DETECTING GENETIC ENGINEERING WITH A KNOWLEDGE-RICH DNA SEQUENCE CLASSIFIER

by
Yuchen Ge

A thesis submitted to The Johns Hopkins University in conformity
with the requirements for the degree of Master of Science in Engineering

Baltimore, Maryland
July 2020

Abstract

Detecting evidence of genetic engineering in the wild is a problem of growing importance for biosecurity, provenance, and intellectual property rights. This thesis describes a computational system designed to detect engineering from DNA sequencing of biological samples and presents its performance on fully blinded test data. The pipeline builds on existing computational resources for metagenomics, including methods that use the full set of reference genomes deposited in GenBank. Starting from raw reads generated from short-read sequencers, the dominant host species are identified by k-mer analysis. Next, all the sequencing reads are mapped to the imputed host strain; those reads that do not map are retained as suspicious. Suspicious reads are de novo assembled to suspicious contigs, followed by sequence alignment against the NCBI non-redundant nucleotide database to annotate the engineered sequence and to identify whether the engineering is in a plasmid or is integrated into the host genome. Our initial system applied to blinded samples provides excellent identification of foreign gene content, the changes most likely to be functional. We have less ability to detect functional structural variants and small indels and SNPs produced by genetic engineering but which are more difficult to distinguish from natural variation. Future work will focus on improved methods for detecting synonymous recoding, used to introduce watermarks and for compatibility with synthesis and assembly methods, for using long read sequence data, and for distinguishing engineered sequence from natural variation.

Thesis Readers

Dr. Joel S. Bader (Primary Advisor)
Professor
Department of Biomedical Engineering
Johns Hopkins University

Dr. Benjamin T. Langmead
Associate Professor
Department of Computer Science
Johns Hopkins University

Dr. Michael Schatz
Associate Professor
Department of Computer Science
Johns Hopkins University

Dedicated to my beloved mother and father,

for their unwavering and unconditional support,

even at the most difficult times.

I could not ask for better parents.

Acknowledgements

I would first like to thank my thesis advisor, Professor Joel S. Bader. The door to Professor Bader's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right the direction whenever he thought I needed it. I want to give special thank to him not only because of his great mentorship, but also his close participation in this project and his excitement in sharing his knowledge of synthetic biology. Without his passionate input, this research could not have been successfully conducted.

I would also like to thank Ms. Jitong Cai, a PhD student from Bader's lab, who devoted her time to writing scripts to analyze the data.

I would like to express my gratitude to Professor Steven Salzberg, Benjamin Langmead, and Derrick Wood for helpful discussions in terms of software choices.

I would also like to acknowledge Professor Benjamin Langmead and Michael Schatz as the secondary readers of this thesis, and I am gratefully indebted to their very valuable comments on this thesis.

This work was funded by the IARPA FELIX program, and I thank them and the other members of our larger team led by BBN/Raytheon. They are fantastic colleagues. I am proud to say that of all the methods developed by our team, the Johns Hopkins classifier described here worked the best.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout the two years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

Thank you all!

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background of the study

Genetic engineering and synthetic biology focus on technologies that deliver foreign genes into host organisms to introduce desired new functions as well as to understand the function of the native genome. For decades, the U.S. government has been funding synthetic biology research to develop genetically engineered organisms in order to expand application and boost production across industries such as agriculture, pharmaceuticals, and biofuels. While this has considerable benefits, deliberate misuse or release of engineered organisms can pose threats to human health and the environment. As a result, challenges regarding both ethics and safety arise — if an engineered organism is deliberately released into the environment, how can we tell it apart from the millions of microorganisms that exist naturally in the wild?

Our group is funded by the US Intelligence Advanced Research Projects Activity (IARPA) under their Finding Engineering-Linked Indicators (FELIX) program as part of an effort led by Raytheon BBN Technologies (BBN). BBN terms this effort GUARDIAN (Guard for Uncovering Accidental Release, Detecting Intentional Alterations, and Nefariousness), an initiative that combines wet-lab and computational analyses. As part of this multidisciplinary collaborative project led by BBN, we aim to improve the ability to detect engineered organisms and enable a new era of synthetic biology focused on safety and efficacy.

The IARPA program also funds Testing and Evaluation groups that generate fully blinded data from real organisms, both native and genetically engineered. Most of the initial data has been generated using Illumina short-read DNA sequencing data, but Nanopore and PacBio long-read DNA sequencing data will be generated in increasing volume to increase throughput and sensitivity for detection. According to Gargis et al. (2019) [1], short-read sequencing technologies have lower error rates than long-read technologies, which makes short-read sequencing particularly useful for microbial strain matching and single nucleotide polymorphism (SNP) identification [2, 3]. However, the disadvantage is that the read length (100-300 nt) is too short to resolve repetitive regions or structural variants, and their de novo assemblies often result in contigs with many gaps [3, 4]. Long-read sequences can span more than 10 kilobases (kb) in length, which makes them effective for closing the gaps in the scaffolds [3]. Unfortunately, long reads have higher error rates [5] and suffer from insertions and deletions (indels) in homopolymeric regions [2, 6]. It is also difficult to accurately assemble small plasmids (<7 kb) from PacBio long-read because shorter reads will typically be down-selected from library preparation and data analysis [7, 8]. While IARPA's assessment strategy favors an initial focus on short-reads, the complementary characteristics of the two types of sequencing data available motivates an ultimate desire to develop a detection system that combines short- and long-read data in order to generate high-quality hybrid assemblies [3, 4, 9, 10] and more accurate detection of engineering.

This thesis describes a semi-automated cross-platform pipeline, the "JHU pipeline" in the context of the BBN GUARDIAN system, that detects evidence of genetic engineering. We aimed to develop a generalizable methods requiring no training data beyond native genomes deposited in GenBank. We note that other groups within the BBN group pursued machine learning approaches using engineered and non-engineered training sets. Our hypothesis was that classifiers based on training sets of engineered genomes would suffer from large generalization error, performing well on the relatively small body of positive training examples but degrading when applied to novel examples. Comparisons of our generalizable system to the other BBN system, provided in the results, support this hypothesis. It is our sincere hope that our efforts will contribute to the diverse computational analytics developed in addressing growing challenges in the field of synthetic biology.

1.2 Statement of the problem

A genetically modified organism (GMO) can be created using multiple techniques. Traditional methods of genetic engineering insert genes of interest randomly into the host organism's genome via a suitable vector. Technology advances in gene targeting and gene editing have allowed genes to be delivered at specific locations more accurately. Since 2009, gene editing has made major breakthroughs, with Transcription Activator-Like Effector Nucleases (TALENs) and the CRISPR-Cas9-gRNA (guide RNA) system developed and commonly used in research and industry alike. Therefore, as the capability of genome-wide engineering techniques is rapidly growing, bioanalytical as well as computational tools must catch up with the growing need to distinguish naturally occurring organisms from those that are engineered.

Before discussing detection of engineering in cells, we note that the availability of synthetic biology resources has made it important to check DNA sequence orders for potentially dangerous sequences, such as those encoding pathogens or toxins. At present, identification of target nucleic acid sequences from a list of dangerous sequences relies heavily on exhaustive alignment via BLAST [11], or other related techniques for biosynthesis screening, such as the International Gene Synthesis Consortium (IGSC)'s "Harmonized Screening Protocol" [12]. In screening for detection, typical standards are approximately 80% sequence similarity with a length of at least 200 bp [13].

Sequences shorter than 200 bp have been difficult to classify accurately as native versus engineered, or similarly as mapping to a pathogenic versus non-pathogenic strain [13]. Methods have been designed to search for specific sequences that are often used for engineering [14], including recombination sites and selectable markers, but these are limited to known examples and can be evaded by newer synthetic biology techniques for scarless editing. Moreover, the set of known signatures should have a broader scale to include specific signatures for different organisms: yeast, bacterial, mammalian, or other hosts.

In physical samples, detection of genetic alterations in GMOs commonly identifies target nucleic acid sequences that are previously known. One of the wet-lab approaches is by Polymerase Chain Reaction (PCR), which has seen wide use for GMO detection [15]. The principle behind this method is to design a pair of primers that bind, in reverse complementary fashion, to regions that flank specific sequence of interest. Then going through controlled PCR cycles, the target sequence is amplified. The PCR products are subjected to electrophoresis, which separates the products by length. If the fragment amplified has the expected length, then it is presumably not engineered. For a specific locus, different primer sets may be designed for the native and the engineered versions, with either presence/absence or length as a read-out. Other methods are possible, such as Southern blotting or brute-force sequencing.

This PCR approach for physical samples is analogous to the methods for screening DNA sequence orders: both rely on a curated list of DNA signatures of engineering. This requirement for pre-existing knowledge is an intrinsic disadvantages of the PCR detection method. It certainly does not serve well in GMOs that are modified using modern-day gene editing techniques, like the CRIPR-Cas system. Rather than PCR, the low costs of next-generation sequencing (NGS) have made full genome sequencing a viable approach for engineering detection.

This thesis describes methods that use whole-genome sequencing data as a source of evidence of genetics engineering. Other members of the BBN consortium, and other teams funded by IARPA, started by re-purposing existing genome annotation pipelines, attempting to build high-quality full genome assemblies. We viewed this as wasteful: even in an engineered genome, only a small fraction of the bases are engineered, perhaps only 10 kb in a typical 5-10 Mb bacterial or fungal genome. We took a different approach, unique across the IARPA teams, to first filter out native and near-native reads, and only then attempt to analyze the (hopefully) small fraction of suspicious reads.

Phase 1 of the IARPA FELIX program concluded in October 2019 with a challenge to detect engineering in blinded samples from four yeast species: *Saccharomyces cerevisiae*, *Yarrowia lipolytica*, *Komagataella pastoris*, and *Komagataella phaffii*. This thesis primarily discusses what we achieved by the end of Phase 1. Limitations are followed in later sections and are the subject of ongoing work. The JHU pipeline described here has the ability to analyze sequencing data generated from mixtures of multiple species including engineered and non-engineered strains, with minimal meta-data needed (it is helpful to know, for example, whether sequencing standards such as PhiX have been spiked in as controls). The pipeline employs several best-performing and widely cited molecular sequence analysis tools developed in other labs at JHU. Removing reads that are native to the host organism as soon as possible and then assembling the suspicious reads only reduces computational requirements substantially. Compared to other groups within BBN that focused on machine learning (e.g. Hidden Markov Model) or deep learning (e.g. Neural Network for Natural Language Processing), the JHU pipeline extracts features by annotating them using NCBI non-redundant databases. In Phase 1, semi-automated annotation was performed to classify these features as native vs. engineered; full automation of this step is underway.

1.3 Objectives of the pipeline

We aim to identify foreign gene content (extrachromosomal plasmid vs integrated), structural variants (breakpoints, amplifications, deletions), non-synonymous editing, synonymous recoding (watermarks, restriction enzyme site addition or subtraction, codon optimization, codon minimization), and other engineering markers in DNA sequencing data from pure strains and mixtures.

These performance objectives are summarized by the mnemonic **GRAIP**.

Generalization. Generalize readily to multiple species and mixed or diluted samples.

Re-use. Rely on top-of-class implementations of unit operations for biological sequence analysis.

Automation. Design towards a system that permits beginning-to-end automation.

Interpretation. Make decisions that a person can interpret.

Performance. Discard as many native reads as possible and as quickly as possible.

Chapter 2

Background: Existing approaches to detect evidence of genetic engineering

2.1 Introduction

Conventional genetic engineering integrate DNA elements into the host genome via techniques such as homologous recombination for nuclear integration or transformation or transfection for plasmids. Two general considerations are the type of sequence introduced and the method used to synthesize, assemble, and integrate the target sequence into the plasmid or genome. If the introduced sequence is foreign, for example a foreign gene, the resulting combination of DNA sequences does not occur naturally and therefore can be readily detected via site-specific PCR-based methods targeting if the insertion site or the insert is known. Modifications that do not introduce foreign genes are also readily introduced, especially since gene editing and other related techniques have been added to the engineering toolbox. These more subtle modifications can entirely resemble natural mutations or gene polymorphisms but nevertheless achieve intended functional changes, such as functional knock-out or unregulated activation. Therefore, improved analytical tools for detection, identification, and quantification of such GMOs are needed.

The second consideration, the method used to synthesize, assemble, and integrate, in the past placed constraints on the DNA sequences flanking an engineered region. These constraints, for example requirements for the presence or absence of restriction enzyme recognition sites, are often termed “scars”, and at one time provided additional evidence of engineering. With the advent of methods such as homologous recombination and CRISPR/Cas9 editing, however, scarless editing is much more feasible and scars can be avoided. Scarless editing is therefore an additional challenge for detection.

In this chapter, we will review existing and proposed detection and identification methods for various types of engineering, with a focus on the use of bioinformatics and other computational approaches. Then we will investigate several gaps in the literature that our JHU pipeline aims to address. These gaps motivated the performance objectives denoted by the mnemonic GRAIP in the preceding chapter: generalization, re-use, automation, interpretation, performance.

2.2 Targeted DNA Amplification-Based Methods

A standard PCR-based method to detect genetic engineering requires knowing the target DNA sequence of the modified locus, so that a pair of primers that are complementary to the sites flanking the locus of interest can be designed accordingly. The resulting molecular constructs, with the addition of a suitable polymerase, undergo cyclic DNA replication to amplify the intended product. Because PCR-based detection methods are highly sensitive and specific, PCR-based detection remains a leading method for known targets [16].

How about shorter sequence modifications, such as those induced by genome editing techniques such as site-directed nuclease systems (SDN) 1, 2, and 3 [17]? Shorter sequence changes (substitutions or indels of one or a few nucleotides) are also detectable using specific probes, for instance TaqMan real-time-PCR or digital

PCR [18]. SNP genotyping approaches can be applied to detect very small sequence differences of a few nucleotides, given an adequate reference sequence [19, 20].

These bioanalytical detection approaches have continued to improve. They are most valuable when a large number of biospecimens are to be assayed for a small number of pre-defined targets. This remains a limitation — PCR and related techniques are only able to detect a small subset of targets with known sequence. For the efforts devoted in the FELIX project, we aim to achieve similar goals for unknown targets by using sequencing data instead. The resulting computational system will be more generalizable to detect a wider range of engineering.

2.3 Untargeted DNA Sequencing-Based Methods

Next-generation sequencing (NGS) technologies takes advantage of massive parallel sequencing for the whole genome, therefore boosting the speed and decreasing the cost. Whole genome sequencing (WGS) has become increasingly feasible as an analytical approach for GMO detection. It requires no prior information of a specific genetic alteration, or even of the host species, and thus can be applied readily as an untargeted detection approach for generic alterations [16]. NGS platforms produce millions of DNA sequencing reads in parallel.

Traditional methods use the raw sequence output as input to traditional genome sequencing pipelines that perform assembly (either de novo or with the assistance of a reference genome) and then compare the result to high-quality reference genomes using bioinformatics software.

The comparison between de novo genome assemblies and a well-annotated reference has its own complications. Even without engineering, a substantial amount of sequence differences are to be expected due to natural genome diversity within strains of the same species. Distinguishing sequence polymorphisms from true engineering is therefore a critical task. Furthermore, another challenge of WGS is that highly repetitive regions in the genome can be difficult to assemble, and even difficult to assign if multiple species with similar repeats are present in a single sample. Flanking an engineered region with native repeats, for example, can be very effective in obfuscating its location. Also considering heterogeneous or contaminated samples, WGS might find its limitations in separating sequences that are discrete in their genetic coordinates. Finally, assembly itself can introduce errors that might resemble engineering, for example non-native junctions that appear chimeric in connecting DNA from different species present in a single sample.

If the resulting genome assembly reveal foreign DNA sequences, it is possible that the genetic modification was introduced by either genome editing or conventional genetic engineering. However, due to sequence homology shared among multiple organisms, WGS approaches are prone to false positives and often confounded by sequences not only from the target organism but also from a wide array of contaminants or pathogens [16]. Furthermore, gene content can vary widely across different strains of a single species, and many such genes may be absent from the reference strain or even the full complement of strains with sequenced genomes. Therefore, detected sequences that are not present in a reference genome or previously sequenced genomes must be carefully evaluated, or experimentally verified if needed, to assess whether they are native or engineered.

Beyond detecting what was introduced or engineered, it is also desirable to detect how it was introduced: classic genetic engineering, genome editing, genome synthesis, or even mutagenesis and selection. These hopes are becoming more and more futile. Modifications introduced by conventional mutagenesis techniques, such as irradiation or mutagenic chemicals as well as genome editing applications, do not leave specific imprints in the genome. Classic genetic engineering using restriction enzymes is decreasing as newer scarless editing becomes easier. If the genes coding for the genome editing components are absent, it cannot be deduced from the altered sequence which specific process has been used. Even for the conventional genetic engineering techniques, it may be impossible to unequivocally identify the underlying technique for the integration of foreign DNA [16]. For these reasons, distinguishing between conventional genetic engineering and genome editing or identifying the process of genetic engineering is difficult.

2.4 Bioinformatics and Statistical Methods

In Lusser et al. (2011) [21], the minimum length of a unique random sequence in a genome was calculated by correlating the genome size with the possible number of combinations for this sequence length. For example in a plant genome, it reports a DNA sequence of at least 20 nt is needed to be considered as unique. However, the probability calculation for theoretical uniqueness is based on the naive assumption that the four bases, A, C, G, and T, are equally distributed and statistically independent. The complexity of alteration, the amount of repetition, and the diversity of genomes are not taken into account. This piece of information, however, might provide insights into how long a DNA sequence we should reliably call for a foreign gene content in terms of an insertion of longer sequences.

The integration of nucleic acid sequences from foreign organisms can also occur naturally, although rarely, as seen in the sweet potato, which was shown to contain *Agrobacterium* genes [22]. In most cases, a sequence of foreign origin can be detected if it has sufficient length. This enables identification of more complicated pathway insertions, typically consecutive foreign genetic elements comprising a combination of promoter, coding sequence, and terminator from different species. For this purpose, search packages like BLAST or k-mer based tools like NIKS [23] can be used to find such DNA elements in WGS data.

A European study applied NGS data to characterize transgenic insertions in GMOs [24]. The reference genome for *Oryza sativa ssp. japonica* was input for the Illumina “Consensus Assessment of Sequence and Variation” (CASAVA) software (Elandv2e), which reported all DNA polymorphisms (SNPs, insertions, deletions and breakpoints) between the rice reference genome and the reads obtained for the LLRice62 DNA sample. CASAVA’s Elandv2e generated 5′- and 3′-borders of inserted sequences at the breakpoints (breakpoint border sequences) by collecting orphan reads, defined as reads that do not map to the reference genome but have a mate read that maps to the reference genome at regions that flank a breakpoint. For each breakpoint, orphan reads were collected and de novo assembled to breakpoint border sequences. To distinguish between natural insertions and insertions caused by transformation of the native genome, all breakpoint border sequences larger than 30 bp were compared against a plant transformation vector (pCAMBIA-1300) using MegaBLAST algorithm [25]. The result was promising — one breakpoint on chromosome 6 was identified in the LLRice62 sample as a putative insertion site, with a left breakpoint border sequence of 79 bp and a right breakpoint border sequence of 275 bp exhibiting homology to the transformation vector. This is in complete agreement with the information provided by the LLRice62 breeding company (EFSA 2007).

Essentially orphan reads correspond to reads only at the breakpoint, because to qualify it must have a mate read that maps to the native reference. These two paired-end reads are within the same fragment so that they should be close enough to each other. Assembly of the orphan reads result in contigs covering either flank of the inserted region, but not the middle if the insertion is larger than a single read length 100 bp. This means that this approach is unable to detect transgenic insertions less than 100 bp, but hardly can such short sequence produce any important functions. Based on this effort, we would like to extend the underlying methods to investigate not only the breakpoint border sequence but the whole sequence within the inserted region. More importantly, we would like to generalize such approach to deal with GMOs beyond the domain of food and crops.

Statistical methods are also appropriate for detecting synonymous recoding. The more common type of synonymous recoding is to optimize codon usage for high protein expression when moving a protein-coding sequence across species. Usually the foreign gene content is already easier to detect than the more subtle changes in codon usage, however. This can be phrased more quantitatively by noting that the information content or Shannon entropy of an amino acid is much larger than the conditional entropy of a codon given its amino acid, averaged over amino acids. Two less common types of synonymous recoding involve codon minimization, in which certain codons are eliminated from a genome to permit an enlarged genetic code encoding unnatural amino acids, and watermarks, in which synonymous recoding is used to introduce detectable signatures or perhaps to encode hidden information with DNA steganography. Codon minimization is easy to detect given WGS sequence, although it is more difficult in shorter sequences. Watermarks that are contiguous may be detectable if they are sufficiently long and different from native sequences. Distributed watermarks, for example a unique haplotype constructed from natural alleles scattered along a chromosome, can be impossible to detect statistically.

2.5 Considerations

As discussed above, targeted and untargeted detection methods are useful in complementary scenarios: targeted approaches may be favored when the engineered sequences or loci are known, and untargeted methods become essential when these are unknown. Since deep sequencing data is relatively unbiased and agnostic as to the type of engineering, the untargeted approach is more easy to generalize to detect combinations of engineering and to unforeseen types of engineering. Some types of engineering may be difficult to impossible to detect versus the native background, for example distributed watermarks or small indels that could occur naturally and have functional consequences.

In the realm of WGS approaches, perhaps the the biggest limitation is for methods that require use of a reference genome or require a wide array of native genomes to build a model for natural variation. Usually, any species that is a possible host for genetic engineering already has well-characterized genetics, at least one high-quality reference genome, and an increasing number of well-annotated diversity strains. Even as the number of diversity strains increases, however, genome comparative approaches remain unlikely to be definitive in describing a DNA sequence as engineered merely because it does not occur in any of the known strains (provided, of course, that the sequence also does not match the DNA sequence or protein sequence of a foreign source).

What if we don't know which reference genome or genomes to use? Luckily, given sufficient NGS coverage, this problem is within the domain of meta-genomics and environmental genomics. Meta-genomics refers the study of genetic material recovered directly from environmental samples. Meta-genomics has established methods for inferring the species present in an environmental sample that are applicable to host inference in the context of detecting GMOs as well. We will describe how we have used these methods to infer host species in blinded samples, and then use available sequence data for these inferred hosts as part of further analysis.

Another consideration is computational performance. Broadly speaking, systems that take NGS data can be divided into (1) read-first (2) assembly-first approaches. Namely, read-first approaches first match the reads to some references, either by alignment or by k-mer analysis, whereas assembly-first approaches first assemble the reads to contigs, essentially reproducing a genome assembly pipeline and then compare. The method used in Wahler et al. (2013) [24] is considered a read-first approach. In the literature there have been insufficient studies that compare the classification performance of read-first vs. assembly-first methods. However, even for an engineered sample, the sequences are still predominately native. And the native reads really just provide information about the native background host, not whether it has been engineered. Finally, assembly errors themselves can create spurious sequences or non-native junctions that may appear to be evidence of engineering. These errors are more likely if an entire genome assembly is attempted, rather than just assembly of a small number of reads that are marked as suspicious. Consequently, discarding native reads early should both reduce computational burden and also increase classification accuracy.

2.6 Summary

Several DNA-based procedures are suitable for detection of various types of engineering. For the targeted PCR-based methods, sequence information of the modified locus is indispensable. For the untargeted WGS-based methods focusing on sequence differences, a robust reference genome is helpful. Combined approaches of the two might work well — once a difference is revealed, this knowledge may be used to develop a targeted PCR-based detect method to further confirm the identity of the suspicious content or detect it in new samples.

Detecting conventional genetic engineering, where transgenic elements are fused by promoters or terminators with target sequence in between, is quite feasible. But there are also challenges: for genome edited sites that do not carry foreign DNA traces as screening targets, detection and identification methods are limited; for genome sequence difference between a genome assembly and a reference genome, it is often difficult to tell if it is because of assembly artifacts, natural mutations, or real engineering; for identification of how the engineering was done, it is nearly impossible if the engineering was scarless.

Chapter 3

Methods

3.1 Data

IARPA enlisted Testing and Evaluation (T&E) groups to perform genetic engineering on fungal species and generate DNA-Seq data provided as 50 blinded DNA-Seq data sets. We were to (1) classify each sample as native vs. engineered and (2) if engineered, report the host species, the type of engineering, and the and evidence, including the engineered sequence, whether it resided on a plasmid or was integrated, and if integrated the location relative to the host reference genome. Each sequencing data set had approximately 30x coverage by Illumina short-read paired-end sequence, with a read length of 300 nt and an insert size of 400-500 bp, with the paired ends reads often overlapping. Along with the Illumina short reads, T&E also provided Nanopore long reads for a subset of samples. Given the high coverage of the short reads, our Phase 1 pipeline focused on analyzing the short-read data, with long read analysis deferred for planned future work in Phase 2.

3.2 Identifying the host strain

Our first step was to determine the host strain or strains that might exist in our sequencing data. Although a standard mapping approach could work for this purpose, this is computationally expensive. Alternative mapping-free methods that instead use k-mer matching are much more computationally efficient, without much loss of accuracy for host strain identification (personal communication, Steven Salzberg and Ben Langmead). We then tested one such method, Kraken2 [26, 27], using default parameters. This software first builds a k-mer hash table from GenBank genomes with known taxonomic classification, where the key is the minimizer of the k-mer and the value is NCBI Taxonomic Identifier (taxon ID). Kraken2's performance is boosted through several mechanisms. First, the 31-mer minimizers are themselves spaced seeds of 35-mers. Second, these minimizers are used to create a sketch of the true 35-mer distribution. These improvements allow Kraken2 to run with a small memory footprint and faster operation.

The output of Kraken2 is a taxonomic classification of every continuous 35-mer subsequence within a read. Each genome containing that unique k-mer is identified, and the subsequence is then annotated to the least common taxonomic ancestor and assigned a taxon ID. This resulting assignment might correspond to an individual strain if the k-mer happens to be unique to that strain. Otherwise the assignment is made at the lowest taxonomic category in which the k-mer is unique: species, genus, or higher level category. The special taxon '0' is reserved for k-mers that do not map to any known sequences. For the choice of databases, we used NCBI non-redundant nucleotide database (nt) together with the vector and plasmid database (UniVec) to build Kraken2 k-mer indices.

Initially, we intended to use the Kraken2 output to not only detect the native host but also to remove sequences that are perfect match to a native host. When we attempted this, however, we ran into difficulties in interpreting the mapping from k-mers to taxon ID's. For example, how many 0's in a row as a cutoff would we require to call a sequence non-native? What should we do with long foreign gene content where short reads would be entirely within the foreign gene? These could map perfectly to the source organism, yet we do not want to remove them without checking that the organism's entire genome was present, rather than just an isolated gene without whole genome context. How should we account structural variations or breakpoints, which correspond to rearrangements within the native genome? Additional issues arise because a single nucleotide variant and breakpoint both affect all the k-mers within the 35 nt window, even though they are very different types of variation.

After encountering these difficulties, we decided to split apart host species identification from read subtraction: for host species identification, k-mer matching is sufficient, but for read subtraction, mapping is required. Speed-ups are still possible for read subtraction, though, because a custom database can be built for the dominant host inferred from k-mer analysis. To determine the dominant species, the read-level analysis of KRAKEN2 can be used as input for companion software BRACKEN [28], which performs Bayesian re-estimation of species abundance, and all higher levels of taxonomy hierarchy as requested by the user, at the sample level. We therefore used BRACKEN post-processing to identify the host dominant species for each sample. We classified each sample at species and genus level, and simply chose the organism which has the highest abundance as host.

Our annotation of the dominant host species was correct in every case. Some samples contained mixtures of multiple hosts. For these, our Phase 2 plans are to run species identification and read subtraction (described immediately below) iteratively, identifying the major species and their relative frequencies in the sample.

3.3 Mapping-based subtraction of native reads

Our second step was to align the reads against the host reference genomes, and then remove reads that were presumably native based on high sequence identity with the reference. We anticipated that it would take too long and too much memory to map against a non-redundant database using any existing aligner. By determining the host species first, we saved computational resources by mapping only to that single host species, rather than to a much larger non-redundant database.

We downloaded the best-quality genome assemblies for the identified species, labeled as either the “reference” or “representative” in the RefSeq section of GenBank. We used one of the best-performing aligners, Bowtie2 [29], for mapping the reads to the reference. Although HISAT2 is a newer version of this read-mapper, it was optimized for mammalian applications rather than for fungal and bacterial applications, for which Bowtie2 remains well-suited (B. Langmead, personal communication). For paired-end reads, native read-pairs were defined as read-pairs that mapped either concordantly (perfect match to native) or discordantly (sequences match native but with a different distance or orientation, suggesting a structural variation). These native read-pairs were filtered out. In future research, we plan to investigate by retaining the discordant pairs as evidence of structural variation. For reads where neither end mapped, we kept both ends as suspicious. For reads where a single end mapped and the other end did not map, we kept only the non-mapping end as suspicious. Although keeping both ends may provide greater sequence context for the native region from the host if the purposed engineering is integrated, we found in the end that our choice of keeping only the non-mapping read gave a sufficiently long native flanking region for native genome context.

A schematic illustrates the workflow described in Sections 3.2 and 3.3 (Fig. 3-1).

In most samples, this procedure retained only 5-15% of the reads for downstream analysis. The retained fraction is higher than necessary because we used only a single best reference or representative genome assembly for mapping. We anticipate that when we use all available native genomes for a species, the fraction retained should drop substantially to the true fraction of the reads that are engineered. For example, in the highly engineered Sc2.0 genome, this fraction is roughly 10 KB engineered / 12 MB genome, about 0.1% [30]. Of course an engineered nucleotide affects all reads that

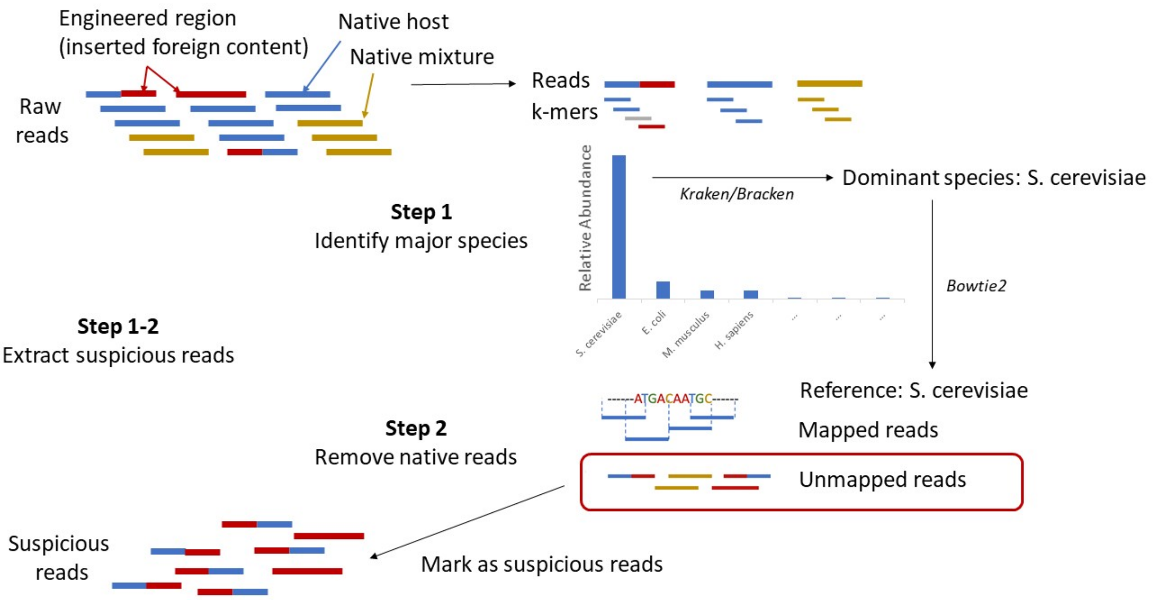

Figure 3-1. Host identification and native read subtraction. We used Kraken2 and Bracken to identify taxonomic sources based on 35-mer subsequences within each read. The top species was chosen as the native host. The corresponding GenBank assembly denoted as the RefSeq "reference" or "representative" was then selected as the reference genome for Bowtie2 mapping. Unmapped reads were sent to assembly.

contain it. Much of the Sc2.0 engineering comes in the form of PCRTags in which approximately 30 nt within a coding domain are synonymously recoded to create a watermark. Since these 30 nt will affect reads starting within roughly a 300 nt window, the result is 10× more suspicious reads than based on recoded sequence along, or approximately 1% for Sc2.0.

The mapping results were also used to estimate the coverage of the host genome. We used a simple estimate calculated as the number of reads mapped to the native host multiplied by the read length and divided by the host genome size. Although more sophisticated methods are available, this method was suitable for our purposes of estimating the relative copy number of suspicious contigs.

3.4 Assembling suspicious reads into suspicious contigs

Our third step was to assemble the suspicious reads into suspicious contigs. Most assemblers use similar algorithms but optimize for different scenarios or needs: memory footprint, CPU time, sample ploidy. MEGAHIT [31] was suggested as a robust short-read assembler appropriate for our scenario of 30× coverage of multiple mini-contigs of primarily fungal or bacterial origin, without complexities of high ploidy or long repeats (personal communication, S. Salzberg). MEGAHIT performs de novo metagenomics assembly using a succinct de Bruijn graph, where all non-branching paths become the output contigs. Reads were assembled successfully at the first attempt. Multiplicity of the assembled contigs was compared with the original sequencing coverage. These comparisons were quite informative: contigs with coverage much greater than the host tended to indicate multi-copy plasmids; contigs with coverage much less than the host suggested dilution of an engineered strain in a native background of the same species, which was possible in the T&E samples.

3.5 Classification of suspicious contigs as native or engineered

The final step was to annotate suspicious contigs as engineered vs. native origin. Sequence analysis was performed by automated BLAST [32] against several databases available from NCBI: non-redundant nucleotide (nt), 6-frame translation vs. non-redundant protein (nr), and UniVec database. Again for pragmatic reasons, only the top-scoring hit was retained for each contig and database. In many cases, the nucleotide hit and the protein hit both pointed to a single foreign content, increasing the confidence in the annotation.

As mentioned above, reads from native genes that happen to be absent from a reference genome can be marked as suspicious and lead to suspicious contigs. BLAST annotation using a full non-redundant database, rather than a single reference genome, should identify these as native, however. We therefore removed any suspicious contigs that matched non-engineered genomes from the host species.

Unfortunately, we were unable to fully automate the removal of contigs matching alternative genomes from the host species because most of the genomes deposited in GenBank lack meta-data describing the biological sample as native vs. engineered. A logical place for this meta-data would be in the NCBI BioSample table, which provisions a unique identifier for each biological sample. Submissions do not require meta-data indicating native vs. engineered, however, and this information is not reported for most samples. A single BioProject corresponds to one or more scientific publications with PubMed identifiers (PMIDs), which can provide more complete information about the biological samples. The PMIDs links are not always included, however, and can be ambiguous because the link is at the BioProject level rather than the BioSample| level. This makes it difficult to match strains described in research publications and assembly IDs in GenBank. Thus, for many of the matches to alternative genomes for a species, manual analysis was required to distinguish between a match to a native gene versus a match to a foreign gene in an engineered host, for example an integrated cloning vector or antibiotic resistance marker.

Subsequent analysis was partially automated and involved manual sequence analysis, including visual inspection of the junctions between regions of the suspicious contig query matching native host sequence and regions of the query matching engineered sequence. Manual analysis was also performed to identify the location of engineering as integrated into the host genome (either based on native flanks or presence of an integrating vector) vs. presence of a plasmid. In certain cases, literature analysis was also used to identify the likely provenance of a strain.

The assembly and annotation steps described in Sections 3.4 and 3.5 are illustrated in a schematic (Fig. 3-2).

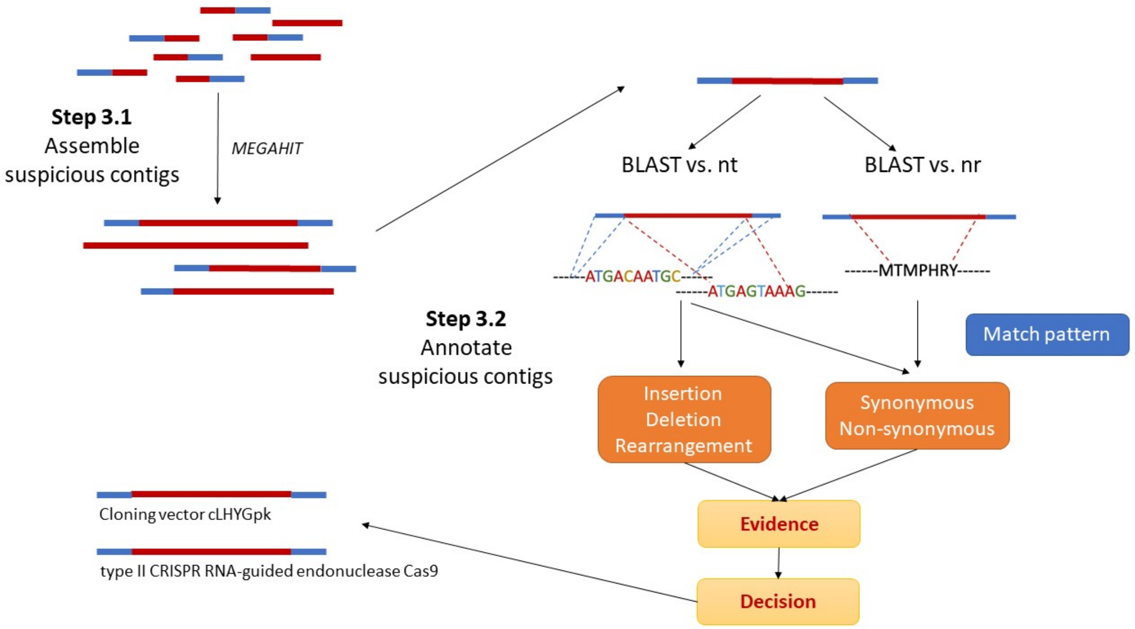

Figure 3-2. Assembly and annotation. Suspicious reads were assembled into contigs with MEGAHIT and then annotated using BLAST against non-redundant nucleotide (nt) and protein (nr) databases. Flanking sequence on a contig that matched a native host provided the context of the integrated engineering relative to the reference. Matches to UniVec and differences in copy number between the suspicious contig and the host genome identified plasmid-based engineering (often higher copy number of the suspicious contig) or dilution (integrating plasmid and lower copy number of the suspicious contig).

3.6 Alternatives and Improvements

3.6.1 Chronology

The methods described here were developed under time pressure. At the start of the FELIX effort in September 2018, our assigned role within the BBN consortium was to provide biological knowledge in the form of databases of non-engineered genomes, obtained from GenBank, and engineered genomes, including the Sc2.0 genomes with sequences designed at JHU [30]. We planned to advise other participants on methods, but not to create our out method. A first round of T&E occurred with sequences provided to BBN in April 2019. Results were disclosed in May-June 2019.

At the June 2019 Face-to-Face meeting organized by IARPA for all FELIX performers, the results we observed from all the groups, including BBN, were disappointing. Most methods had only 70-80% accuracy for what seemed should be an easy classification problem. These methods generally followed the pattern of generating full-genome assemblies, then annotating with existing genome annotation pipelines or with sequence-based features extracted from comparisons of native vs. engineered genome training sets. Unfortunately, these were the only type of methods being developed within the BBN consortium. Notably, all of the funded efforts took a similar approach; no groups were pursuing what appeared to us to be a more efficient and potentially more accurate approach of first removing all the native sequences and then analyzing what could be a much smaller data set.

After discussing our thoughts with BBN, we began work on the reads-first methods described here in June 2019, with the knowledge that a new round of T&E data would be arriving in August 2019 with answers due in September-October 2019. Our goal was to construct a working pipeline end-to-end in time for T&E, rather than to optimize any particular step. As will be presented in a subsequent chapter, our initial pipeline, developed over approximately three months primarily by a single person, out-performed all the other methods, which had been developed for a year by much larger and better resourced groups.

3.6.2 Input data: short vs. long reads

As discussed above, while T&E provided short-read data for all samples, long-read data was only available for a subset of samples. Furthermore, the estimated coverage of the short-read data was higher than the coverage of the long-read data. Short-read technologies remain superior to long-read technologies for error rates. Our impression was that the long-read data was requested by groups influenced by whole genome assembly, in which long reads are important for linking together contigs. We therefore postponed work on long reads.

Focusing on short reads simplified subsequent steps. Meta-genomics methods such as Kraken2 and Bracken often assume similar lengths for each read, which is not true of long-reads. Assembly methods for mixtures of long and short reads remain an active area of research.

When we compared with results from other groups using long-read data (estimated coverage 15x), there were one or two samples in which evidence of engineering was present in the long-read data but not the short-read data. The most likely explanation appears to be incorrect deconvolution of barcodes used for multiplexed sequencing runs. The effort that would be required for a definitive answer may be beyond the resources of the T&E team that generated the data, however.

3.6.3 Mapping to the native reference

For simplicity, reads were mapped to a single native reference genome. This is problematic for two reasons. First, species do not have a true single reference genome because gene content (not just alleles) can vary from strain to strain. For example, the reference genome for *Saccharomyces cerevisiae* was generated from the S288 strain. Gene content in the S288 strain and the CEN.PK strain differ by over 100 genes, however. And, most importantly, they are both non-engineered, although they are laboratory strains that were derived by selection. Our plan, which was successful, was to permit reads for these genes to be marked as suspicious, to build them into suspicious contigs, and then to identify them as native in the BLAST-based annotation step. In future work, it would likely be better to use genomes from multiple strains to identify and filter out native reads.

This brings up the second problem: GenBank has both native and engineered genomes, and there is no clear annotation in meta-data of whether a BioSample (biological specimen from which the sequencing data are derived) is native or engineered. Furthermore, in addition to the native and engineered categories, there is also a derived category, in which cells have been subjected to classical mutagenesis with the mutants then selected for desired properties. Some of the genomes marked as “reference” or “representative” in GenBank are actually derived strains rather than native isolates. The standard reference for *Bacillus subtilis* is a derived strain, for example. Therefore, we also delayed work on a multiple-strain mapping reference until we could reliable identify just native strains, as opposed to derived and engineered strains. We return to the genome curation problem later in Chapter 5.

Chapter 4

Results

Please refer to Appendix A for decisions as well as evidence the JHU pipeline presented, and Appendix B for the sample key table provided by T&E. Summary results and interpretation are provided here.

4.1 Native hosts

Although the JHU pipeline is agnostic as to host species, other classifiers require training data including samples of engineered and native genomes. Before T&E, BBN and IARPA agreed upon three host species: *Saccharomyces cerevisiae* (Sce), *Yarrowia lipolytica* (Yli), and *Pichia pastoris* (Ppa). Depending on personal preference, Sce is better known as either bakers' yeast or brewers' yeast. More genetic engineering has been performed with Sce that with all other eukaryotes put together, at least according to literature publication counts. The fungus Yli has gained recent interest for bioenergy because of its robust lipid metabolism, as indicated by its name. Ppa is yeast used for industrial production because it grows to high density in batch culture. While 'pichia' remains used as a common term, in fact it no longer exists as part of formal taxonomy. Instead, all Pichia strains have been formally re-classified into two closely related species, *Komagataella pastoris* (Kpa) and *Komagataella phaffii* (Kph). This came as a surprise to many of the IARPA program managers and performers and demonstrates the fluidity of taxonomy at the species and genus levels.

The number of assemblies in GenBank varies widely for these hosts (Table 4-I). These numbers are relevant in determining whether classification accuracy depends on the number of genomes available as background to assess natural diversity and to distinguish diversity from engineering.

Species	Taxon ID	Size (Mb)	RefSeq	Total
Saccharomyces cerevisiae	4932	12.2	S288C	854
Yarrowia lipolytica	4952	20.5	CLIB122	22
Komagataella pastoris	4922	9.4	ATCC 28485	33
Komagataella phaffii	460519	9.4	GS115	6

Table 4-I. A summary table of GenBank genome assemblies. RefSeq: reference strain (S. cerevisiase) or representative strain (*Y. lipolitica*, *K. pastoris*, *K. phaffii*) in RefSeq. Total: total number of assemblies in GenBank.

4.2 Host selection

Sequence data was provided for 50 T&E samples. The number of reads available for each sample ranged from about 700,000 to over 2,000,000 (Fig. 4-1). The reads were input to Kraken/Bracken for read-level analysis and sample-level species composition prediction. We selected the highest predicted frequency species as the as host organism (Figure 4-1). The JHU pipeline assigned 16 samples as *S. cerevisiae*, 25 *Y. lipolytica*, 6 *K. phaffii*, and 3 *K. pastoris*. Their identities were all confirmed with the sample key provided.

Some of the *S. cerevisiae* samples had significantly greater read counts and coverage (around 150x) compared to most of the other samples (around 40-50x) (Figure 4-1a, 4-1b). In the Bracken assignment of dominant species (Figure 4-1c), we noticed that some of the native *Y. lipolytica* samples, for example Y105, Y134, and Y154, had only around 50% of the reads assigned. The second major species that co-existed was *Y. deformans*, which accounted for an additional 20% of reads. The reason

Species	Total	Native	Engineered	TP	TN	FP	FN
S. cerevisiae	16	3	13	9	3	0	4
Y. lipolytica	25	6	19	16	6	0	3
K. pastoris	3	3	0	0	3	0	0
K. phaffii	6	3	3	3	3	0	0
Total	50	15	35	28	15	0	7

Table 4-II. Species: known species as provided by IARPA. Total, Native, Engineered: number of samples provided for this species total, without engineering, and with engineering. TP, TN, FP, FN: classification performance, true positive = engineered classified as engineered, true negative = native classified as native, false positive = native classified as engineered, false negative = engineered classified as native. Host species also had to be detected, and in all cases was assigned correctly.

was similar to Ppa case discussed before — taxonomic classification is entirely a human-labeled process, which sometimes places what could be considered different strains into different species. Furthermore, the small number of *Y. lipolytica* genomes available probably under-represents the full diversity of the species. Due to large sequence similarity shared among multiple species within the Yarrowia fugal genus, it is nearly impossible to classify at species level for these samples in our case.

4.3 Suspicious reads and suspicious contigs

For each of the four host species, we downloaded the RefSeq Reference or Representative genome (as of October 2019) (Table 4-I), and then built the corresponding Bowtie2 database, creating four corresponding databases. Reads from each sample were then mapped to the database corresponding to the samples' dominant host. Unmapped reads, specified as SAM flag '4', were collected as suspicious, with about 5-10% of reads classified as suspicious per sample (Fig. 4-2).

While most samples had 5% to 10% of the reads marked as suspicious, the samples Y105, Y134, and Y164 had much higher rates, approximately 90% marked as suspicious. These were all *Y. lipolytica* samples. Possible reasons could include the choice of a

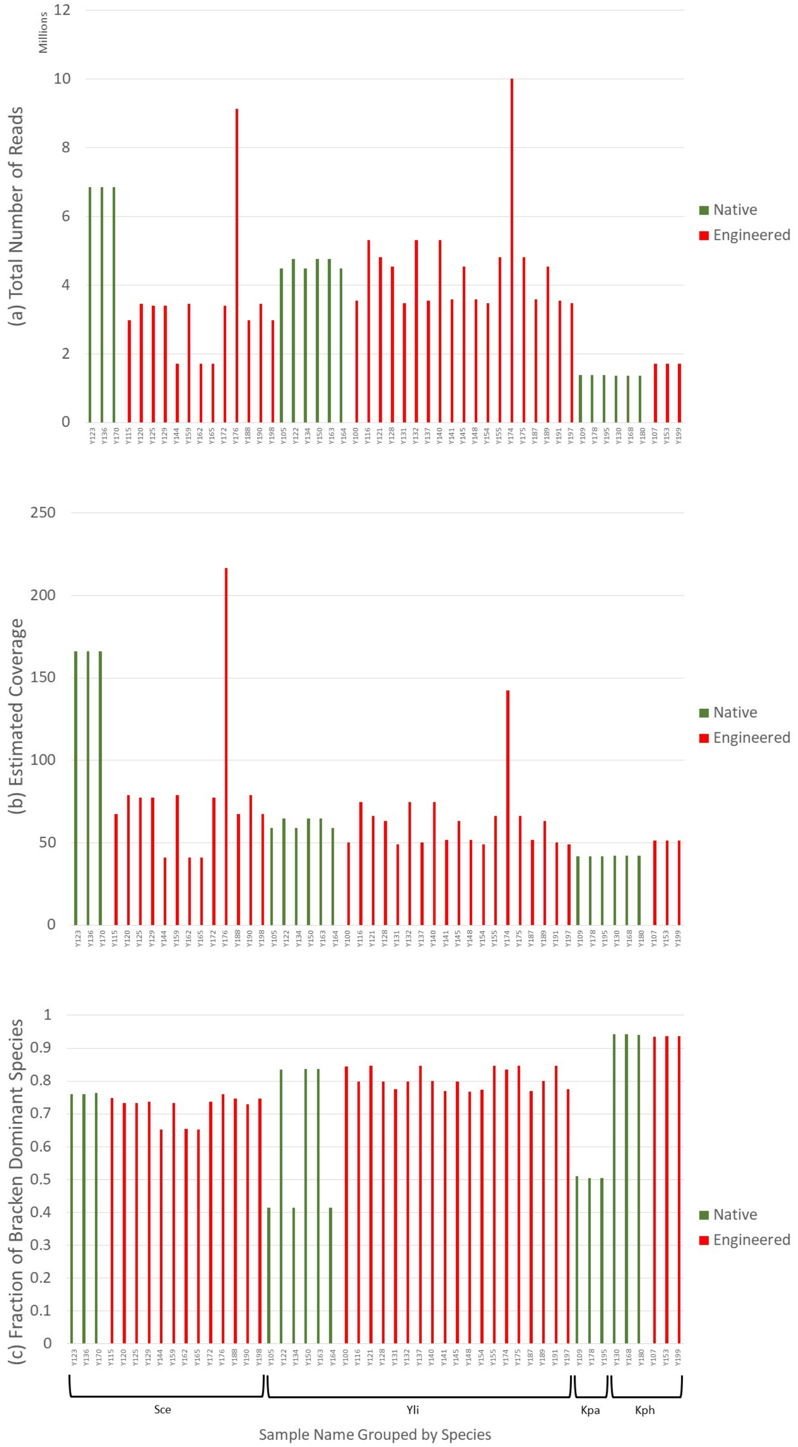

Figure 4-1. Samples are ordered by dominant host species as determined by sc Kraken2/Bracken, and for each predicted host ordered by native (green) and then engineered (red). (a) Total number of read-pairs (mates count as two). (b) Estimated coverage of the host genome. (c) Fraction of read-pairs assigned to the dominant host species.

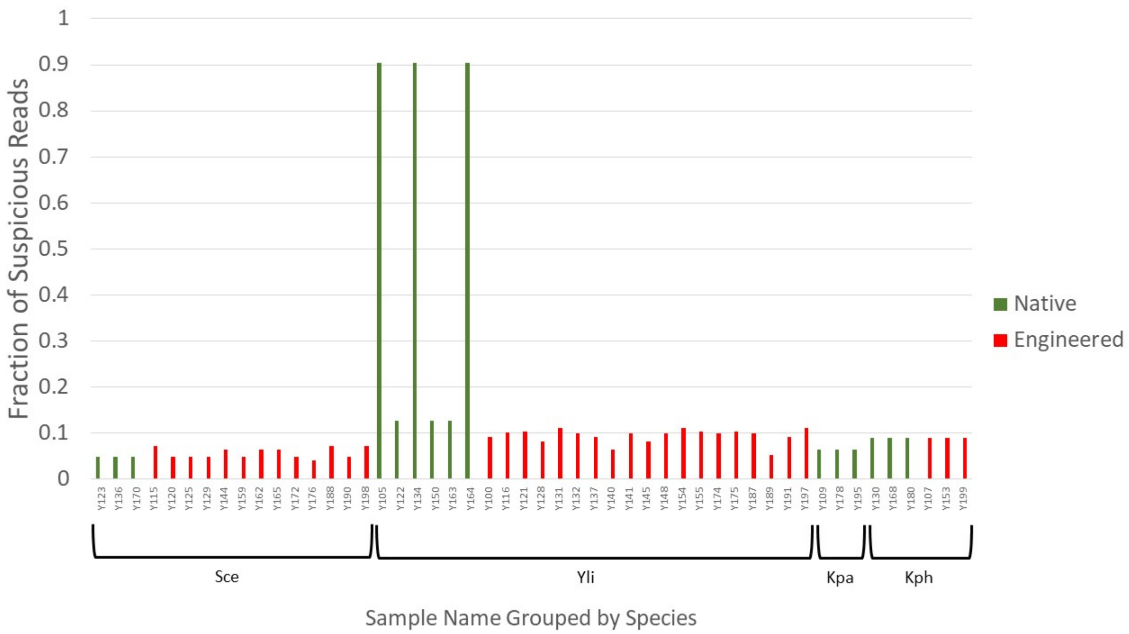

Figure 4-2. Fraction of reads that did not map to the reference genomes, customized for each species. The spikes correspond to the three *Y. lipolytica* samples, Y105, Y134, and Y164, reporting over 90% of the total reads as suspicious.

single best reference for *Y. lipolytica* was not representative enough of these three specific strains. Post-analysis was performed on these three samples by mapping those reads to all 22 *Y. lipolytica* assemblies deposited in GenBank. The unmapped percentage dropped to around 83.5%. We next expanded the native database to include all 42 genomes within the *Yarrowia* genus. At this point only about 5% of the reads were suspicious, comparable to other samples. These results suggest that using the entire native genus database instead of a species-level database could be beneficial when the species-level under-represents the true diversity. We have not implemented this type of expansion yet, however, due to the need to ensure that the assemblies in GenBank correspond to native isolates rather than engineered strains.

These suspicious reads were then sent to assembly using MEGAHIT. Contig coverage was estimated by MEGAHIT from k-mers, and the relative coverage was defined as the ratio of the contig coverage estimated from k-mers to the host coverage estimated from read mapping.

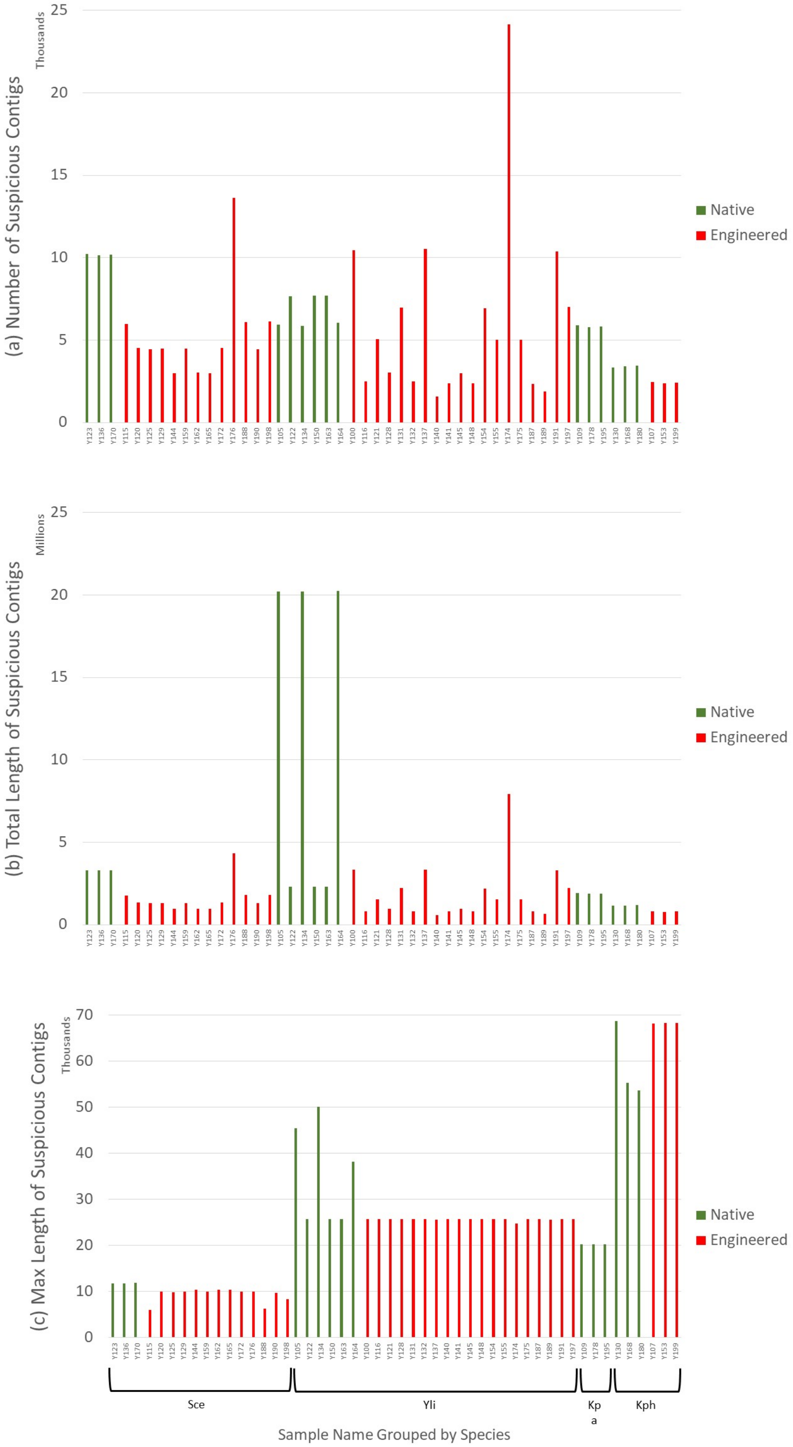

Figure 4-3. Statistics of the suspicious contigs. (a) Total number of suspicious contigs assembled. (b) Total length of suspicious contigs, as an estimate of the length of the suspicious region. (c) Maximum length of suspicious contigs, as a measure of the assembly quality.

4.4 Contig annotation and classification

The resulting assembly were BLAST against non-redundant nucleotide, protein and vector databases. We noted that of the many suspicious contigs we obtained, only a handful few had hits in the database, which allowed for manual inspection at this point. In fact, after this post-processing, we retained only 10-20 contigs per sample with meaningful annotation. The decision on whether a sample was engineered was made following the method described in Section 3.4. Figure 4-4 shows an example contig with foreign genetic content integrated into the host genome.

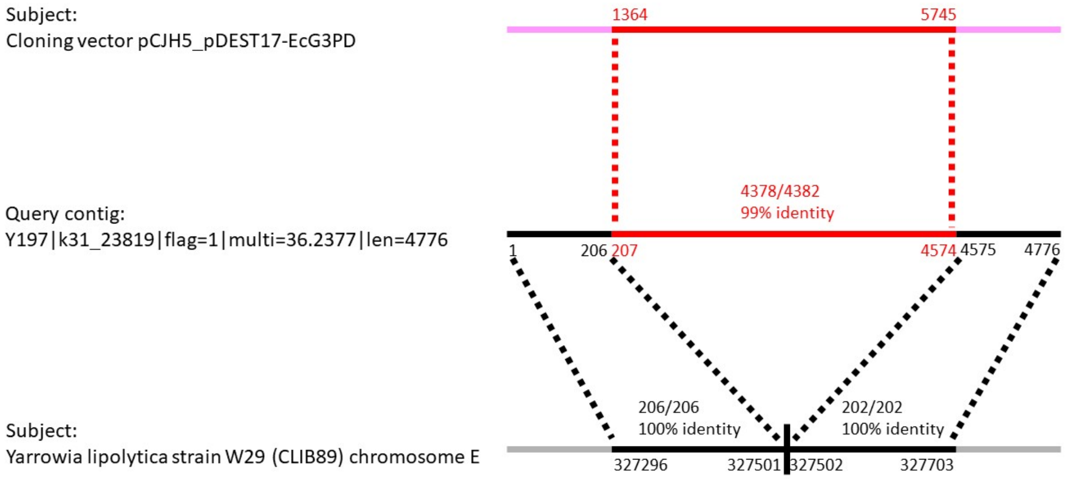

Figure 4-4. This diagram shows a typical instance of foreign gene content integrated into the host genome. Y197, a *Y. lipolytica* sample, has a suspicious contig of length 4776. flag=1 means this is an isolated contig with in-degree and out-degree equals to 0. multi means the multiplicity or the depth of the contig. Annotation with BLAST mapped the flanking regions (black) perfectly to *Y. lipolytica* chromosome E, in which the junction is the site of the insertion. The sequence in the middle (red) matches a cloning vector with high identity. The chimeric structure of this contig, together with coverage similar to the host genome, indicates an engineered insertion into the nuclear genome.

4.5 Other classifiers

We note that there are no standard methods that could serve as benchmarks for classification performance. As part of the overall BBN effort, however, several other types of classifiers were developed. We describe results for two general strategies: expert knowledge, represented by targeted search, and machine learning, represented by FAST-NA [33]. Targeted search relied on expert knowledge to define a list of DNA sequences used in engineering. These signature sequences included cloning vectors and selectable markers. The machine learning classifier was trained on native genomes to build a Bloom filter (similar to a hash table thats permits collisions) of k-mers that occur in non-engineered genomes, then flags as suspicious reads containing k-mers that are not in the hash table.

4.5.1 Targeted Search

The short-read targeted search approach used known engineered targets identified during targeted panel design, and re-used them directly on short-read data. First, Illumina adapter and low quality sequences were trimmed off. Next, BWA [34] aligned the reads first to *S. cerevisiae*, *Y. lipolytica*, *K. pastoris*, and *K. phaffi* reference genomes, and secondly to known engineered targets. Targeted Search then filtered reads that aligned to both a reference genome and an engineering signature using a PostgreSQL database, with a goal of capturing only reads that crossed an integration junction. For Batch 2 of T&E samples, an additional filter step was added to validate a junction point in the read between the engineering signature and the reference genome, reducing false positives from multi-mapping regions.

An additional use of the targeted search database was for an experimental approach to use wet-lab methods to identify the presence of these signature sequences in physical samples.

4.5.2 FAST-NA

FAST-NA first created four contrasting Bloom filters, one for each taxon, by processing native assemblies from GenBank. Then for an unknown sample, FAST-NA first assembled raw reads using ABySS [35]. The resulting contigs were then aligned against four yeast reference genomes using BWA [34]. The best alignment was found so the host was therefore determined. All contigs that belonged to a particular taxon were merged together into a single FASTA file. The associated contigs were then run through FAST-NA using the contrasting Bloom filter corresponding to the host species. The unique k-mers that did not have a hit to the Bloom filters were reconstructed into contiguous regions for manual inspection via BLAST. Engineering decisions were made by visualizing a histogram of region of interest (ROI) lengths in comparison to that for native samples of that taxon, which was considered to be the background noise level.

4.6 Classification performance

Of the 50 unknown samples provided, 35 were revealed to be positives (of which 28 were correctly classified by the JHU pipeline) and 15 were revealed to be negatives (all correctly classified by the JHU pipeline), with overall accuracy = 0.86, sensitivity = 0.8, and specificity = 1 (Table 4-II).

4.6.1 Performance across types of engineering

Four of the known positives had two types of engineering, with 39 total instances to detect. Point mutations, native indels, and structural variations were more difficult to detect than insertion of foreign genetic content.

As shown in Table 4-III, the JHU pipeline performs readily in detecting inserted foreign gene content. The capability to detect minor genetic alterations, such as

Pattern D-P-S-I	# Strains	Sensitivity JHU Pipeline	 FAST-NA	 Targeted Search
N-Y-N-N	8	0.500	0.375	0.500
Y-N-N-N	6	0.667	0.000	0.000
Y-N-Y-N	3	0.667	0.000	0.333
N-N-N-Y	6	1.000	0.833	1.000
N-N-Y-Y	9	1.000	0.556	0.556
Y-N-Y-Y	3	1.000	0.000	0.000

Table 4-III. Performance is categorized by engineering type. D-P-S-I stands for Deletion-PtMut-StrutVar-Insertion, where Deletion = part of the host genome is deleted or truncated; PtMut = point mutation or small edits induced by CRISPR; StructVar = mislocalized/recombinant structural variant involving native genes; Insert = introduction of foreign gene content. Patterns were organized in combinations of these engineering types, in the order mentioned above. For instance, a pattern of N-Y-N-N means only PtMut was present, while the other three were absent.

single base changes, is much limited. This is because during the stage of Bowtie2's mapping to a native genome, reads with small edits may be still be aligned. To address some of these issues, we are planning to build a reverse comparison where a native genome is used as query against the DNA sequencing reads of an unknown sample as subject. This may be an effective approach to identify gene deletions, synonymous recoding, and other small edits associated with functional relevance or watermarks. It is still questionable, however, if these types of changes will be recognizable above the background level of natural variation. Many yeast and bacterial species vary substantially in gene content at the strain level, possibly making it unclear whether a deletion is native or engineered. Watermarks created by synonymous recoding may actually be easier to detect with a dedicated method if nucleotide sequences are conserved among strains that carry a particular gene.

4.6.2 Performance across host species

As shown in Figure 4-5a, the JHU pipeline performed well across all host species, suggesting that the generalizable approach fulfilled its design goals. The small performance differences between species, with better performance for *Y. lipolytica*, *K. phaffii*, and *K.pastoris* than for *S. cerevisiae*, was due to the type of engineering rather than to the host species: the engineering in *S. cerevisiae* included more subtle single-nucleotide changes without introduction of foreign genes, whereas the the other species were engineered primarily with foreign gene content or cloning vectors.

4.6.3 Comparison with other classifiers

As shown in both Figure 4-5 and 4-III, JHU pipeline performed the best, with the highest accuracy and sensitivity on the two major species *S. cerevisiae* (16) and *Y. lipolytica* (25). The performance on other two minor species is all perfect. Our JHU pipeline also performed better than FAST-NA and Targeted Search when looking at different engineering signatures. Specially, JHU pipeline had perfect sensitivity in terms of detecting insertion and foreign gene content. To certain extent, it can detect deletion and structural variant, which are more limited in the other two, given reads covering the breakpoint junction were successfully captured in the mapping step. All of the three methods suffer greatly in detecting point mutation or small edits.

4.7 Computational requirements

4.7.1 Identifying the host species

CPU requirements were moderate, 5-10 minutes estimated for typical 30x coverage sequence data. Data storage was primarily for the one-time build of the Kraken2 indices. Construction of a Kraken2 standard database required approximately 100 GB of disk space. Around 50 GB of RAM was required to hold the database. Output size was approximately the same size as the input as one line of text was generated per read or read-pair.

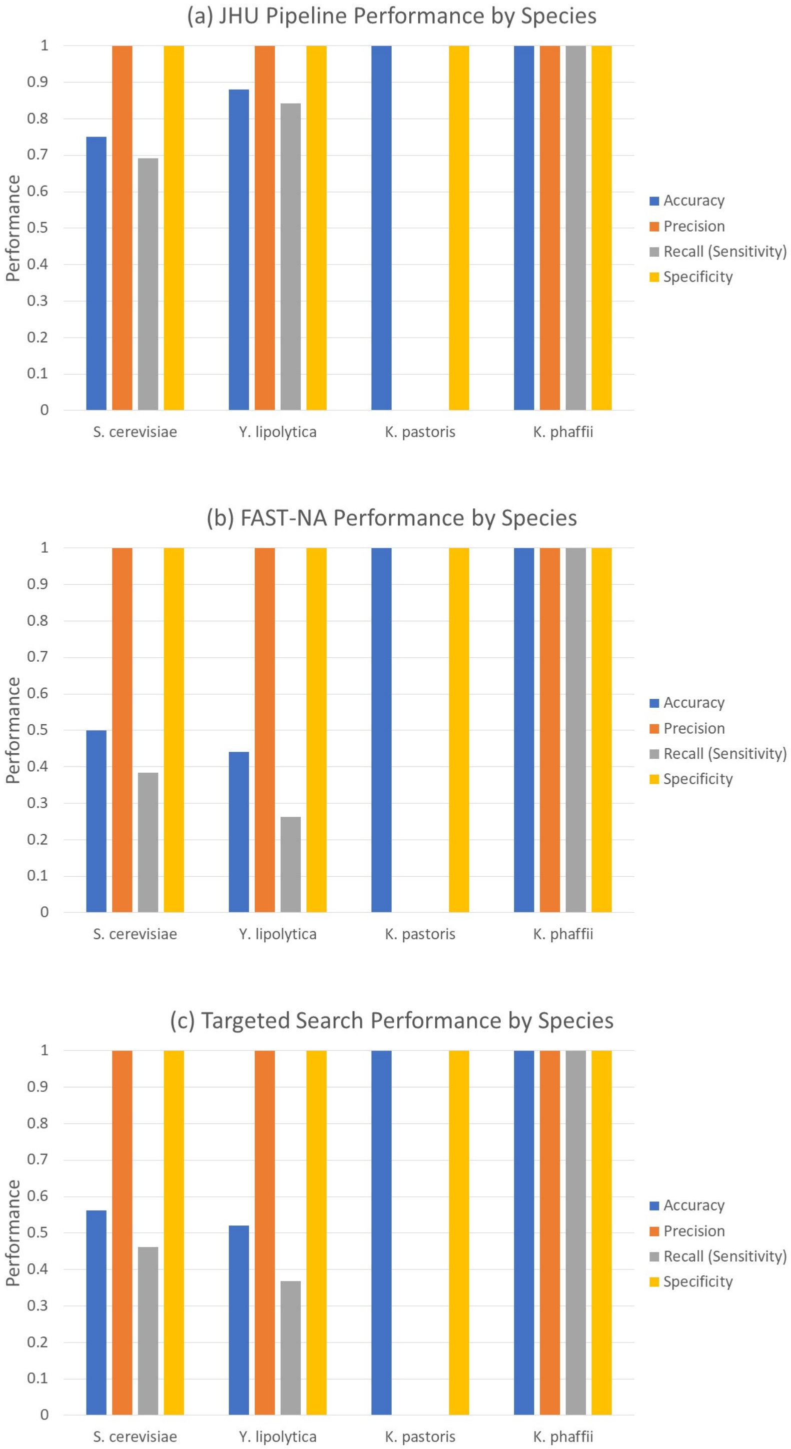

Figure 4-5. Performance is categorized by species, where each colored bar shows accuracy (blue), precision (orange), recall or sensitivity (grey), and specificity (yellow). Bars for precision and recall are absent for *K. pastoris* because all samples were native, making these quantities undefined.

4.7.2 Mapping native reads

CPU requirements were minimal, around 1 hour per 30x coverage data set. Disk space to hold the FM indices used in Bowtie2 increased linearly with respect to the input. Memory requirements peaked around 5 GB, which is available on most modern PCs. Output size was about 10% of input size, corresponding to the fraction of reads marked as suspicious.

4.7.3 Assembly and annotation

Assembly was quite fast, 10–15 min per sample. Annotation required more time, 30-60 min per sample for BLAST searches versus non-redundant databases. This could be reduced by better approaches for removing native reads, for example the iterative approach discussed for Yarrowia samples. We anticipate that future implementations will need 1-5 min for annotation. We downloaded the offline pre-indexed non-redundant databases (nt/nr), which together took up about 500 GB of disk space. Memory usage was around 15 GB. Output size was relatively small — only the annotation from the top BLAST hit was selected for each contig, and typically around 10-20 contigs had meaningful annotations for each sample.

Chapter 5

Discussion

We have presented a method that builds on robust sequence analysis and metagenomics computational infrastructure to predict the presence or absence of genetic engineering from DNA sequence data. While there are no standard methods for this application, our methods were developed in the context of a joint effort pursuing several types of approaches. Our method had two fundamental differences from other methods developed for this effort. First, we filtered out native and near-native reads at the start, and then assembled the remaining suspicious reads into suspicious contigs. The other methods first assembled all the reads and then identified suspicious regions. The benefit of our approach is far lower CPU requirements, and possible additional benefits in avoiding assembly errors that may be confused with engineering. Second, rather than using training examples, our methods used traditional sequence annotation pipelines for classification. This approach maintained good performance for species with insufficient examples of native and engineered genomes for traditional machine learning classifiers. In an assessment with 50 blinded Testing and Evaluation samples, our method performed better than these other types of methods, with 0.86 accuracy, 0.8 sensitivity, and perfect specificity and precision.

Unlike machine learning methods that learn discriminating features from training sets of known positives and known negatives, the JHU pipeline draws conclusions by inferring engineering from sequence that differs from native genomes and by subsequent annotation of the suspicious content. Our interpretation is that the JHU pipeline in general performed at least as well as methods based on machine learning. We correctly identified essentially all engineering that involved foreign gene content or vectors. However, the major challenge is that we have difficulty with engineering involving only host genome sequence, such as point mutations, structural variants, and deletions.

We note that when introducing new functions to genomes, the types of engineering that this approach was able to detect, namely foreign gene content, appears to be the most important to detect. Moreover, some examples of engineering in the T&E samples, such as point mutations, may be fundamentally impossible to distinguish from natural variations — it is likely that their sequences are completely the same.

We also note that our GRAIP design goals (generalization, re-use, automation, interpretation, performance) were for the most part achieved.

The results from Phase 1 suggest a path for improving classification performance for Phase 2, outlined below and already underway.

5.1 Native genome curation in GenBank

An important assumption is that the reference genomes used to filter out the native reads are indeed native. However, many of the GenBank genomes are in fact engineered, and even for those marked as “reference” or “representative”, some are “derived” (which means mutagenized and then selected), and some are “engineered”. The BioSample submission process does not enforce an explicit annotation of native diversity versus derived or engineered. In Phase 2, we will use expert curation to classify genome assemblies from target species as native or engineered, which could in turn be used to develop better automatic classifiers. One interesting early observation is that some genomes annotated as “native isolates” have sequence features that resemble engineering; these are primarily hospital isolates, and the features may be antibiotic resistance markers acquired by natural processes and selected for in the hospital setting.

For a select number of species, it should be possible to read all the publications associated with the corresponding genomes. Curated genomes could then serve as a gold standard for selecting native genomes. It might then be possible to train a classifier for native versus engineered based on full text analysis and natural language processing of BioSample and BioProject descriptions and journal articles linked through PubMed IDs. A weakness of this approach is that many GenBank assemblies have no PubMed links. Even with PubMed links, an assembly can be difficult to assign definitively. A publication may describe a mixture of native and engineered genomes, for example, making it unclear which linked genomes belong in which category. A simple alternative approach is to use methods like "targeted search" to exclude genomes, and to require "native isolate" in the BioSample annotation. These approaches could reduce the number of genomes available, which underestimates native diversity.

Once a subset of genomes are identified as likely native, these could be used to improve methods by filtering out more native reads. As described in the results, using only a single reference genome can result in 90% of reads being flagged as suspicious; expanding the native database to include other species-level and genus-level assemblies can improve performance.

5.2 Long-read pipeline

In Phase 1, we used only Illumina short reads as input. The reason, as mentioned before, is that we believed 30× coverage would be high enough to cover every foreign-native-foreign or native-foreign-native junction. We also had practical constraints on the time available to build a classifier. In Phase 2, we will incorporate Nanopore long reads as well. Because of the intrinsic error rate in long reads, k-mer approaches may not work well for host identification. Thus, we will likely continue to use the short reads to determine the dominant host species. We plan instead to use Minimap2 [36] to map long reads to native genomes (analogous to the use of Bowtie2 for short reads) and then retain suspicious reads that do not map.

Combining the suspicious short reads and long reads, we plan to use MaSuRCA [37] the hybrid assembler to construct assemblies from the both types of reads. The resulting suspicious contigs will then be annotated.

5.3 Iterative mapping for complex samples

To deal with mixed samples (samples containing multiple species of host), our plan is to perform iterative host identification and native read removal. The dominant species could be identified, its reads filtered out, and the remaining suspicious reads re-analyzed to identify a new dominant species.

We plan to use Kraken2/Bracken to identify the major host, followed by removing reads that are native to that host. In the next iteration, we will re-analyze the remaining reads using Kraken2/Bracken, which will give the next most abundant host. Reads mapping to this host will then be removed. The Kraken2/Bracken/Bowtie2 for short-read, or Kraken2/Bracken/Minimap2 for long-read, would then be repeated to iteratively remove additional native species. This could be repeated until falling below a threshold based on a criterion such as fraction of reads remaining, estimated coverage of the dominant species genome, or re-appearance of a previous dominant species.

5.4 Annotation automation

Although we have relied on expert analysis of BLAST output for the final decisions about engineering, the process that proved most useful could be automated in part or whole. The typical pattern of a suspicious contig involves a junction between a shorter native flanking region and a longer engineered region. The BLAST output usually placed the long engineered region at the top of the list of matches because longer sequence matches give higher scores. The shorter flanking regions appeared further down the list, or sometimes not at all if there were many matches to the engineered region.

To automate this step, we propose a simple greedy algorithm. We will use BLAST as before for a suspicious contig (the query) versus the non-redundant nucleotide database (the subject). The top hit will be recorded and used to annotate the matching region of the query. Next, we mask the matched region and re-run BLAST. This should provide a robust approach for automated annotation.

5.5 Dedicated predictors for complex samples

Meanwhile we will implement dedicated predictors for types of engineering that may not be readily detected. These include gene deletion (possibly by suing the host genome as the query and the sequencing data as the subject), identification of synonymous recoding (identifying reads that are perfect matches to host protein but poor matches to host DNA), and other more subtle watermarks of engineering. Watermarks based on synonymous recoding could potentially be identified as regions where the DNA does not match the host but the protein sequence it encodes matches exactly. Tests with the Sc2.0 genome have been successful in identifying PCRtags, which are synonymous recoding watermarks that can be rapidly checked by PCR [30]. Engineered stop codons in non-essential genes could be difficult to identify if similar variations occur at a sufficient frequency in native isolates. Similarly, engineered deletions of native genes could also be difficult to identify because gene content can vary substantially for different strains of the same species.

One mitigating factor is that loss-of-function created by stop codons or deletions seems less likely to be dangerous than gain-of-function engineering. Thus, while detecting loss-of-function may be difficult or impossible in the background of natural variation, methods that are able to detect gain-of-function (foreign gene insertions) and gain-of-information (watermarks) may already detect the most important types of variation. The methods that we have presented achieve these goals.

Chapter 6

Conclusion

We describe results of a computational system designed to detect engineering from DNA sequencing of biological samples, including automated identification of host strains and detection of foreign gene content, host structural variation and smaller edits, and watermarks. Our initial system applied to blinded samples provides excellent identification of foreign gene content, the changes most likely to be functional. We have less ability to detect structural variation and small indels and SNPs produced by genetic engineering. Future work will focus on improved methods for detecting synonymous recoding (for watermarks and recoding) and for distinguishing engineered sequence from natural variation.

References

1. Gargis, A. S., Cherney, B., Conley, A. B., McLaughlin, H. P. & Sue, D. Rapid detection of genetic engineering, structural variation, and antimicrobial resistance markers in bacterial biothreat pathogens by nanopore sequencing. *Scientific Reports* **9,** 1–14 (2019).

2. Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nature Reviews Genetics* **17,** 333 (2016).

3. Wick, R. R., Judd, L. M., Gorrie, C. L. & Holt, K. E. Completing bacterial genome assemblies with multiplex MinION sequencing. *Microbial Genomics* **3** (2017).

4. Lu, H., Giordano, F. & Ning, Z. Oxford Nanopore MinION sequencing and genome assembly. *Genomics, Proteomics & Bioinformatics* **14,** 265–279 (2016).

5. Besser, J., Carleton, H. A., Gerner-Smidt, P., Lindsey, R. L. & Trees, E. Next-generation sequencing technologies and their application to the study and control of bacterial infections. *Clinical Microbiology and Infection* **24,** 335–341 (2018).

6. Laehnemann, D., Borkhardt, A. & McHardy, A. C. Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction. *Briefings in Bioinformatics* **17,** 154–179 (2016).

7. Stoesser, N., Sheppard, A. E., Peirano, G., Sebra, R. P., Lynch, T., Anson, L. W., Kasarskis, A., Motyl, M. R., Crook, D. W. & Pitout, J. D. First report of blaIMP-14 on a plasmid harboring multiple drug resistance genes in Escherichia coli sequence Type 131. *Antimicrobial Agents and Chemotherapy* **60,** 5068–5071 (2016).

8. Forde, B. M., Zakour, N. L. B., Stanton-Cook, M., Phan, M.-D., Totsika, M., Peters, K. M., Chan, K. G., Schembri, M. A., Upton, M. & Beatson, S. A. The complete genome sequence of Escherichia coli EC958: a high quality reference sequence for the globally disseminated multidrug resistant E. coli O25b: H4-ST131 clone. *PloS One* **9** (2014).

9. Bashir, A., Klammer, A. A., Robins, W. P., Chin, C.-S., Webster, D., Paxinos, E., Hsu, D., Ashby, M., Wang, S., Peluso, P., Sebra, R., Sorenson, J., Bullard, J., Yen, J., Valdovino, M., Mollova, E., Luong, K., Lin, S., LaMay, B., Joshi, A., Rowe, L., Frace, M., Tarr, C. L., Turnsek, M., Davis, B. M., Kasarskis, A., Mekalanos, J. J., Waldor, M. K. & Schadt, E. E. A hybrid approach for the automated finishing of bacterial genomes. *Nature Biotechnology* **30,** 701 (2012).

10. Goldstein, S., Beka, L., Graf, J. & Klassen, J. L. Evaluation of strategies for the assembly of diverse bacterial genomes using MinION long-read sequencing. *BMC Genomics* **20,** 23 (2019).

11. Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research* **25,** 3389–3402 (1997).

12. International Gene Synthesis Consortium and others. Harmonized screening protocol; gene sequence and customer screening to promote biosecurity (2009).

13. Carter, S. R. & Friedman, R. M. DNA synthesis and biosecurity: Lessons learned and options for the future. *J Craig Venter Institute, La Jolla, CA* (2015).

14. Allen, J. E., Gardner, S. N. & Slezak, T. R. DNA signatures for detecting genetic engineering in bacteria. *Genome Biology* **9,** R56 (2008).

15. Gachet, E., Martin, G., Vigneau, F. & Meyer, G. Detection of genetically modified organisms (GMOs) by PCR: a brief review of methodologies available. *Trends in Food Science & Technology* **9,** 380–388 (1998).

16. Grohmann, L., Keilwagen, J., Duensing, N., Dagand, E., Hartung, F., Wilhelm, R., Bendiek, J. & Sprink, T. Detection and identification of genome editing in plants: challenges and opportunities. *Frontiers in Plant Science* **10** (2019).

17. EFSA Panel on Genetically Modified Organisms (GMO). Scientific opinion addressing the safety assessment of plants developed using Zinc Finger Nuclease 3 and other Site-Directed Nucleases with similar function. *EFSA Journal* **10,** 2943 (2012).

18. Stevanato, P. & Biscarini, F. Digital PCR as new approach to SNP genotyping in sugar beet. *Sugar Tech* **18,** 429–432 (2016).

19. Huggett, J. F., Cowen, S. & Foy, C. A. Considerations for digital PCR as an accurate molecular diagnostic tool. *Clinical Chemistry* **61,** 79–88 (2015).

20. Broccanello, C., Chiodi, C., Funk, A., McGrath, J. M., Panella, L. & Stevanato, P. Comparison of three PCR-based assays for SNP genotyping in plants. *Plant Methods* **14,** 28 (2018).

21. Lusser, M., Parisi, C., Plan, D. & Rodriguez-Cerezo, E. *New plant breeding techniques: state-of-the-art and prospects for commercial development* (Publications Office of the European Union Luxembourg, 2011).

22. Kyndt, T., Quispe, D., Zhai, H., Jarret, R., Ghislain, M., Liu, Q., Gheysen, G. & Kreuze, J. F. The genome of cultivated sweet potato contains Agrobacterium T-DNAs with expressed genes: an example of a naturally transgenic food crop. *Proceedings of the National Academy of Sciences* **112,** 5844–5849 (2015).

23. Nordstrom, K. J., Albani, M. C., James, G. V., Gutjahr, C., Hartwig, B., Turck, F., Paszkowski, U., Coupland, G. & Schneeberger, K. Mutation identification by direct comparison of whole-genome sequencing data from mutant and wild-type individuals using k-mers. *Nature Biotechnology* **31,** 325 (2013).

24. Wahler, D., Schauser, L., Bendiek, J. & Grohmann, L. Next-generation sequencing as a tool for detailed molecular characterisation of genomic insertions and flanking regions in genetically modified plants: a pilot study using a rice event unauthorised in the EU. *Food Analytical Methods* **6,** 1718–1727 (2013).

25. Zhang, Z., Schwartz, S., Wagner, L. & Miller, W. A greedy algorithm for aligning DNA sequences. *Journal of Computational Biology* **7,** 203–214 (2000).

26. Wood, D. E. & Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biology* **15,** R46 (2014).

27. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *Genome Biology* **20,** 257 (2019).

28. Lu, J., Breitwieser, F. P., Thielen, P. & Salzberg, S. L. Bracken: estimating species abundance in metagenomics data. *PeerJ Computer Science* **3,** e104 (2017).

29. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nature Methods* **9,** 357 (2012).

30. Richardson, S. M., Mitchell, L. A., Stracquadanio, G., Yang, K., Dymond, J. S., DiCarlo, J. E., Lee, D., Huang, C. L. V., Chandrasegaran, S., Cai, Y., Boeke, J. D. & Bader, J. S. Design of a synthetic yeast genome. *Science* **355,** 1040–1044 (2017).

31. Li, D., Liu, C.-M., Luo, R., Sadakane, K. & Lam, T.-W. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* **31,** 1674–1676 (2015).

32. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. & Madden, T. L. BLAST+: architecture and applications. *BMC Bioinformatics* **10,** 421 (2009).

33. Roehner, N. Personal communication. Email: nicholas.roehner@raytheon.com. Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, United States.

34. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25,** 1754–1760 (2009).

35. Simpson, J. T., Wong, K., Jackman, S. D., Schein, J. E., Jones, S. J. & Birol, I. ABySS: a parallel assembler for short read sequence data. *Genome Research* **19,** 1117–1123 (2009).

36. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34,** 3094–3100 (2018).

37. Zimin, A. V., Marçais, G., Puiu, D., Roberts, M., Salzberg, S. L. & Yorke, J. A. The MaSuRCA genome assembler. *Bioinformatics* **29,** 2669–2677 (2013).

Appendix A

JHU Pipeline Decisions and Evidence

Sample	Engineered?	Detected?	What was Detected?
Y100	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y105	no	no	
Y107	yes	yes	MG883678.1, Expression vector pCPC66;WP_000027060.1 MULTISPECIES: class A beta-lactamase TEM-181 [Bacteria][Archaea] YP_001409239.1 b-lactamase (plasmid) [Escherichia sp. Sflu5]
Y109	no	no	
Y115	yes	no	
Y116	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y120	yes	yes	
Y121	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y122	no	no	

Y123	no	no	
Y125	yes	yes	
Y128	yes	yes	KU378202.1,Cloning vector pYL2;KU378203.1 Cloning vector pYL2, complete sequence
Y129	yes	yes	JN160804.1,Cloning vector pRS316-1B9
Y130	no	no	
Y131	yes	yes	
Y132	yes	no	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y134	no	no	
Y136	no	no	
Y137	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y140	yes	no	
Y141	yes	yes	WP_007048783.1 tetracycline resistance MFS efflux pump, partial [Subdoligranulumvariabile];WP_010922251.1 type II CRISPR RNA-guided endonuclease Cas9 [Streptococcus pyogenes] NP_269215.1 hypothetical protein SPy_1046 [Streptococcus pyogenes M1 GAS]
Y144	yes	yes	
Y145	yes	yes	MG252999.1 Synthetic construct green fluorescent protein gene, complete cds
Y148	yes	yes	WP_010922251.1 type II CRISPR RNA-guided endonuclease Cas9 [Streptococcus pyogenes] NP_269215.1 hypothetical protein SPy_1046 [Streptococcus pyogenes M1 GAS]
Y150	no	no	

Y153	yes	yes	U14125.1 Cloning vector pSG926, HIS4-based plasmid, complete sequence
Y154	yes	no	
Y155	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y159	yes	yes	XM_013480424.1 Eimeria maxima hypothetical protein, conserved partial mRNA
Y162	yes	yes	
Y163	no	no	
Y164	no	no	
Y165	yes	yes	
Y168	no	no	
Y170	no	no	
Y172	yes	yes	WP_000027060.1 MULTISPECIES: class A beta-lactamase TEM-181 [Bacteria][Archaea] YP_001409239.1 b-lactamase (plasmid) [Escherichia sp. Sflu5]
Y174	yes	yes	
Y175	yes	yes	WP_120322739.1 aminoglycoside O-phosphotransferase APH(4)-Ia [Bacillus subtilis]
Y176	yes	no	
Y178	no	no	
Y180	no	no	
Y187	yes	yes	WP_010922251.1 type II CRISPR RNA-guided endonuclease Cas9 [Streptococcus pyogenes] NP_269215.1 hypothetical protein SPy_1046 [Streptococcus pyogenes M1 GAS]
Y188	yes	no	
Y189	yes	yes	MG252999.1 Synthetic construct green fluorescent protein gene, complete cds

Y190	yes	yes	WP_000027060.1 MULTISPECIES: class A beta-lactamase TEM-181 [Bacteria][Archaea] YP_001409239.1 b-lactamase (plasmid) [Escherichia sp. Sflu5]
Y191	yes	yes	U59231.1 Cloning vector cLHYGpk, complete sequence;XP_003071652.1 hygromycin-B [synthetic construct]
Y195	no	no	
Y197	yes	yes	WP_120322739.1 aminoglycoside O-phosphotransferase APH(4)-Ia [Bacillus subtilis]
Y198	yes	no	
Y199	yes	yes	U14125.1 Cloning vector pSG926, HIS4-based plasmid, complete sequence

Table A-I. Decisions and evidence. The second column is from T&E Sample Key, revealed after our decisions were submitted. Evidence corresponds to BLAST annotations of suspicious contigs.

Appendix B

T&E Phase 1 Batch 2 Sample Key

Sample	Organism	Engineered?	Which Gene was Engineered?
Y100	Y. lipolytica	yes	v5 epitope
Y105	Y. lipolytica	no	
Y107	P. pastoris	yes	mislocalized GAP promoter
Y109	K. pastoris	no	
Y115	S. cerevisiae	yes	point mutation in HIS3
Y116	Y. lipolytica	yes	truncate NADK2, YALI0E27874g
Y120	S. cerevisiae	yes	m13_origin
Y121	Y. lipolytica	yes	ura3 distruption
Y122	Y. lipolytica	no	
Y123	S. cerevisiae	no	
Y125	S. cerevisiae	yes	m13_origin
Y128	Y. lipolytica	yes	mislocalized leu2
Y129	S. cerevisiae	yes	m13_origin
Y130	P. pastoris	no	
Y131	Y. lipolytica	yes	deletion of gene YALI0A19844
Y132	Y. lipolytica	yes	truncate NADK2, YALI0E27874g
Y134	Y. lipolytica	no	
Y136	S. cerevisiae	no	
Y137	Y. lipolytica	yes	v5 epitope
Y140	Y. lipolytica	yes	truncate NADK2, YALI0E27874g
Y141	Y. lipolytica	yes	Mutation to stop codons in open reading frame (ORF) of the hap4 gene
Y144	S. cerevisiae	yes	delete HO
Y145	Y. lipolytica	yes	mislocalized leu2
Y148	Y. lipolytica	yes	Mutation to stop codons in open reading frame (ORF) of the hap4 gene
Y150	Y. lipolytica	no	

Y153	P. pastoris	yes	mislocalized GAP promoter
Y154	Y. lipolytica	yes	deletion of gene YALI0A19844
Y155	Y. lipolytica	yes	ura3 distruption
Y159	S. cerevisiae	yes	m13_origin
Y162	S. cerevisiae	yes	delete HO
Y163	Y. lipolytica	no	
Y164	Y. lipolytica	no	
Y165	S. cerevisiae	yes	delete HO
Y168	P. pastoris	no	
Y170	S. cerevisiae	no	
Y172	S. cerevisiae	yes	m13_origin
Y174	Y. lipolytica	yes	Missense point mutations in protein coding genes
Y175	Y. lipolytica	yes	ura3 distruption
Y176	S. cerevisiae	yes	3bp change on gene Cdc48 at location 105 bases into the gene so that it codes and R instead of a G
Y178	K. pastoris	no	
Y180	P. pastoris	no	
Y187	Y. lipolytica	yes	Mutation to stop codons in open reading frame (ORF) of the hap4 gene
Y188	S. cerevisiae	yes	point mutation in HIS3
Y189	Y. lipolytica	yes	mislocalized leu2
Y190	S. cerevisiae	yes	m13_origin
Y191	Y. lipolytica	yes	v5 epitope
Y195	K. pastoris	no	
Y197	Y. lipolytica	yes	deletion of gene YALI0A19844
Y198	S. cerevisiae	yes	point mutation in HIS3
Y199	P. pastoris	yes	mislocalized GAP promoter

Table B-I. Sample Key. Summary of the full information provided by T&E after the submission of classification results.

Curriculum Vitae **Yuchen Ge** **May, 2020**

2700 Remington Avenue, Apartment 431
Baltimore, Maryland 21211 USA
(+1) 858.281.3601
yge15@jhu.edu

EDUCATION AND DEGREES

2018–Present Master of Science in Engineering
Department of Biomedical Engineering, Johns Hopkins University

2014–2018 Bachelor of Science
Division of Biological Sciences, University of California San Diego

RESEARCH EXPERIENCE

Johns Hopkins University, Whiting School of Engineering
Department of Biomedical Engineering and Institute of Computational Medicine (Jun 2019-Present)
Master's thesis completed under supervision of Professor Joel Bader: Detecting Genetic Engineering with a Knowledge-rich DNA Sequence Classifier

- Building a classifier that identifies bioengineered sequences in yeast genome
- Identified native host using Kraken2/Bracken and collected yeast reference genomes for Saccharomyces cerevisiae, Yarrowia lipolytica, Pichia pastoris
- Extracted unmapped reads against the references using Bowtie2 and de novo assembled the suspicious reads for downstream contigs annotation
- Classified each suspicious contig as engineered vs non-engineered against NCBI non-redundant nucleotide/protein databases, to detect foreign gene content, structural variants, missense mutation, and synonymous recoding
- Tested the automated pipeline on T&E Batch 1/2. Of the 50 blinded samples, 35 were known positives (28 correct) and 15 were known negatives (all correct), with overall accuracy 0.86, sensitivity 1, and specificity 0.8

Johns Hopkins University, Whiting School of Engineering
Department of Biomedical Engineering and Institute of Computational Medicine (Sep 2018-Aug 2019)
Course project, Precision Care Medicine I (EN.580.680) & II (EN.580.681), Professor Raimond Winslow: A Pulse Arrival Time Based Method to Establish Blood Pressure Limits of Autoregulation and Optimal Blood Pressure in Individual Patients During Surgery

- Proposed three Pulse Arrive Time (PAT)-based methods, including static variance-based method (SVBM), time-dependent variance-based method (TVBM), and time-dependent correlation-based method (TCBM), to estimate autoregulation limits and optimal blood pressure, using real-time continuous electrocardiogram (ECG) and arterial blood pressure (ABP) waveform data from 111 patients
- Performed PAT and mean arterial pressure (MAP) calculation with Matlab, and developed Graphical User Interface (GUI) and Signal Quality Index (SQI) to inspect the quality of detection manually and automatically
- Via comparison with cerebral oximetry (Cb-Oxi) method, demonstrated that a TCBM performed the best of the three proposed methods, which could be implemented without any additional instrumentation as long as an arterial catheter was in place and could achieve results similar to those obtained via Cb-Oxi
- Validated the bias between Cb-Oxi and TCBM is clinically insignificant (<5 mmHg), and planning to introduce more samples that have established autoregulation limits for improved parameter learning and method refinement

Johns Hopkins University, Whiting School of Engineering
Department of Biomedical Engineering and Institute of Computational Medicine (Apr 2019-May 2019)
Course project, Foundations of Computational Biology & Bioinformatics II (EN.580.688), Professor Rachel Karchin: Network-based Approach to Prioritize Breast Cancer Driver Genes

- Selected a panel of genes from the Cancer Genome Atlas (TCGA) database whose expression differs significantly in stage iv breast cancer from the other stages
- Prioritized the genes as candidate drivers using a network of known interactions
- Validated the candidate drivers by comparing them to confirmed breast cancer drivers, by performing gene ontology analysis, and by evaluating their influence on survival time
- Concluded that the network approach yielded 111 potential driver genes, among which TFPI2 and FABP7 were determined to have significant influence on patient survival; moreover, PAX5 overlapped with the published 46 drivers

Johns Hopkins University, Whiting School of Engineering
Department of Biomedical Engineering and Institute of Computational Medicine (Nov 2018-Dec 2018)
Course project, Introduction to Computational Medicine I (EN.580.631), Professor Raimond Winslow: Classification of Sepsis Based on Predictive Models from Real-time ICU Data Feeds

- Used generalized linear model (both static and dynamic) to predict early onset of sepsis
- Implemented the two models: Static (physiological parameters that do not vary by time) & Dynamic (physiological parameters that vary by time) by steps of feature selection, 10-fold cross-validation, and test
- Plotted the ROC curve for the test dataset, which was useful because the maximum accuracy and AUC indicated how well the best model from the training set performed to separate sepsis from non-sepsis patients on test set, using the combination of features selected
- Concluded that the selected threshold from training is optimal under the given best-performing model, with accuracy of 62.5% and AUC of 0.6 on the test set

TECHNIQUES AND SKILLS

Languages

- English: Native or bilingual proficiency
- Chinese: Native or bilingual proficiency

Computer Literacy

- Expert in Microsoft Office Suite, with a focus on Excel
- Programming languages with proficiency: Bash (2yrs), Python (2yrs), R (2yrs), Matlab (2yrs), Java (2yrs), C++/C (2yrs)

Bioinformatics Skills

- Metagenomics, Genome Assembly, DNA/RNA-seq, Differential gene expression, Gene Ontology, Genome-wide association study (GWAS), Proteomics, Network Modeling

Molecular Techniques

- DNA/RNA/Protein extraction and purification, PCR, RT-PCR, Quantitative PCR, Southern Blot, Western Blot, DNA/RNA(cDNA) Sequence Analysis, Agarose Gel Electrophoresis/Imaging, Recombinant DNA, Transcriptional gene regulation

TEACHING EXPERIENCE

University of California San Diego, Division of Biological Sciences (Mar 2018-Jun 2018)
Teaching Assistant

- Undergraduate Student Instructor for BIMM 100: Molecular Biology, taught by Dr. Keefe Reuther

University of California San Diego, Division of Biological Sciences *(Mar 2016-Jun 2016)*
Teaching Assistant

- Undergraduate Student Instructor for BILD 1: The Cell, taught by Dr. Keefe Reuther