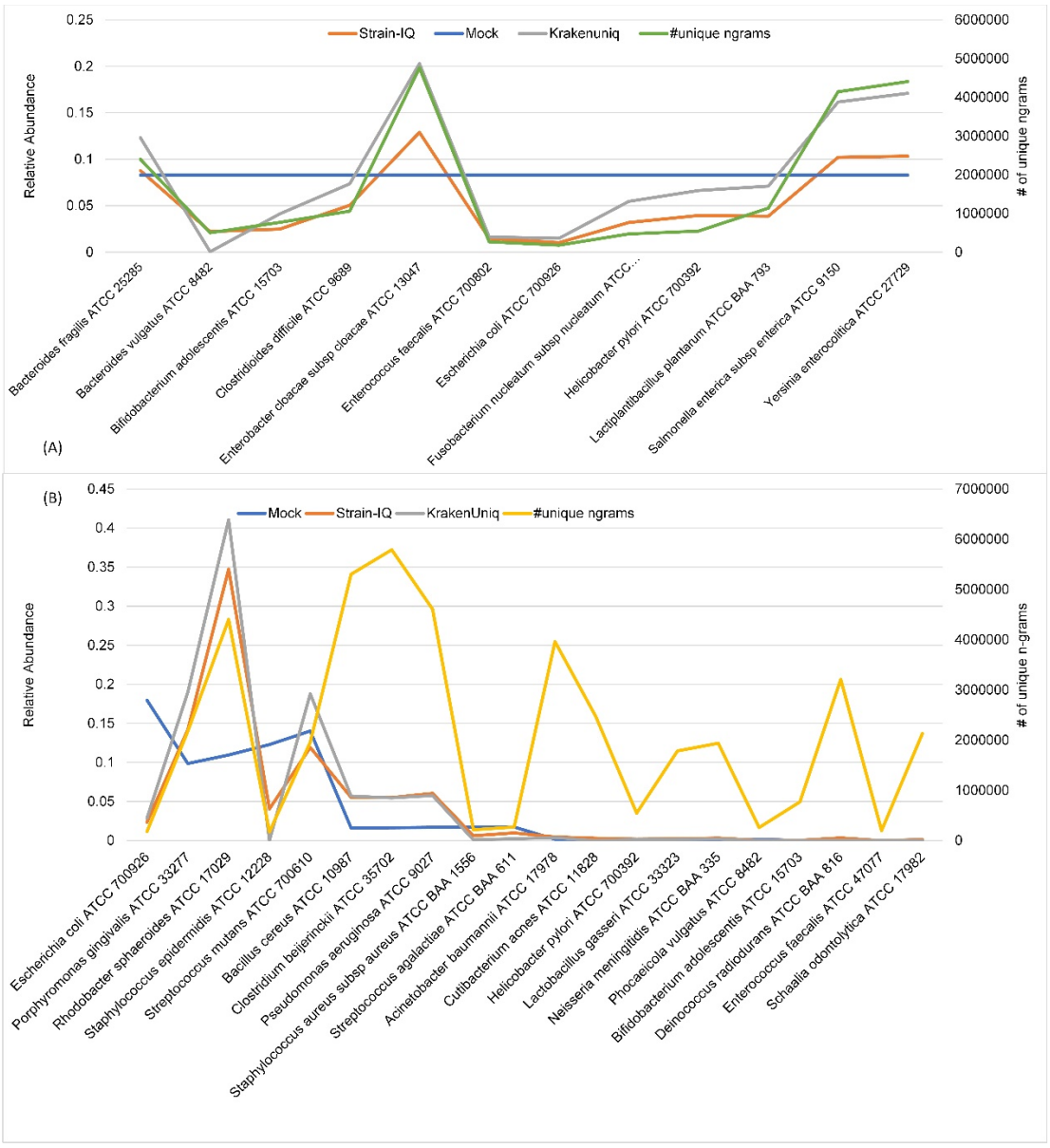

Figure 30: Relative abundance comparison between StrainIQ and KrakenUniq for even (A) and staggered (B) communities

6. Discussion and conclusions

We tested StrainIQ with simulated and experimental datasets. StrainIQ was able to identify taxa at the strain-level in the simulated datasets at an average sensitivity and specificity of 85.8% and 78.2%, respectively. StrainIQ was able to identify metagenomics strains in the evenly distributed mock communities at 100% sensitivity and 89.4%

specificity. We also showed that StrainIQ is robust enough to withstand the variability in the completeness of reference genomes used for training models. Our method was able to accurately identify taxa even when the reference genomes were only 25% complete. For the experimental samples, we tested the ability of StrainIQ to identify metagenomes in low coverage datasets and found that the prediction had a sensitivity of 100% and specificity of 90% for the even mix samples.

StrainIQ performed well when compared against other popular methods. Despite its primary objective of identifying strains in metagenomic samples, StrainIQ was able to identify genus and species with a better F1 score compared to MetaPhlAn and CLARK. StrainIQ outperforms KrakenUniq at the strain level with an F1 score of 0.82 as against 0.639 for KrakenUniq. We also compared StrainIQ against KrakenUniq using the experimental dataset with varying reference genome quality and sample coverage. StrainIQ outperformed KrakenUniq at strain level prediction in all cases. StrainIQ performed better at abundance estimation as well. None of the tools could predict the relative abundance with 100% accuracy, but StrainIQ was closer to the simulated abundance levels more often than KrakenUniq.

In the future, we plan to optimize DSEMs based on the taxa level to improve the accuracy of StrainIQ at a higher taxonomic level.

Chapter 4: DISTRIBUTION of StrainIQ

1. Introduction

During the development of StrainIQ, we prioritized the availability and ease of use as these are important features for any bioinformatics tool. We distribute StrainIQ using the popular source code hosting platform GitHub[59]. The tool is freely available to any user under the GPL License V3.0 [60] which in part states:

“This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <https://www.gnu.org/licenses/>.”

2. Configuration and installation

StrainIQ was developed using Python 3 and relies, intentionally, on very few packages for ease of configuration and installation. Table 13 lists other dependencies and package versions. In addition to Python version 3 or higher, StrainIQ requires BioPython version 1.72 or higher and Pandas 0.23.4 or higher.

Table 13: StrainIQ dependencies	
Package	**Version**
Python	3.6.2
BioPython	1.72
Pandas	0.23.4

StrainIQ can simply be downloaded from GitHub (https://github.com/sanpande/StrainIQ) and run as described in Chapter 3 section 2.2. It consists of a driving script StrainIQ.py which runs three main programs StrainIQ_builder.py, StrainIQ_identifier.py, and StrainIQ_quantifier.py.

3. Supporting database and configuration files

StrainIQ requires several DSEMs, configuration files, and other taxonomy files to run all three programs. Because of the large size of the DSEMs, these files are shared via box.com and can be accessed using this link: https://unmcresearch.box.com/s/3vw007n9os83prgme4zo87ykz1yz89dg.

Chapter 5: PROJECT SUMMARY AND FUTURE DIRECTIONS

StrainIQ leverages the discriminatory nature of unique and weighted common *n*-grams to efficiently identify the taxa in any metagenomic sample. With the appropriate size of *n*, the combination of unique and weighted common *n*-grams can distinguish different taxa up to the strain level present in any metagenomic samples with high accuracy. We optimized the size of *n* by generating and comparing the uniqueness of *n*-grams present in the reference genomes for different sizes of *n*. We used different *n* as multiples of 3 between 12 and 27 because the genetic code is a triplet code made of a series of three nucleotides. The number of unique/common *n*-grams increases with the size of *n*. For *n*=12, the number of unique *n*-grams is in thousands whereas for higher *n*'s it reaches millions. The memory requirement also increases with the size of *n*. The time required to generate *n*-grams increase with the size of the *n*-grams ranging from 1.54 hours for n=12 to 4.4 hours for n=27. Based on the number of unique *n*-grams, and the memory requirements for different *n*-grams, we choose n=21 for generating models for gut genomes. Unlike most other methods, we use a comprehensive list of all overlapping *n*-grams for DSEM development and taxa prediction that requires us to store and process a large amount of data. We used Huffman encoding to encode the bases in *n*-grams to binary to reduce the amount of storage required to store them. We were able to reduce the memory and storage requirement by almost 75%.

We used a scoring function that takes into account the discriminatory nature of the *n*-grams in a body site and assigns weights to them. The score is the reflection of the discriminatory nature of the *n*-gram. The *n*-grams occurring in fewer genomes are assigned higher weights and the *n*-grams occurring in more genomes get lower weights. The weight of the *n*-gram decays rapidly as its uniqueness decreases.

Our method uses the knowledge of body site specific microbial communities to accurately identify and quantify the genomes. This helps to reduce false positives significantly. The DSEMs are built for each body site based on the genomes of microbes known to reside in the body site. The method can easily be implemented for other environments such as ocean floors, ponds, and agricultural sites for accurate identification and quantification by building the environment-specific DSEMs.

Unlike other methods, StrainIQ starts by identifying strains in the metagenomic samples and builds up to higher-level taxa. StrainIQ performs better than CLARK, MetaPhlAn, and KrakenUniq at the strain level. Based on the F1 scores shown in Table 11, StrainIQ performs better than MetaPhlAn or CLARK at higher-level taxa but underperforms to KrakenUniq. The DSEM captures the uniqueness of *n*-grams at strain level because of which it performs better at strain level predictions.

StrainIQ calculates the relative abundance by assigning the reads to the organisms identified during the identification step. The relative abundance predicted by StrainIQ and KrakenUniq follow a similar trend but, StrainIQ has a lower false positive rate compared to KrakenUniq. The relative abundance values for both methods is dependent on the number of unique *n*-grams in the genome for both tools.

StrainIQ performs better than all the three tools compared above at the strain level. While its strength is at the strain level, StrainIQ does a better job at predicting higher-level taxa compared to MetaPhlAn and CLARK. We plan to improve the accuracy of StrainIQ by creating individual DSEMs at each taxonomic level. We believe that the change in the composition of unique and common *n*-grams will make the DSEM better suited for predicting taxa at a higher taxonomic level over the strain level.

REFERENCES

1. Davis CD: **The Gut Microbiome and Its Role in Obesity**. *Nutr Today* 2016, **51**(4):167-174.
2. Clemente JC, Ursell LK, Parfrey LW, Knight R: **The impact of the gut microbiota on human health: an integrative view**. *Cell* 2012, **148**(6):1258-1270.
3. Toor D, Wsson MK, Kumar P, Karthikeyan G, Kaushik NK, Goel C, Singh S, Kumar A, Prakash H: **Dysbiosis Disrupts Gut Immune Homeostasis and Promotes Gastric Diseases**. *Int J Mol Sci* 2019, **20**(10).
4. Clapp M, Aurora N, Herrera L, Bhatia M, Wilen E, Wakefield S: **Gut microbiota's effect on mental health: The gut-brain axis**. *Clin Pract* 2017, **7**(4):987.
5. Chen K, Pachter L: **Bioinformatics for whole-genome shotgun sequencing of microbial communities**. *PLoS Comput Biol* 2005, **1**(2):106-112.
6. Turnbaugh PJ, Ley RE, Hamady M, Fraser-Liggett CM, Knight R, Gordon JI: **The human microbiome project**. *Nature* 2007, **449**(7164):804-810.
7. Mullard A: **Microbiology: the inside story**. *Nature* 2008, **453**(7195):578-580.
8. Zhang YJ, Li S, Gan RY, Zhou T, Xu DP, Li HB: **Impacts of gut bacteria on human health and diseases**. *Int J Mol Sci* 2015, **16**(4):7493-7519.
9. Lloyd-Price J, Abu-Ali G, Huttenhower C: **The healthy human microbiome**. *Genome Med* 2016, **8**(1):51.
10. O'Callaghan A, van Sinderen D: **Bifidobacteria and Their Role as Members of the Human Gut Microbiota**. *Front Microbiol* 2016, **7**:925.
11. Kalliomaki M, Collado MC, Salminen S, Isolauri E: **Early differences in fecal microbiota composition in children may predict overweight**. *Am J Clin Nutr* 2008, **87**(3):534-538.
12. Venturi A, Gionchetti P, Rizzello F, Johansson R, Zucconi E, Brigidi P, Matteuzzi D, Campieri M: **Impact on the composition of the faecal flora by a new probiotic preparation: preliminary data on maintenance treatment of patients with ulcerative colitis**. *Aliment Pharmacol Ther* 1999, **13**(8):1103-1108.
13. Gionchetti P, Rizzello F, Venturi A, Campieri M: **Probiotics in infective diarrhoea and inflammatory bowel diseases**. *J Gastroenterol Hepatol* 2000, **15**(5):489-493.
14. Bae E.-A. HMJ, Song M.-J., Kim D.-H. : **Purification of Rotavirus Infection-Inhibitory Protein from Bifidobacterium Breve K-110.** . *Korean Society for Applied Microbiology* 2002.
15. Chenoll E, Rivero M, Codoner FM, Martinez-Blanch JF, Ramon D, Genoves S, Moreno Munoz JA: **Complete Genome Sequence of Bifidobacterium longum subsp. infantis Strain CECT 7210, a Probiotic Strain Active against Rotavirus Infections**. *Genome Announc* 2015, **3**(2).
16. Freter R: **Parameters affecting the association of vibrios with the intestinal surface in experimental cholera**. *Infect Immun* 1972, **6**(2):134-141.
17. Hudault S, Guignot J, Servin AL: **Escherichia coli strains colonising the gastrointestinal tract protect germfree mice against Salmonella typhimurium infection**. *Gut* 2001, **49**(1):47-55.

18. Bentley R, Meganathan R: **Biosynthesis of vitamin K (menaquinone) in bacteria**. *Microbiol Rev* 1982, **46**(3):241-280.
19. Lawrence JG, Roth JR: **Evolution of coenzyme B12 synthesis among enteric bacteria: evidence for loss and reacquisition of a multigene complex**. *Genetics* 1996, **142**(1):11-24.
20. Ahrne S, Hagslatt ML: **Effect of lactobacilli on paracellular permeability in the gut**. *Nutrients* 2011, **3**(1):104-117.
21. Igwaran A, Okoh AI: **Human campylobacteriosis: A public health concern of global importance**. *Heliyon* 2019, **5**(11):e02814.
22. Kau AL, Martin SM, Lyon W, Hayes E, Caparon MG, Hultgren SJ: **Enterococcus faecalis tropism for the kidneys in the urinary tract of C57BL/6J mice**. *Infect Immun* 2005, **73**(4):2461-2468.
23. Rajkumari N, Mathur P, Misra MC: **Soft Tissue and Wound Infections Due to Enterococcus spp. Among Hospitalized Trauma Patients in a Developing Country**. *J Glob Infect Dis* 2014, **6**(4):189-193.
24. Czepiel J, Drozdz M, Pituch H, Kuijper EJ, Perucki W, Mielimonka A, Goldman S, Wultanska D, Garlicki A, Biesiada G: **Clostridium difficile infection: review**. *Eur J Clin Microbiol Infect Dis* 2019, **38**(7):1211-1221.
25. Häggström M: **Medical gallery of Mikael Häggström 2014**. *WikiJournal of Medicine* 2014.
26. Di Domenico EG, Cavallo I, Pontone M, Toma L, Ensoli F: **Biofilm Producing Salmonella Typhi: Chronic Colonization and Development of Gallbladder Cancer**. *Int J Mol Sci* 2017, **18**(9).
27. Cindoruk M, Cirak MY, Unal S, Karakan T, Erkan G, Engin D, Dumlu S, Turet S: **Identification of Helicobacter species by 16S rDNA PCR and sequence analysis in human liver samples from patients with various etiologies of benign liver diseases**. *Eur J Gastroenterol Hepatol* 2008, **20**(1):33-36.
28. Le Chatelier E, Nielsen T, Qin J, Prifti E, Hildebrand F, Falony G, Almeida M, Arumugam M, Batto JM, Kennedy S *et al*: **Richness of human gut microbiome correlates with metabolic markers**. *Nature* 2013, **500**(7464):541-546.
29. Garrett WS: **Cancer and the microbiota**. *Science* 2015, **348**(6230):80-86.
30. Tsilimigras MC, Fodor A, Jobin C: **Carcinogenesis and therapeutics: the microbiota perspective**. *Nat Microbiol* 2017, **2**:17008.
31. Human Microbiome Project C: **A framework for human microbiome research**. *Nature* 2012, **486**(7402):215-221.
32. Wooley JC, Godzik A, Friedberg I: **A primer on metagenomics**. *PLoS Comput Biol* 2010, **6**(2):e1000667.
33. Woese CR, Fox GE: **Phylogenetic structure of the prokaryotic domain: the primary kingdoms**. *Proc Natl Acad Sci U S A* 1977, **74**(11):5088-5090.
34. Escobar-Zepeda A, Vera-Ponce de Leon A, Sanchez-Flores A: **The Road to Metagenomics: From Microbiology to DNA Sequencing Technologies and Bioinformatics**. *Front Genet* 2015, **6**:348.

35. Langmead B, Trapnell C, Pop M, Salzberg SL: **Ultrafast and memory-efficient alignment of short DNA sequences to the human genome**. *Genome Biol* 2009, **10**(3):R25.
36. Li H, Durbin R: **Fast and accurate short read alignment with Burrows-Wheeler transform**. *Bioinformatics* 2009, **25**(14):1754-1760.
37. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: **Basic local alignment search tool**. *J Mol Biol* 1990, **215**(3):403-410.
38. Buchfink B, Reuter K, Drost HG: **Sensitive protein alignments at tree-of-life scale using DIAMOND**. *Nat Methods* 2021, **18**(4):366-368.
39. Huson DH, Auch AF, Qi J, Schuster SC: **MEGAN analysis of metagenomic data**. *Genome Res* 2007, **17**(3):377-386.
40. Segata N, Waldron L, Ballarini A, Narasimhan V, Jousson O, Huttenhower C: **Metagenomic microbial community profiling using unique clade-specific marker genes**. *Nat Methods* 2012, **9**(8):811-814.
41. Meyer F, Paarmann D, D'Souza M, Olson R, Glass EM, Kubal M, Paczian T, Rodriguez A, Stevens R, Wilke A *et al*: **The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes**. *BMC Bioinformatics* 2008, **9**:386.
42. Zielezinski A, Vinga S, Almeida J, Karlowski WM: **Alignment-free sequence comparison: benefits, applications, and tools**. *Genome Biol* 2017, **18**(1):186.
43. Chan CX, Bernard G, Poirion O, Hogan JM, Ragan MA: **Inferring phylogenies of evolving sequences without multiple sequence alignment**. *Sci Rep* 2014, **4**:6504.
44. Caporaso JG, Kuczynski J, Stombaugh J, Bittinger K, Bushman FD, Costello EK, Fierer N, Pena AG, Goodrich JK, Gordon JI *et al*: **QIIME allows analysis of high-throughput community sequencing data**. *Nat Methods* 2010, **7**(5):335-336.
45. Liu B, Gibbons T, Ghodsi M, Treangen T, Pop M: **Accurate and fast estimation of taxonomic profiles from metagenomic shotgun sequences**. *BMC Genomics* 2011, **12 Suppl 2**:S4.
46. Freitas TA, Li PE, Scholz MB, Chain PS: **Accurate read-based metagenome characterization using a hierarchical suite of unique signatures**. *Nucleic Acids Res* 2015, **43**(10):e69.
47. Rho M, Tang H, Ye Y: **FragGeneScan: predicting genes in short and error-prone reads**. *Nucleic Acids Res* 2010, **38**(20):e191.
48. Rognes T, Flouri T, Nichols B, Quince C, Mahe F: **VSEARCH: a versatile open source tool for metagenomics**. *PeerJ* 2016, **4**:e2584.
49. Wood DE, Salzberg SL: **Kraken: ultrafast metagenomic sequence classification using exact alignments**. *Genome Biol* 2014, **15**(3):R46.
50. Crick FH, Barnett L, Brenner S, Watts-Tobin RJ: **General nature of the genetic code for proteins**. *Nature* 1961, **192**:1227-1232.
51. Huffman DA: **A method for the construction of minimum-redundancy codes**. *Proceedings of the IRE* 1976, **40**(9):1098-1101.
52. Srinivasan SM, Vural S, King BR, Guda C: **Mining for class-specific motifs in protein sequence classification**. *BMC Bioinformatics* 2013, **14**:96.

53. Guda C, King BR, Pal LR, Guda P: **A top-down approach to infer and compare domain-domain interactions across eight model organisms**. *PLoS One* 2009, **4**(3):e5096.
54. **https://www.atcc.org/**
55. Ounit R, Wanamaker S, Close TJ, Lonardi S: **CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers**. *BMC Genomics* 2015, **16**:236.
56. Breitwieser FP, Baker DN, Salzberg SL: **KrakenUniq: confident and fast metagenomics classification using unique k-mer counts**. *Genome Biol* 2018, **19**(1):198.
57. Clark DP, Pazdernik NJ, McGehee MR: **Plasmids**. In: *Molecular Biology.* 2019: 712-748.
58. Gourle H, Karlsson-Lindsjo O, Hayer J, Bongcam-Rudloff E: **Simulating Illumina metagenomic data with InSilicoSeq**. *Bioinformatics* 2019, **35**(3):521-522.
59. **GitHub: Where the world builds software** [https://github.com/]
60. **GNU General Public License V3.0**.

Appendix 1: GI tract DSEM stats

GI tract DSEM stats				
GenomeID	Genome	# non-repetitive *n*-grams	# Unique *n*-grams	% Unique *n*-grams
1	GCF_000003135.1	2366754	527731	22.2977
2	GCF_000010385.1	5064254	489439	9.66458
3	GCF_000143745.1	3187268	1024995	32.159
4	GCF_000145315.2	3571291	3519193	98.5412
5	GCF_000146325.1	1924613	1903245	98.8898
6	GCF_000146835.1	1676089	1647450	98.2913
7	GCF_000147295.1	2913976	184360	6.32675
8	GCF_000147455.1	2775118	375032	13.5141
9	GCF_000147475.1	2827289	264871	9.36837
10	GCF_000147495.1	2872359	257721	8.97245
11	GCF_000147595.1	2953068	219200	7.42279
12	GCF_000148065.1	3109470	295668	9.50863
13	GCF_000148225.1	2987831	229804	7.69133
14	GCF_000148995.1	2570703	2534722	98.6003
15	GCF_000153885.1	2830453	2682464	94.7715
16	GCF_000153905.1	5264812	3281162	62.3225
17	GCF_000153925.1	4575095	2501830	54.6837
18	GCF_000154065.1	5238277	2644431	50.4828
19	GCF_000154085.1	5189006	1046268	20.1632
20	GCF_000154105.1	6931876	3605393	52.0118
21	GCF_000154125.1	9310820	2225788	23.9054
22	GCF_000154205.1	7645012	3044848	39.8279
23	GCF_000154245.1	6112106	2746904	44.942
24	GCF_000154285.1	5097095	2034481	39.9145
25	GCF_000154305.1	6443148	3429130	53.2213
26	GCF_000154325.1	5743333	2603968	45.339
27	GCF_000154345.1	6041223	3148096	52.1102
28	GCF_000154365.1	9284485	2214024	23.8465
29	GCF_000154385.1	6626884	2620164	39.5384
30	GCF_000154405.1	6870364	1653150	24.062
31	GCF_000154425.1	8131647	2943244	36.1949
32	GCF_000154465.1	7086490	2419603	34.1439
33	GCF_000154485.1	8306004	1833524	22.0747
34	GCF_000154505.1	8781191	3357755	38.238
35	GCF_000154525.1	9151255	3304567	36.1105
36	GCF_000154545.1	9168268	1962598	21.4064
37	GCF_000154565.1	9412069	3450089	36.656

38	GCF_000154805.1	8425834	2307731	27.3888
39	GCF_000154825.1	9141884	2248954	24.6006
40	GCF_000154845.1	10444678	3514833	33.6519
41	GCF_000154865.1	11371261	4274353	37.5891
42	GCF_000154985.1	9547998	1726438	18.0817
43	GCF_000155085.1	10673795	3991000	37.3906
44	GCF_000155205.1	9706977	2461927	25.3624
45	GCF_000155395.1	10487794	239509	2.28369
46	GCF_000155415.1	11615945	724212	6.23464
47	GCF_000155455.1	9733039	1564199	16.071
48	GCF_000155475.1	11316029	2097335	18.5342
49	GCF_000155495.1	11221182	1865485	16.6247
50	GCF_000155515.2	12528713	2451331	19.5657
51	GCF_000155815.1	13558679	2380977	17.5605
52	GCF_000155855.1	12129946	2754943	22.7119
53	GCF_000155875.1	11952081	2849617	23.842
54	GCF_000155955.1	10973342	2520533	22.9696
55	GCF_000155975.1	14769953	4685054	31.7202
56	GCF_000155995.1	15825625	3993969	25.2374
57	GCF_000156015.1	12945693	2315849	17.889
58	GCF_000156055.1	13728020	2367624	17.2467
59	GCF_000156075.1	14761229	2739895	18.5614
60	GCF_000156175.1	11500527	1719719	14.9534
61	GCF_000156195.1	16456693	3761454	22.8567
62	GCF_000156215.1	12834899	2295418	17.8842
63	GCF_000156375.1	14184633	2753676	19.4131
64	GCF_000156395.1	14590787	3572077	24.4817
65	GCF_000156495.1	15563981	3929459	25.2471
66	GCF_000156515.1	16355040	3773070	23.0698
67	GCF_000156535.1	16197546	3936593	24.3036
68	GCF_000156635.1	15683932	1818949	11.5975
69	GCF_000156655.1	13901220	2316219	16.662
70	GCF_000156675.1	15146766	2858084	18.8693
71	GCF_000157015.1	18410692	4468784	24.2728
72	GCF_000157035.2	18666478	874045	4.68243
73	GCF_000157075.2	18335175	2756763	15.0354
74	GCF_000157095.2	21601695	1062015	4.91635
75	GCF_000157915.1	19659278	3383072	17.2085
76	GCF_000157935.1	17579673	3353675	19.077
77	GCF_000157955.1	17112536	3046341	17.8018
78	GCF_000157975.1	18318445	3478015	18.9864

79	GCF_000157995.1	18929285	3674999	19.4144
80	GCF_000158015.1	16908517	1078515	6.37853
81	GCF_000158035.1	23038504	6120915	26.5682
82	GCF_000158055.1	19458216	4454099	22.8906
83	GCF_000158075.1	22651787	6045349	26.6882
84	GCF_000158195.2	18838255	2561320	13.5964
85	GCF_000158235.1	17405397	1373493	7.89119
86	GCF_000158255.2	19149252	572864	2.99157
87	GCF_000158295.2	19516927	1827828	9.36535
88	GCF_000158315.2	23102833	3063225	13.2591
89	GCF_000158395.1	23116240	305793	1.32285
90	GCF_000158415.2	20327279	296018	1.45626
91	GCF_000158435.2	21141973	2401847	11.3606
92	GCF_000158455.1	20488076	1614226	7.87886
93	GCF_000158475.2	19580470	2435443	12.4381
94	GCF_000158495.1	19817422	2388138	12.0507
95	GCF_000158655.1	21725578	3316468	15.2653
96	GCF_000158835.2	20610763	1227365	5.95497
97	GCF_000159055.1	22928368	3521972	15.3608
98	GCF_000159075.2	28040081	4160045	14.8361
99	GCF_000159175.1	22578384	1557133	6.89657
100	GCF_000159215.1	22927418	1753800	7.64936
101	GCF_000159255.1	23585073	315877	1.33931
102	GCF_000159275.1	23527940	200158	0.85072
103	GCF_000159315.1	25711455	1118592	4.35056
104	GCF_000159375.2	21527741	2040699	9.47939
105	GCF_000159415.1	21978853	2044371	9.30154
106	GCF_000159455.2	21648819	773895	3.57477
107	GCF_000159475.2	24897027	623160	2.50295
108	GCF_000159495.1	24596030	2342809	9.52515
109	GCF_000159615.1	24670170	893062	3.62001
110	GCF_000159675.1	25281591	972634	3.8472
111	GCF_000159715.1	25031238	1879607	7.50905
112	GCF_000159855.2	27821543	897221	3.22491
113	GCF_000159875.2	29635120	3475530	11.7277
114	GCF_000159915.2	23580204	1689092	7.16318
115	GCF_000159975.2	29530742	3565259	12.073
116	GCF_000160095.1	28061358	6266510	22.3315
117	GCF_000160175.1	25065164	963475	3.84388
118	GCF_000160455.2	25635660	2103462	8.20522
119	GCF_000160575.1	25539758	1929772	7.55595

120	GCF_000160595.1	23276830	1601222	6.87904
121	GCF_000160715.1	26103555	568829	2.17912
122	GCF_000160835.1	27261176	2188240	8.02695
123	GCF_000160855.1	26693621	1660492	6.22056
124	GCF_000162015.1	27634638	2469301	8.93553
125	GCF_000162075.1	28606801	3253122	11.3718
126	GCF_000162115.1	28767281	3083311	10.7181
127	GCF_000162135.1	30735314	1547779	5.03583
128	GCF_000162155.1	29169525	1113222	3.81639
129	GCF_000162275.1	33264934	1746888	5.25144
130	GCF_000162515.1	32327764	3470514	10.7354
131	GCF_000162555.1	30030003	2251501	7.49751
132	GCF_000163655.1	28870911	921101	3.19041
133	GCF_000163735.1	27223149	2063833	7.58117
134	GCF_000163915.2	31679232	461272	1.45607
135	GCF_000163935.1	29667055	2072201	6.98486
136	GCF_000163955.1	31417107	3514333	11.186
137	GCF_000164115.1	31367819	2742435	8.74283
138	GCF_000164175.1	32768494	1230513	3.75517
139	GCF_000164195.1	36827006	851916	2.31329
140	GCF_000164215.1	32895885	477011	1.45006
141	GCF_000164235.1	31998143	587255	1.83528
142	GCF_000164255.1	34626600	389854	1.12588
143	GCF_000164275.1	34348942	276468	0.80488
144	GCF_000164295.1	33030595	152035	0.46029
145	GCF_000164315.1	36546021	889892	2.43499
146	GCF_000164335.1	35101716	272649	0.77674
147	GCF_000164355.1	34126448	963605	2.82363
148	GCF_000164375.1	34160966	313949	0.91903
149	GCF_000164415.1	38303244	426498	1.11348
150	GCF_000164435.1	35405170	223969	0.63259
151	GCF_000164455.1	37060554	269335	0.72674
152	GCF_000164475.1	37586072	259045	0.6892
153	GCF_000164495.1	33509377	461031	1.37583
154	GCF_000164515.1	35099799	192017	0.54706
155	GCF_000164535.1	36532277	231991	0.63503
156	GCF_000164555.1	38000420	337507	0.88817
157	GCF_000164575.1	40414056	195257	0.48314
158	GCF_000164595.1	38156742	320495	0.83994
159	GCF_000164615.1	38659100	502859	1.30075
160	GCF_000164655.1	36877497	1375909	3.73103

161	GCF_000166035.1	39443146	2519001	6.38641
162	GCF_000169015.1	42652396	4053397	9.50333
163	GCF_000169035.1	36495101	2170517	5.94742
164	GCF_000169235.1	37267851	2249396	6.03575
165	GCF_000169255.2	39587546	3925320	9.91554
166	GCF_000169475.1	41001727	1894076	4.6195
167	GCF_000172135.1	39652280	2466272	6.21975
168	GCF_000172175.1	39473165	5025537	12.7315
169	GCF_000173355.1	36959637	1831578	4.95562
170	GCF_000173375.1	41628058	1947358	4.67799
171	GCF_000173415.1	40499207	633615	1.56451
172	GCF_000173435.1	39542277	1840359	4.65416
173	GCF_000173455.1	41497552	1623040	3.91117
174	GCF_000173795.1	40649143	2605761	6.41037
175	GCF_000173815.1	41028792	4424886	10.7848
176	GCF_000173975.1	41856963	3084030	7.36802
177	GCF_000174195.1	44416354	3543493	7.9779
178	GCF_000177015.3	39998367	2859979	7.15024
179	GCF_000178475.1	39691190	1476998	3.72122
180	GCF_000178935.2	38508688	410705	1.06653
181	GCF_000183585.1	45708860	1193083	2.61018
182	GCF_000185325.1	43743060	2220239	5.07564
183	GCF_000185345.1	42618437	1594412	3.74113
184	GCF_000185605.1	47097317	2430277	5.16012
185	GCF_000185705.2	43940856	2666446	6.06826
186	GCF_000185845.1	42787987	3016163	7.04909
187	GCF_000186105.1	44751103	1854828	4.14476
188	GCF_000186505.1	43562249	2155593	4.94831
189	GCF_000186525.1	42288629	3667417	8.67235
190	GCF_000186545.1	44417705	853587	1.92173
191	GCF_000187265.1	43563900	1539991	3.53502
192	GCF_000187895.1	48053003	3826753	7.96361
193	GCF_000188175.1	42603733	2020533	4.74262
194	GCF_000188195.1	42293594	2284301	5.40106
195	GCF_000189595.1	44622621	2141263	4.7986
196	GCF_000189615.1	49067899	1773101	3.61357
197	GCF_000190355.1	48857659	5147341	10.5354
198	GCF_000191805.1	44047476	1279117	2.90395
199	GCF_000191845.1	46135277	1681977	3.64575
200	GCF_000191865.1	47353842	2912220	6.14991
201	GCF_000192165.1	47568412	1827846	3.84256

202	GCF_000195615.1	47262170	3325514	7.03631
203	GCF_000195635.1	48225599	3901729	8.09058
204	GCF_000195655.1	52562838	1019922	1.94039
205	GCF_000204455.1	49539107	6908327	13.9452
206	GCF_000205025.1	45112639	2802834	6.21297
207	GCF_000205165.1	46058211	3266727	7.09261
208	GCF_000209385.2	49736477	2097941	4.21811
209	GCF_000213135.1	46343655	444002	0.95806
210	GCF_000213555.1	53150663	5054164	9.50913
211	GCF_000213575.1	51285704	3893533	7.59185
212	GCF_000214295.1	52398954	3520171	6.71802
213	GCF_000218365.1	50264796	1577541	3.13846
214	GCF_000218465.1	53094263	1826053	3.43927
215	GCF_000218645.2	48442159	210057	0.43362
216	GCF_000218655.1	47728615	669751	1.40325
217	GCF_000220865.1	51695336	2450286	4.73986
218	GCF_000224655.1	51696126	2749184	5.31797
219	GCF_000225705.1	47077608	2385717	5.06763
220	GCF_000225745.1	49789011	2817592	5.65906
221	GCF_000231275.1	51065906	2822979	5.52811
222	GCF_000233455.1	55414698	2638983	4.76224
223	GCF_000233955.1	52626118	3642171	6.92084
224	GCF_000234155.1	57920132	3851463	6.64961
225	GCF_000234175.1	50174111	3020601	6.02024
226	GCF_000235505.1	53261996	5435262	10.2048
227	GCF_000235865.1	54544525	950961	1.74346
228	GCF_000235885.1	54348027	2800386	5.15269
229	GCF_000235905.1	57077794	780639	1.36768
230	GCF_000238035.1	58328111	5162476	8.85075
231	GCF_000238615.1	52769529	3216612	6.09559
232	GCF_000239255.1	55896547	4757614	8.51146
233	GCF_000239295.1	55484677	1256482	2.26456
234	GCF_000239335.1	54388266	4285213	7.87893
235	GCF_000239735.1	56310291	2510071	4.45757
236	GCF_000241405.1	53058374	2845028	5.36207
237	GCF_000242155.1	60041640	2244271	3.73786
238	GCF_000242195.1	56970953	1369310	2.40352
239	GCF_000242435.1	52587480	2303983	4.38124
240	GCF_000243175.1	57782911	3383644	5.85579
241	GCF_000243215.1	56490986	3676873	6.50878
242	GCF_000245775.1	59573883	2469293	4.14493

243	GCF_000250875.1	57770252	2311683	4.00151
244	GCF_000261205.1	60476941	875656	1.44792
245	GCF_000261265.1	60857751	753789	1.23861
246	GCF_000273525.1	57669278	338390	0.58678
247	GCF_000296445.1	55344283	2060340	3.72277
248	GCF_000296465.1	56410358	3168152	5.61626
249	GCF_000297775.1	55127001	1427850	2.59011
250	GCF_000297815.1	60317054	3927106	6.51077
251	GCF_000300935.1	61276325	2134053	3.48267
252	GCF_000300955.1	63456906	3040380	4.79125
253	GCF_000307475.1	59482632	849116	1.4275
254	GCF_000310005.2	61200732	102369	0.16727
255	GCF_000315485.1	63831067	6171481	9.66846
256	GCF_000320405.1	58886147	3481027	5.91145
257	GCF_000332875.2	63617580	1739641	2.73453
258	GCF_000344945.2	62079261	82274	0.13253
259	GCF_000344965.2	59380769	111222	0.1873
260	GCF_000344985.2	59328565	117684	0.19836
261	GCF_000345005.2	58103601	50185	0.08637
262	GCF_000345025.2	61651172	105538	0.17119
263	GCF_000345045.2	62369569	89301	0.14318
264	GCF_000345065.2	58013262	126222	0.21757
265	GCF_000345085.2	63353499	73496	0.11601
266	GCF_000345105.2	56776859	91204	0.16064
267	GCF_000345125.2	60512236	72527	0.11986
268	GCF_000345145.2	60517369	74683	0.12341
269	GCF_000345165.2	62816259	131421	0.20921
270	GCF_000345185.2	59302762	62227	0.10493
271	GCF_000345205.2	61939418	87767	0.1417
272	GCF_000345225.2	63184730	96593	0.15287
273	GCF_000345245.2	65259170	86065	0.13188
274	GCF_000345265.2	58098587	96639	0.16634
275	GCF_000345285.2	64157609	103527	0.16136
276	GCF_000345305.2	58977869	85	0.00014
277	GCF_000345325.2	60458587	232	0.00038
278	GCF_000345345.2	61394262	538	0.00088
279	GCF_000345365.2	63645204	175	0.00027
280	GCF_000345385.2	61334653	109753	0.17894
281	GCF_000345405.2	63098333	390	0.00062
282	GCF_000345425.2	60868510	37146	0.06103
283	GCF_000345445.2	60230773	62285	0.10341

284	GCF_000345465.2	63926278	81363	0.12728
285	GCF_000345485.2	64814369	98764	0.15238
286	GCF_000345505.2	66522881	121272	0.1823
287	GCF_000345525.2	62897470	3521	0.0056
288	GCF_000345545.2	58957806	99837	0.16934
289	GCF_000345565.2	59694178	43372	0.07266
290	GCF_000345585.2	65047741	86248	0.13259
291	GCF_000345605.2	62102420	118948	0.19154
292	GCF_000345625.2	61488680	86509	0.14069
293	GCF_000345645.2	64034242	79675	0.12443
294	GCF_000345665.2	61946852	128079	0.20676
295	GCF_000345685.2	64426397	207415	0.32194
296	GCF_000345705.2	64407340	82749	0.12848
297	GCF_000345725.2	61949439	85705	0.13835
298	GCF_000345745.2	65462135	223	0.00034
299	GCF_000345765.2	60864904	196	0.00032
300	GCF_000345785.2	67372732	204	0.0003
301	GCF_000345805.2	64163175	703	0.0011
302	GCF_000345825.2	59582931	610	0.00102
303	GCF_000345845.2	62316305	1479	0.00237
304	GCF_000345865.2	66191446	628	0.00095
305	GCF_000345885.2	60667736	168	0.00028
306	GCF_000345905.2	62466307	624	0.001
307	GCF_000345925.2	64853333	1182	0.00182
308	GCF_000345945.2	65007012	5565	0.00856
309	GCF_000345965.2	65265239	472	0.00072
310	GCF_000345985.2	62461448	3957	0.00634
311	GCF_000346005.2	64849552	58599	0.09036
312	GCF_000346025.2	62381037	38568	0.06183
313	GCF_000346815.2	65968168	92421	0.1401
314	GCF_000346835.2	61280859	64510	0.10527
315	GCF_000346855.2	67988892	75051	0.11039
316	GCF_000346875.2	65111695	87880	0.13497
317	GCF_000381365.1	64769232	4901492	7.56762
318	GCF_000382465.1	66421177	5518825	8.30883
319	GCF_000398925.1	68224815	3139910	4.6023
320	GCF_000400875.1	68408446	547618	0.80051
321	GCF_000411235.1	64908576	583037	0.89824
322	GCF_000411255.1	68585752	3526602	5.14189
323	GCF_000411275.1	67257549	2351864	3.4968
324	GCF_000411295.1	65925806	3442988	5.22252

325	GCF_000411315.1	74503813	9144371	12.2737
326	GCF_000411335.1	67730398	1226732	1.8112
327	GCF_000411355.1	65116902	1854987	2.8487
328	GCF_000411395.1	68175283	2196175	3.22137
329	GCF_000411415.1	67497655	2145776	3.17904
330	GCF_000411435.1	63753149	1186596	1.86124
331	GCF_000411475.1	69712054	1642643	2.35633
332	GCF_000411495.1	76157225	7663124	10.0622
333	GCF_000411515.1	71070686	2144438	3.01733
334	GCF_000411535.1	69284179	1931385	2.78763
335	GCF_000412335.2	69703341	4549799	6.52738
336	GCF_000413355.1	71037066	2787153	3.92352
337	GCF_000413375.1	72056904	2326669	3.22893
338	GCF_000455765.1	68164154	114251	0.16761
339	GCF_000466385.1	72993349	5243816	7.18396
340	GCF_000466445.2	68045805	1708198	2.51036
341	GCF_000466465.2	71722528	6512274	9.07982
342	GCF_000466485.1	70678708	1889953	2.67401
343	GCF_000466525.1	71439784	711287	0.99565
344	GCF_000466565.1	71701643	3232098	4.5077
345	GCF_000466605.1	75114724	2963290	3.94502
346	GCF_000468015.1	67780758	2970048	4.38185
347	GCF_000469305.1	73004831	1859298	2.54682
348	GCF_000469345.1	69817153	3143266	4.50214
349	GCF_000469365.1	73495432	2442002	3.32266
350	GCF_000469425.1	74986364	1534553	2.04644
351	GCF_000469445.2	76567254	1529380	1.99743
352	GCF_000478505.2	74203021	5568775	7.50478
353	GCF_000479045.1	72752343	2998866	4.12202
354	GCF_000479185.1	70475781	1141939	1.62033
355	GCF_000479205.1	77217646	877161	1.13596
356	GCF_000479225.1	75346989	492866	0.65413
357	GCF_000479245.1	75467117	245461	0.32526
358	GCF_000479265.1	70007288	491879	0.70261
359	GCF_000479285.1	73053098	464139	0.63534
360	GCF_000507845.1	74039022	2322863	3.13735
361	GCF_000507865.1	71715603	1864975	2.60051
362	GCF_000517745.1	78577171	1239448	1.57736
363	GCF_000517805.1	80442643	443572	0.55141
364	GCF_000523555.1	74222010	1152906	1.55332
365	GCF_000527215.1	83039332	2692779	3.24278

366	GCF_000527235.1	81055876	2509111	3.09553
367	GCF_000527255.1	73089755	331888	0.45408
368	GCF_000527275.1	78911697	1216256	1.54129
369	GCF_000527295.1	80593067	474411	0.58865
370	GCF_000527315.1	76709748	995192	1.29735
371	GCF_000527335.1	79491509	530616	0.66751
372	GCF_000690925.1	72272197	266980	0.36941
373	GCF_000760655.1	75950491	1295330	1.70549
374	GCF_000763035.1	76697947	1973674	2.57331
375	GCF_000763055.1	76225474	2013563	2.64159
376	GCF_000969835.1	80545159	6078398	7.54657
377	GCF_000969845.1	80886962	6403360	7.91643
378	GCF_001078315.1	81026213	2367635	2.92206
379	GCF_001078425.1	82722674	2248181	2.71773
380	GCF_001078435.1	86602137	2316935	2.67538
381	GCF_001078445.1	77605039	3538400	4.5595
382	GCF_001078555.1	81730803	1585293	1.93965
383	GCF_001185845.1	84966349	2322525	2.73346
384	GCF_001571425.1	82387963	2836211	3.44251
385	GCF_001578555.1	84949914	2905161	3.41985
386	GCF_001578585.1	75649160	1452456	1.91999
387	GCF_001578645.1	79790462	1101090	1.37998
388	GCF_001580195.1	79048896	709072	0.897
389	GCF_001641065.1	84881912	3673604	4.3279
390	GCF_001647615.1	86134269	455305	0.5286
391	GCF_001807055.1	85454403	4221981	4.94062
392	GCF_001807785.1	84540016	2083727	2.46478
393	GCF_001807865.1	79748442	3934560	4.93371
394	GCF_001807895.1	83511716	4292596	5.14011
395	GCF_001808325.1	86090440	3429903	3.98407
396	GCF_001808745.1	81864841	1703392	2.08074
397	GCF_001808795.1	86995243	188786	0.21701
398	GCF_001809065.1	88533733	296302	0.33468
399	GCF_001809145.1	83695902	282058	0.337
400	GCF_001809445.1	86706316	1987220	2.2919
401	GCF_001809485.1	87755365	867309	0.98833
402	GCF_001809495.1	80090742	63064	0.07874
403	GCF_001809645.1	89550257	659825	0.73682
404	GCF_001810115.1	87458621	725870	0.82996
405	GCF_001810435.1	85156802	80141	0.09411
406	GCF_001810475.1	85278345	1946052	2.282

407	GCF_001810595.1	87792208	324364	0.36947
408	GCF_001810625.1	84190994	136597	0.16225
409	GCF_001810915.1	88166469	260240	0.29517
410	GCF_001811035.1	90476252	138565	0.15315
411	GCF_001811205.1	82166507	434387	0.52867
412	GCF_001811225.1	92975987	1871201	2.01256
413	GCF_001811285.1	86327441	575185	0.66628
414	GCF_001811595.1	81484266	289829	0.35569
415	GCF_001811695.1	93684553	4868767	5.19698
416	GCF_001811715.1	89090482	203116	0.22799
417	GCF_001811805.1	87585485	491303	0.56094
418	GCF_001811815.1	91897166	693462	0.75461
419	GCF_001812015.1	89459864	1637625	1.83057
420	GCF_001812445.1	92886005	68847	0.07412
421	GCF_001812505.1	90619958	1069019	1.17967
422	GCF_001812535.1	86018482	407867	0.47416
423	GCF_001813025.1	92081936	3285829	3.56838
424	GCF_001813035.1	86109217	977887	1.13564
425	GCF_001813195.1	88590941	763216	0.86151
426	GCF_001813255.1	84652284	58772	0.06943
427	GCF_001813275.1	91353031	185765	0.20335
428	GCF_001813405.1	88813863	2342013	2.63699
429	GCF_001813585.1	94351276	161780	0.17147
430	GCF_001813745.1	83377451	253098	0.30356
431	GCF_001813905.1	95438027	644274	0.67507
432	GCF_001814065.1	91035576	137867	0.15144
433	GCF_001814235.1	87370104	355755	0.40718
434	GCF_001814745.1	93894979	1147176	1.22177
435	GCF_001814765.1	88262188	983454	1.11424
436	GCF_001814855.1	93842396	3129232	3.33456
437	GCF_001815345.1	91100062	1142027	1.2536
438	GCF_001815665.1	91588207	324661	0.35448
439	GCF_001815745.1	93478889	389313	0.41647
440	GCF_001815825.1	86081209	604108	0.70179
441	GCF_001815925.1	98866393	4451860	4.50291
442	GCF_001835885.1	94076109	338494	0.35981
443	GCF_001836465.1	90111828	634220	0.70381
444	GCF_001836495.1	84871572	216971	0.25565
445	GCF_001836545.1	96190767	315371	0.32786
446	GCF_001836595.1	98166949	314211	0.32008
447	GCF_001837035.1	95365230	22771	0.02388

448	GCF_001837075.1	90651060	32886	0.03628
449	GCF_001837115.1	93853009	609941	0.64989
450	GCF_001837215.1	96206509	629020	0.65382
451	GCF_001837535.1	87043415	115829	0.13307
452	GCF_001838125.1	94722536	215781	0.2278
453	GCF_001838135.1	93808341	2066341	2.20273
454	GCF_001838215.1	94556798	632447	0.66885
455	GCF_001838615.1	96210027	2117468	2.20088
456	GCF_001839265.1	97847557	353758	0.36154
457	GCF_001839285.1	91838753	313884	0.34178
458	GCF_001839345.1	97595171	352179	0.36086
459	GCF_001857645.1	93657055	1251569	1.33633
460	Bifidobacterium adolescentis ATCC 15703	2067125	769535	37.2273
461	Escherichia coli ATCC 700926	9418529	180987	1.92161
462	Helicobacter pylori ATCC 700392	6180584	546669	8.84494
463	Acinetobacter baumannii ATCC 17978	5668133	3960998	69.8819
464	Bacillus cereus ATCC 10987	7413028	5303691	71.5455
465	Bacteroides fragilis ATCC 25285	5203342	2397177	46.0699
466	Bacteroides vulgatus ATCC 8482	5041713	503179	9.98032
467	Clostridioides difficile ATCC 9689	4205468	1061735	25.2465
468	Clostridium beijerinckii ATCC 35702	5867260	5791691	98.712
469	Cutibacterium acnes ATCC 11828	8805494	2469928	28.0499
470	Deinococcus radiodurans ATCC BAA 816	5448125	3211750	58.9515
471	Enterobacter cloacae subsp cloacae ATCC 13047	5466481	4763541	87.1409
472	Enterococcus faecalis ATCC 47077	4857221	198852	4.09395
473	Enterococcus faecalis ATCC 700802	3306074	269069	8.13863
474	Fusobacterium nucleatum subsp	2141474	469751	21.9359

	nucleatum ATCC 25586			
475	Lactiplantibacillus plantarum ATCC BAA 793	3306910	1139589	34.4608
476	Lactobacillus gasseri ATCC 33323	5130270	1786342	34.8196
477	Neisseria meningitidis ATCC BAA 335	6316025	1939452	30.7068
478	Phocaeicola vulgatus ATCC 8482	10176784	261192	2.56655
479	Porphyromonas gingivalis ATCC 33277	2213470	2176966	98.3508
480	Pseudomonas aeruginosa ATCC 9027	10884473	4609633	42.3505
481	Rhodobacter sphaeroides ATCC 17029	9477576	4403102	46.4581
482	Salmonella enterica subsp enterica ATCC 9150	4536490	4145461	91.3804
483	Schaalia odontolytica ATCC 17982	5682205	2131957	37.5199
484	Staphylococcus aureus subsp aureus ATCC BAA 1556	4935593	215675	4.36979
485	Staphylococcus epidermidis ATCC 12228	8376933	173174	2.06727
486	Streptococcus agalactiae ATCC BAA 611	7036210	270332	3.84201
487	Streptococcus mutans ATCC 700610	6522429	1958467	30.0267
488	Yersinia enterocolitica ATCC 27729	4523264	4406846	97.4262

Appendix 2: Mock community genomes

Even mixed genomes (ATCC® MSA-1006™**)**	
Genomes	**Abundance**
Bacteroides fragilis ATCC 25285	0.0830
Bacteroides vulgatus ATCC 8482	0.0830
Bifidobacterium adolescentis ATCC 15703	0.0830
Clostridioides difficile ATCC 9689	0.0830
Enterobacter cloacae subsp cloacae ATCC 13047	0.0830
Enterococcus faecalis ATCC 700802	0.0830
Escherichia coli ATCC 700926	0.0830
Fusobacterium nucleatum subsp nucleatum ATCC 25586	0.0830
Helicobacter pylori ATCC 700392	0.0830
Lactiplantibacillus plantarum ATCC BAA 793	0.0830
Salmonella enterica subsp enterica ATCC 9150	0.0830
Yersinia enterocolitica ATCC 27729	0.0830

Staggered genomes (ATCC® MSA-1003™**)**	
Genomes	**Abundance**
Acinetobacter baumannii ATCC 17978	0.18
Bacillus cereus ATCC 10987	1.8
Bifidobacterium adolescentis ATCC 15703	0.02
Clostridium beijerinckii ATCC 35702	1.8
Cutibacterium acnes ATCC 11828	0.18
Deinococcus radiodurans ATCC BAA 816	0.02
Enterococcus faecalis ATCC 47077	0.02
Escherichia coli ATCC 700926	18
Helicobacter pylori ATCC 700392	0.18
Lactobacillus gasseri ATCC 33323	0.18
Neisseria meningitidis ATCC BAA 335	0.18
Phocaeicola vulgatus ATCC 8482	0.18
Porphyromonas gingivalis ATCC 33277	18
Pseudomonas aeruginosa ATCC 9027	1.8
Rhodobacter sphaeroides ATCC 17029	18
Schaalia odontolytica ATCC 17982	0.02
Staphylococcus aureus subsp aureus ATCC BAA 1556	1.8
Staphylococcus epidermidis ATCC 12228	18
Streptococcus agalactiae ATCC BAA 611	1.8
Streptococcus mutans ATCC 700610	18

Appendix 3: Sensitivity/Specificity comparison of StrainIQ, KrakenUniq, MetaPhlAn, and CLARK

	Genus		Species		Strain	
	Sensitivity	**Specificity**	**Sensitivity**	**Specificity**	**Sensitivity**	**Specificity**
StrainIQ	0.958687	0.97619048	0.8885906	0.86423608	0.867194	0.7514671
KrakenUniq	0.966142	1	0.9378264	0.94463276	0.695642	0.52880991
MetaPhlAn	0.800683	0.97619048	0.9834383	0.16976439	N/A	N/A
CLARK	1	0.03508773	0.5774396	0.96883233	N/A	N/A