ACS Synthetic Biology

pubs.acs.org/synthbio Research Article

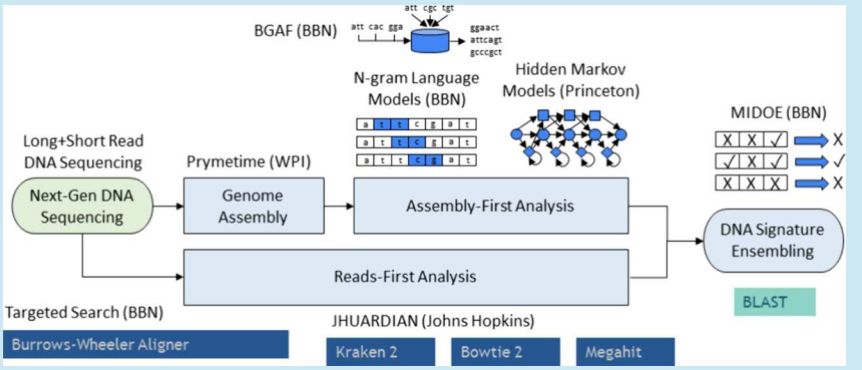
Ensemble Detection of DNA Engineering Signatures

Published as part of ACS Synthetic Biology virtual special issue "IWBDA 2022".

Aaron Adler, Joel S. Bader, Brian Basnight, Benjamin W. Booth, Jitong Cai, Elizabeth Cho, Joseph H. Collins, Yuchen Ge, John Grothendieck, Kevin Keating, Tyler Marshall, Anton Persikov, Helen Scott, Roy Siegelmann, Mona Singh, Allison Taggart, Benjamin Toll, Kenneth H. Wan, Daniel Wyschogrod, Fusun Yaman, Eric M. Young, Susan E. Celniker, and Nicholas Roehner*

Cite This: *ACS Synth. Biol.* 2024, 13, 1105−1115 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: Synthetic biology is creating genetically engineered organisms at an increasing rate for many potentially valuable applications, but this potential comes with the risk of misuse or accidental release. To begin to address this issue, we have developed a system called GUARDIAN that can automatically detect signatures of engineering in DNA sequencing data, and we have conducted a blinded test of this system using a curated Test and Evaluation (T&E) data set. GUARDIAN uses an ensemble approach based on the guiding principle that no single approach is likely to be able to detect engineering with perfect accuracy. Critically, ensembling enables GUARDIAN to detect sequence inserts in 13 target organisms with a high degree of specificity that requires no subject matter expert (SME) review.

KEYWORDS: *artificial intelligence, bioinformatics, biosecurity, engineering detection, machine learning*

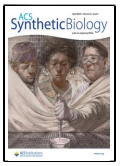
INTRODUCTION

Synthetic biology is creating purposefully engineered organisms at an increasing rate and with increasing complexity. There are many valuable applications of synthetic biology, including treating disease[1] and enhancing production of crops[2] and chemicals.[3] There is also, however, an increasing risk of misuse or accidental release of genetically engineered organisms. The goal of the IARPA FELIX (Finding Engineering-Linked Indicators) program was to determine whether genetic engineering could be accurately detected, with a focus on detecting signatures of engineering in DNA sequencing data (https://www.iarpa.gov/research-programs/felix). Once engineered DNA is detected, a future step would be to understand the function of the engineering or attribute it to its lab-of-origin, which would be a key step toward deterring the malicious application of synthetic biology and genetic engineering technologies.[4]

Prior to the FELIX program, there did not exist any curated data sets or dedicated tools for detecting genetic engineering. A handful of tools had been developed for engineering attribution, but these tools effectively assumed that a sequence's classification as engineered or natural was known beforehand.[5,6] More recently, a genetic engineering attribution competition has been held and has yielded improved results over these first tools,[7] but none of these approaches yet address the question of whether a sequence has been engineered or not.

In the domain of genome editing, work has been done to develop tools to identify edits and off-target sites for CRISPR

Received: June 30, 2023
Revised: February 28, 2024
Accepted: February 28, 2024
Published: March 12, 2024



https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

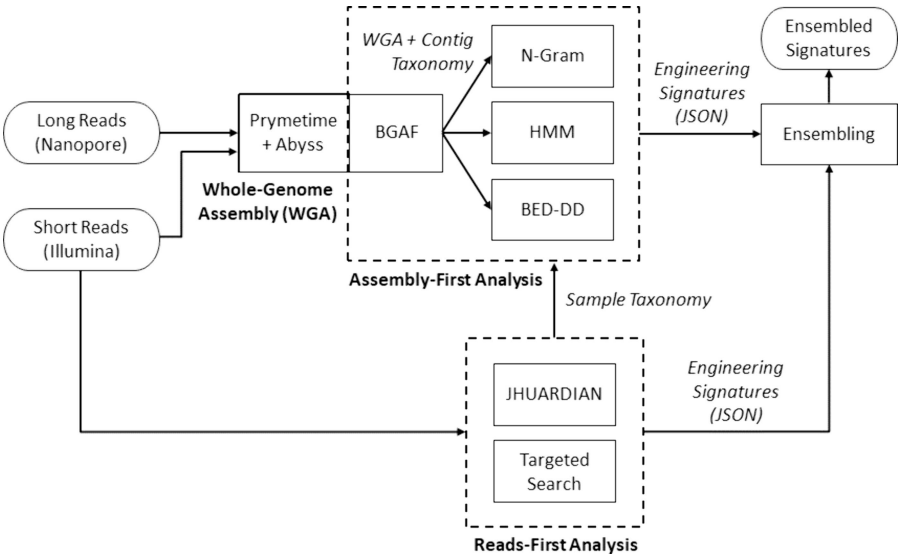


Figure 1. Configuration of GUARDIAN modules for analyzing FELIX T&E samples. Pill box-shaped nodes represent input/output data, while rectangular nodes represent Dockerized software modules. Arrows represent the flow of data to/from modules. Dashed boxes group together like modules. The modules for Prymetime,[10] Abyss,[11] and BGAF overlap to signify their tight integration as part of a single Docker container.

editing experiments,[8,9] which could potentially be adapted to detect CRISPR edits in samples of unknown provenance. While these approaches would initially be limited to the detection of small edits made using CRISPR-based technologies, they would address a type of engineering signature that proved to be difficult to detect by most systems as part of the FELIX program (small edits as opposed to larger sequence inserts and deletions).

As part of the Guard for Uncovering Accidental Release, Detecting Intentional Alterations, and Nefariousness (GUARDIAN) project under the IARPA FELIX program, we have developed software modules and connected them into a system that use a variety of techniques from bioinformatics, artificial intelligence (AI), and machine learning to screen DNA sequencing data for signatures of engineering. As part of FELIX, we have conducted a blinded test of GUARDIAN against samples provided by a Test and Evaluation (T&E) team. In doing so, we have demonstrated that ensembling over standardized evidence of engineering can be an effective approach to automate the detection of sequence inserts in large numbers of samples with unknown provenance. We will show how ensembling can greatly increase specificity (true negative rate) of detection with a minimal decrease in detection sensitivity and without requiring subject matter expert (SME) review. In addition, we will discuss lessons learned from participation in the FELIX program and future directions for detection of engineered organisms and beyond.

RESULTS AND DISCUSSION

The design of our GUARDIAN whole-genome sequencing analysis system is based on the guiding principle that no single approach is likely to be able to detect every signature of engineering for every potential use case with perfect accuracy. Consequently, rather than architect GUARDIAN as a monolithic system, we have loosely connected its modules by standardizing their inputs and outputs in terms of a common data model, sequence file formats (FASTQ, FASTA), and a domain-specific JSON schema for evidence of engineering. Critically, this enables us to rapidly reconfigure GUARDIAN's modules for different use cases while still allowing us to ensemble their detected signatures of engineering.

Figure 1 shows the configuration of GUARDIAN's modules that we applied to the final FELIX T&E test. In this configuration, we send short-read sequencing data for a sample to our two reads-first analysis modules, Targeted Search and JHUARDIAN. These modules work in parallel to produce signatures of engineering and information on sample taxonomy, the latter of which can be used by other modules to optimize their choice of model to use for analysis. In parallel, we send both short-read and long-read sequencing data to the BBN Genetic Anomaly Filter (BGAF), which assembles them into genomes using our custom hybrid assembly pipeline Prymetime[10] and the Abyss[11] assembly tool. BGAF then taxonomically classifies each assembled sequence contig and shares this information with the other assembly first modules N-Gram, Hidden Markov Models (HMM), and BED-DD, at which point they all work in parallel to produce signatures of engineering. These are primarily putative sequence inserts, although BED-DD focuses solely on detecting sequence deletions.

Finally, we ensemble all modules' signatures via a series of pairwise sequence alignments and throw out any that do not match at least one signature detected by another module. Because each module uses different types of models, heuristics, and in some cases training data, they are less likely to make the same false positive calls. Thus, we can expect ensembling in this manner to increase the overall specificity (true negative rate) of our system, ideally without negatively impacting its sensitivity too much. A decrease in sensitivity due to ensembling can occur when one or more modules exhibit truly unique detection capabilities (i.e., when they are capable of detecting signatures that no other module can).

Table 1 counts all 100 samples tested during the final FELIX T&E by their organism(s) and whether or not they have been engineered via a sequence insert (see the Excel file in the Supporting Information for additional sample metadata including BioSample Accession Numbers). These samples were estimated to have 20−40X sequencing coverage. Their organisms include a diverse set of 16 species of bacteria, fungi,

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

Table 1. FELIX T&E Sample Organisms and Number With/Without Inserts

organism	# samples with insert	# samples without insert
Arabadopsis thaliana	1	1
Bacillus subtilis	7	3
Citrobacter freundii	2	1
Escherichia coli	11	3
Influenza A	1	2
Oceanobacillus oncorhynchi	0	1
Oryza sativa	3	1
P1 phage	0	1
Pseudomonas aeruginosa	2	4
Pseudomonas putida	3	1
Rabies lyssavirus	0	1
Rhodosporidium toruloides	2	1
Saccharomyces cerevisiae	10	4
Salmonella enterica	8	1
T7 phage	0	1
Yarrowia lipolytica	2	1
bacteria mixture	0	3
bacteria + yeast mixture	0	6
metagenomic soil	4	2
metagenomic gut	4	2
total	60	40

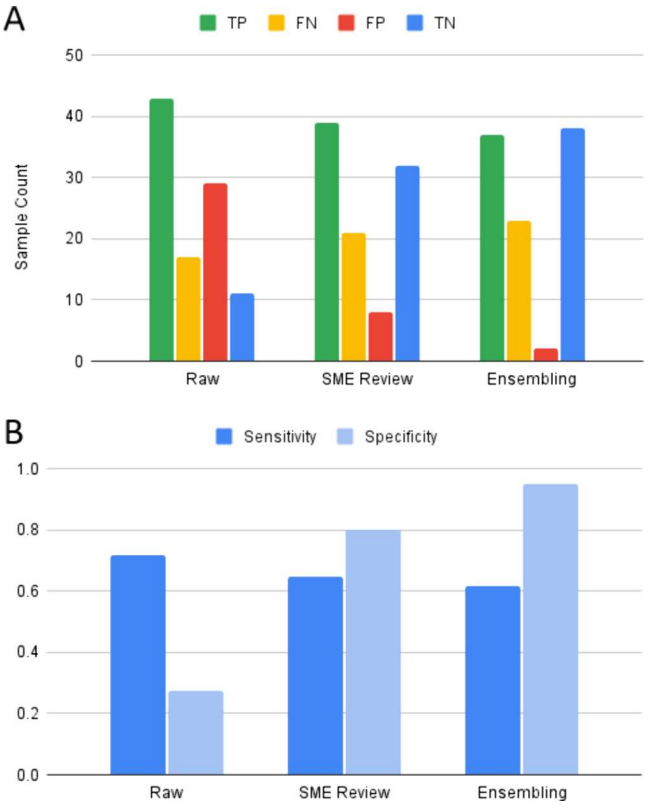


Figure 2. FELIX T&E results for naïve analysis of the raw results from GUARDIAN's modules versus applying SME review or automated ensembling. These include (A) true positive, false negative, false positive, and true negative (TP, FN, FP, and TN) sample counts for these three approaches and (B) sensitivities and specificities (true positive and true negative rates) calculated from these sample counts.

plants, and viruses. While the T&E samples did include types of engineering signatures besides inserts such as small edits and deletions, our analysis will focus on detection of inserts since GUARDIAN as a whole was able to distinguish this signature type from natural variation with the greatest consistency (see the CSV file in the Supporting Information for a complete listing of T&E engineering signatures by sample, and see the FASTA file in the Supporting Information for each engineering signature's DNA sequence). Later on, we will discuss future approaches to improve detection of subtler signatures of engineering such as small edits.

Based on our results for the 100 T&E samples in Table 1, we calculate that GUARDIAN's ensembled detection of samples with sequence inserts has a sensitivity of 0.62 and a specificity of 0.95, compared to a slightly higher sensitivity of 0.65 but a markedly worse specificity of 0.8 when using blinded SME review to identify and discard potential false positive insert detections (see Figure 2). Both SME review and ensembling improve specificity significantly over a naïve approach of calling a sample engineered if any one of GUARDIAN's modules detect engineering in its raw results (0.8 and 0.95 versus 0.28, respectively). In terms of processing time, however, SME review took 5 days to complete, whereas ensembling took less than 2 h. Thus, compared to SME review, ensembling greatly lowers the time and expertise required to achieve engineering detection with few false positives (high specificity).

With ensembling, GUARDIAN only makes two false positive calls for the FELIX T&E samples: one for a sample of the *E. coli* strain Evo1, wild-type (SAMN29939563) and one for a strain of P1 phage that carries a deletion, P1_Δ(cra-darB), (SAMN38524478). The precise cause of these false positives is unknown, in part because these samples were the only examples of their respective strains, and because there was only one sample of P1 phage in the entire T&E batch. The most likely explanation, however, is that these particular strains are not well represented in the nonengineered training data for GUARDIAN, and they also include sequences that are commonly used in artificial cloning vectors and may be mistaken for engineering.

As for the 23 false negative calls that GUARDIAN makes with ensembly, we first examined whether the taxonomy of the host organism appears to have any effect on these missed detections. As shown in Figure 3, while samples of some organisms such as *E. coli*, *P. putida*, soil (containing engineered *P. aeruginosa*), and Influenza A are fractionally overrepresented among false negative samples when compared to their fractions among positively engineered samples, this observation is also consistent with their overrepresented fractions among samples engineered with sequence inserts having total length less than 1000 bp. In other words, missed detections in samples of these organisms seem more likely to have been caused by the character of these organisms' engineering (i.e., having a larger fraction of their samples engineered with short inserts compared to other organisms) than their taxonomy. We will see this again shortly when considering the limits of detection of this configuration of the GUARDIAN.

Ultimately, rather than host organism or another factor, it appears that total insert length and the proportion of engineered cells have the greatest effect on the performance of GUARDIAN's ensembled detection capability. If we exclude samples based on GUARDIAN's apparent limits of detection in Figure 4 (total insert length greater than 1000 base pairs, engineered cell fraction greater than 5.5×10^{-06}), then 19 out of 23 false negatives are removed and sensitivity rises from 0.62 to 0.9 with no loss in specificity. This leaves only four false negative samples:

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

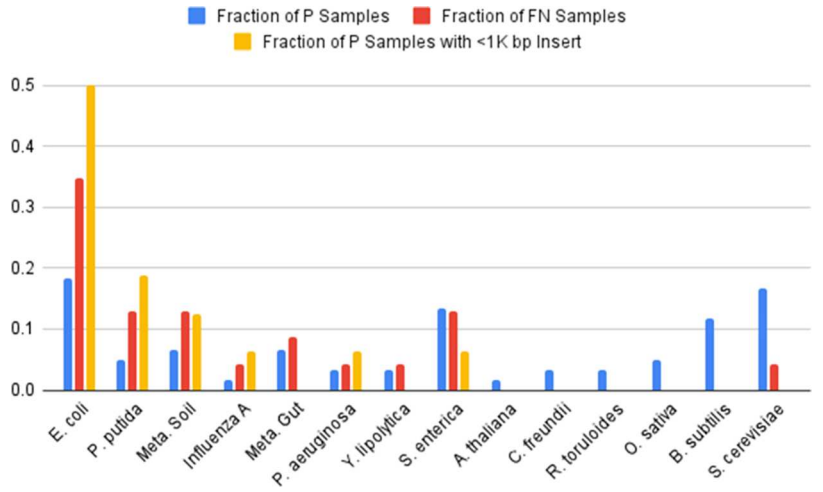


Figure 3. Analyzing the potential effect of host organism taxonomy on GUARDIAN's false negative calls with ensembling. Shown here are fractions of samples of different host organisms among samples that are positively (P) engineered, false negatives (FN) missed by GUARDIAN, and positively engineered with sequence inserts having total length less than 1000 bp. The label "Meta. Soil" refers to metagenomic soil samples containing engineered *P. aeruginosa* (SAMN38676624-7), while "Meta. Gut" refers to metagenomic gut samples evenly divided between those containing engineered *S. enterica* (mouse cecal, SAMN37954766-7) and engineered *E. coli* (cow rumen, SAMN38676619-20).

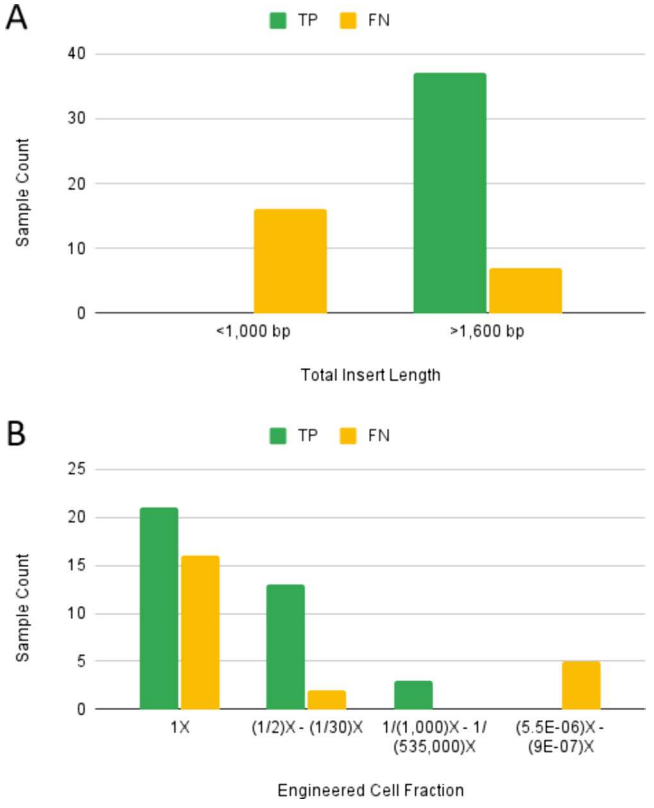


Figure 4. FELIX T&E true positive (TP) and false negative (FN) sample counts by (A) total insert length and (B) engineered cell fraction.

two of *S. enterica* strain JE4199-cured (SAMN38524180 and SAMN38524181, spiked with a compound insertion of 11,184 bp at 1/20 and 1/30, respectively), one of *S. cerevisiae* strain LG811-9A (SAMN37931218, carrying a new TY1 transposon engineered to integrate at the URA3 gene on chromosome V), and one of *Y. lipolytica* strain pex17-1 (SAMN37931223, carrying an insertion of a SacC gene disrupting URA3). In the case of the *S. enterica* samples, these may have been missed since they had the lowest engineered cell fractions (1/20 and 1/30) in a group of samples of the same strain but different dilution factors.

Table 2 shows the effect of excluding samples based on a low total insert length or engineered cell fraction alone. Excluding samples with total insert length of 1000 bp or below removes 16 out of 23 false negatives and causes sensitivity to rise from 0.62 to 0.84. For reference, the average gene length in prokaryotes such as bacteria is close to 1000 base pairs and is over 1300 base pairs in eukaryotes such as yeast and plants. Excluding samples with engineered cell fraction below 5.5×10^{-06}, on the other hand, removes 5 out of 23 false negatives and only causes sensitivity to rise from 0.62 to 0.67. Note that, while GUARDIAN's apparent limits of detection are inserts greater than 1000 bp in length and engineered cell fractions greater than 5.5×10^{-06}, these limits are for ensembled detection and do not necessarily extend to GUARDIAN's individual submodules, which in some cases may have lower limits. Without significant tuning, however, the results of individual modules that go below these limits of detection likely require SME review, which may not be feasible for use cases involving large numbers of samples.

Table 2. GUARDIAN Sensitivity and Specificity Outside/Within Limits of Detection

parameter	samples of all total insert lengths	samples with total insert length length >1000 bp
samples of all engineered cell fractions	sensitivity = 0.62 specificity = 0.95	sensitivity = 0.84 specificity = 0.95
samples with engineered cell fraction >5.5E-06	sensitivity = 0.67 specificity = 0.95	sensitivity = 0.9 specificity = 0.95

We also examined the performance of GUARDIAN's individual modules to determine their contribution to the overall T&E results. Specifically, we looked at these modules' informedness (sensitivity + specificity − 1) for detection of insert samples with and without SME review and ensembling (see Figure 5). Without any SME review or ensembling, most modules have an informedness less than or equal to 0.1, with two notable exceptions being HMM (0.3) and N-Gram (0.18). This

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

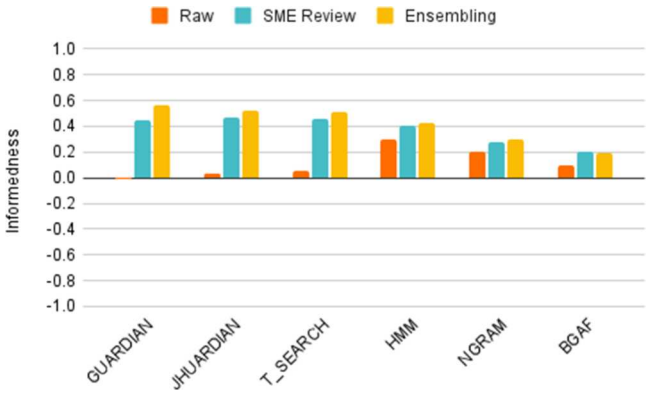


Figure 5. Informedness (sensitivity + specificity − 1) of GUARDIAN and its submodules during the FELIX T&E. Informedness is shown for three different approaches to processing calling samples engineered based on module signatures: calling samples engineered when any signature is present in the raw results, calling based on SME review of signatures, and calling based on automated ensembling of signatures.

is likely due in part to HMM and N-Gram's implementation of better controls to avoid false positive calls. These controls include HMM's exclusion list of UniVec sequences aligning to the reference genomes for our target organisms and N-Gram's BLAST-based check of whether potential engineered sequences align to these reference genomes with high coverage.

Interestingly, two of GUARDIAN's modules with the lowest informedness prior to SME review or ensembling (JHUARDIAN and Targeted Search) have the greatest informedness afterward (0.53 and 0.51, respectively). The initial low informedness of JHUARDIAN and Targeted Search is due to their low specificity (0.4 and 0.51, respectively, see Figure 6),

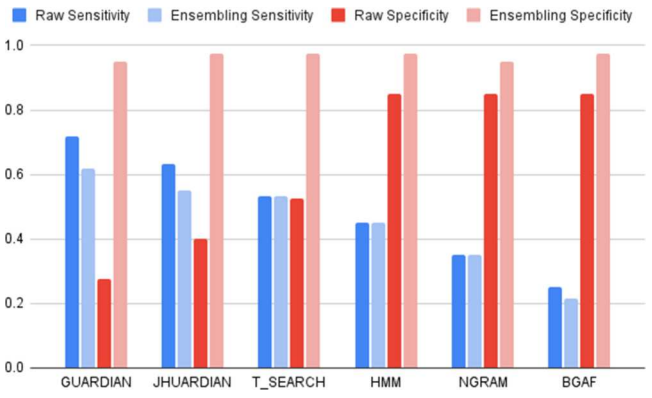


Figure 6. FELIX T&E sensitivity and specificity for naïve analysis of the raw results from GUARDIAN's modules versus automated ensembling of their engineering signatures.

which is partly caused by their having less stringent controls implemented relative to other modules. Using ensembling, however, we can increase their specificity with little or no decrease in their initially high sensitivity. This high sensitivity relative to other modules is partly because JHUARDIAN and Targeted Search, as reads-first modules, were the only ones able to analyze all eight engineered metagenomic samples. Many of these samples could not be assembled in a tractable amount of time during the T&E using Prymetime[10] or Abyss,[11] although, in the future, this could be optimized (for example, by adding additional taxonomic classification steps of raw reads prior to assembly). In addition, there are some assembly first modules (primarily BGAF), which could be reconfigured to analyze raw reads as needed.

While every module increased in specificity and most did not decrease in sensitivity following ensembling, there were two (JHUARDIAN and BGAF) that had small decreases in sensitivity. This is because these modules were unique in being the only modules to detect sequence inserts in certain samples, which may be due to the fact that both of these modules use BLAST[12] to match suspicious sequences against NCBI and not just UniVec, thereby increasing the likelihood of their detecting some genomic inserts that would be missed by approaches focused on plasmid vector sequences.

Finally, we note that ensembling yields roughly equal or better performance than SME review for every one of GUARDIAN's modules. Only two of GUARDIAN's modules (JHUARDIAN and BGAF) experience any loss in sensitivity due to ensembling, and this is because they are able to detect some signatures found by no other module. In addition, the informedness of GUARDIAN's ensembled approach as a whole is greater than its best performing module (0.57 versus 0.53 for JHUARDIAN), and the latter is not possible without ensembling or a time-consuming SME review.

To summarize, we have designed GUARDIAN to be a collection of software modules loosely connected using a common data model. This allows us to not only rapidly reconfigure GUARDIAN's modules for application to different use cases, but it also enables us to readily ensemble their output signatures of engineering, even as new modules are added to the system. Most significantly, ensembling gives us the ability to quickly analyze evidence of DNA engineering while requiring less expertise and maintaining a high degree of specificity. This makes ensembling well-suited as a heuristic for implementing systems to detect engineering in large numbers of samples of potentially unknown provenance at a high rate of throughput. Another benefit of ensembling that we have demonstrated during the FELIX T&E is that not all modules need to be especially well-tuned. Some modules can be overly sensitive and others can be overly specific, but ensembling will help to produce a system that has performance greater than any of its individual parts.

Future Directions. Next, we discuss five future directions that are key to facilitating the deployment of GUARDIAN and similar systems for the detection of genetic engineering and related applications.

Develop Standards and Associated Tooling for Modeling Evidence of Genetic Engineering. Data standards are a critically important but often overlooked component of AI and machine learning. For GUARDIAN, our development of a data model and associated JSON schema to represent evidence of DNA engineering was absolutely necessary to permit ensembling over the output of five different modules and enable highly specific detection of sequence inserts with no SME review. In addition, while our ensembling approach during FELIX T&E primarily focused on pairwise alignment of sequences, our data model encodes other useful metadata that could be used to compare engineering signatures, such as assembly coordinates and host/insert taxonomy. Going forward, standards such as these should be disseminated through open source projects such as the Synthetic Biology Open Language (SBOL)[13] to facilitate collaboration between the greater AI, synthetic biology, and biosecurity communities and help establish networks of interoperable detectors at different organizations.

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

Curate Data Sources for Natural Sequences. A very common failure mode among all FELIX performers was the presence of engineered sequences that were not labeled as such in training data obtained from public databases like NCBI. Errors like these reduce the sensitivity of anomaly-based approaches to engineering detection by causing them to filter out engineered sequences as though they were native to the host. Addressing this issue required weeks of SME effort to curate genomes for new target organisms and build exclusion lists for genomes that contained engineering. In addition, recent studies have highlighted the pitfalls of using general purpose sequence repositories such as NCBI to train screening systems.[14] Consequently, new databases and methods are needed to aggregate and curate natural sequence data for the purposes of biosecurity. In addition, new approaches to screening are needed that are robust to mislabeled training data (whether unintentional or deliberate).

Develop and Aggregate Data Sources for Engineered Sequences. While the argument can be made that additional natural sequence data are needed to enhance anomaly-based approaches to engineering detection, the space of unknown natural DNA is very large, changes continuously, and will be difficult, if not impossible, to characterize fully. Engineered sequences, on the other hand, are far fewer in number and do not change as frequently but have even less availability via publicly accessible data sources. Based on our experiences during FELIX, signature-based approaches trained on engineered sequences were essential for handling real-world samples that contained many different species and novel natural sequences that could be mistaken for engineering by anomaly-based approaches alone.

Enhance Methods for Detecting Subtle Signatures of Engineering. More examples of engineering will also likely be crucial for detecting signatures of engineering that are subtler than sequence inserts such as small sequence edits and deletions. Because these signatures can be difficult to distinguish from natural sequence variation, it becomes necessary to consider their greater sequence context in terms of whether they occur in features known to work together or that have been previously engineered or whether they occur near sequence features required for making sequence modifications (such as CRISPR PAM sites). To obtain greater sequence context, it will not only be important to improve pipelines for sequence annotation and their training data, but it will also be vital to develop methods for generating more contiguous genome assemblies (such those produced by our Prymetime assembly pipeline[10]).

Enhance Methods for Analysis of Complex Metagenomic Samples. Real-world samples are frequently metagenomic and contain DNA sequences from many different species that can be difficult to assemble into contigs. Consequently, we found it valuable to have modules such as Targeted Search and JHUARDIAN for directly analyzing sequencing reads or filtering them before assembling a much smaller fraction of the genome for subsequent analysis. To better prepare GUARDIAN and other engineering detection systems for deployment, we recommend adapting approaches that nominally require whole-genome assemblies as input to alternatively analyze filtered sequencing reads and/or partial assemblies instead. Furthermore, given our experiences using JHUARDIAN and Targeted Search during the FELIX T&E, we recommend using metrics such as read depth to avoid false positives during reads-first analysis due to errors in sequencing, assembly, and/or sample preparation.

Decompile Engineered Sequence Function and Attribute Lab-of-Origin. Beyond engineering detection, our work on FELIX could be extended to help the biosecurity community answer more relevant investigative questions about the function or purpose of an engineered sequence and its lab-of-origin. In particular, we can imagine developing a biological decompiler that takes known design motifs linked to specific biological functions or laboratories and maps these motifs to detected signatures of engineering and other predicted sequence features. Such a decompiler would not only be useful for the purposes of biosecurity but could also significantly enhance Design-Build-Test-Learn (DBTL) cycles for synthetic biology by supporting automated quality control checks on built sequences against their original design specification.

METHODS

GUARDIAN. Our heuristic approach to ensembly consists of aligning the DNA sequence signatures detected by its modules and grouping them based on their sequence similarity. GUARDIAN then calls a sample engineered if it has at least one group of signatures that originated from at least two different modules.

To form DNA signature groups, GUARDIAN pairwise aligns a target sequence with those in any available groups until a matching sequence is found, in which case the target sequence is added to the matching sequence's group. If no match is found, then the target sequence becomes the sole member of the new signature group. All sequence alignments are performed using the BioPython[15] PairwiseAligner class and the following parameters: match_score = 1, mismatch_score = 2, and internal_gap_score = −2.5. An alignment between a pair of sequences is accepted if both are >20 bp in length and if the alignment score is greater than max $(0.5L, L\text{-}0.0005L^2)$, where L is the length of the shorter sequence in the pair. This is a sliding scale threshold for which a sequence with a small L (100s of base pairs) must completely overlap with its partner to be considered a match, whereas a sequence with a large L (1000s of base pairs) can overlap by just half of its length.

To produce DNA sequence signatures for ensembling, most of GUARDIAN's modules combine anomaly-based and signature-based strategies for engineering detection. Anomaly-based strategies use models of natural sequences to predict whether a new sequence is unnatural. Signature-based strategies, on the other hand, use models of engineered sequences to predict whether a new sequence is engineered. Next, we will briefly discuss how each of GUARDIAN's modules implements these strategies and key distinctions between them, including whether they analyze sequencing data directly or whether they require sequencing reads to be assembled into contigs first. BED-DD is excluded from this discussion, since it was used to detect sequence deletions rather than inserts and was not included in our ensembling approach.

JHUARDIAN. This read analysis module includes taxonomic classification with Kraken[16] and Bracken,[17] read mapping with Bowtie,[18] read assembly with Megahit,[19] and sequence annotation with BLAST. The Kraken and Bracken methods are run with the largest available prebuild Kraken database with a read length of 300 and read counts down to the species level. Bracken results are parsed to identify taxa at the family level that account for at least 1,000,000 reads or 5% of the total reads (these parameters are adjustable and were chosen according to the anticipated data). For anomaly-based detection, the reference genomes for all of the corresponding species are

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

gathered from NCBI RefSeq and used to generate a bowtie2 index. Reads are then mapped using bowtie2 using parameters "--end-to-end", "--very-fast", and "--no-discordant". The unmapped reads are collected and assembled using Megahit with default parameters. Megahit assemblies with at least 10 reads are then annotated using BLAST. The minimum read count of 10 is an adjustable parameter, and for low-coverage sequencing, a read count of 1 provides improved sensitivity at the expense of greater computational time. The BLAST annotations are classified as signatures of engineering if the taxonomic classification does not match the host organism's, or if a suspicious keyword is present in the description ("artificial", "cloning", "mutant", "synthetic", or "vector"). For signature-based detection, JHUARDIAN filters in reads that map to engineering vector sequences in the UniVec database. JHUARDIAN then assembles the reads that survive filtering and uses BLAST against the NCBI database to annotate the resulting contigs, as described above.

Targeted Search. This read analysis module uses the Burrows-Wheeler Aligner (BWA)[20] to implement signature-based detection of engineering. Targeted Search focuses on identifying engineering signatures in 300 bp Illumina reads. First, Targeted Search uses BWA to align a sample of Illumina reads against the reference genomes for our target organisms and to a curated list of sequence features commonly used in genetic engineering (named FUNYES). Then, a series of filtering steps can limit the list of suspicious reads to those for which there are partial, nonoverlapping alignments to both the closest matching reference genome and a sequence feature in FUNYES. These partial alignments suggest that an engineering sequence feature has been inserted into the target organism's DNA. Alternatively, Targeted Search can retain all reads with alignments to a sequence feature in FUNYES, which is the mode that we used in this work (this enables detection of engineering sequence features in foreign plasmid vectors). Targeted Search calls a sample engineered if it contains at least 10 reads that align with a sequence feature in FUNYES (this is an adjustable threshold).

BBN Genetic Anomaly Filter (BGAF). This sequence analysis module is based on techniques from malware detection that were later adapted for pathogenic DNA screening as part of BBN's FAST-NA tool.[14] BGAF includes whole-genome assembly with Abyss[11] and Prymetime[10] (our custom hybrid assembly pipeline) taxonomic classification and read mapping with FAST-NA, and annotation with BLAST. Prymetime uses the assembly tools Flye[21] and Unicycler[22] to combine short-read and long-read sequencing data to produce highly contiguous genome assemblies.

The inputs to the training process for BGAF consist of sequences for the target organism (preferably reference genomes) formatted as FASTA and curated to remove any sequences that contain engineering. To train on a target organism, BGAF extracts all k-mers of length k = 16 from the sequences in the corpus for that target organism and then inserts them into a Bloom filter. The Bloom filter is a very space efficient data structure that has a function similar to that of a "set" in higher level languages, such that k-mers from a new sample can be tested for membership in the Bloom filter.

The analysis steps performed by BGAF on each sample separately are as follows:

1. Assembly: The input to the assembly stage consists of both Illumina short reads and Nanopore long reads. The Illumina reads are assembled into contigs using Abyss, and both the Illumina and Nanopore reads are assembled into contigs using GUARDIAN's Prymetime assembly module. If the setting for the "AssemblerTypes" parameter in the configuration file is for "illumina" and "nanopore", both Abyss and Prymetime will be run. If only "illumina" is set, only Abyss will be run. Some of the contigs produced by Abyss will be subregions of the contigs produced by Prymetime. BGAF uses MUMmer[23] to find those Abyss contigs that are wholly contained within a Prymetime contig and removes them from the final assembly.
2. Taxonomic classification: BGAF classifies the contigs in the final assembly by comparing them to the Bloom filters for each target organism. Specifically, BGAF breaks each contig into k-mers and checks them against the Bloom filters for each target organism one at a time. If there is one target organism that all k-mers in a contig match, then that contig is classified as that organism and no further checks are performed on its k-mers. If there is no such organism, BGAF then determines if the majority of the contig's k-mers match a single target organism, in which case the contig is as classified as that organism. If there is no such majority, then the contig is classified as "unknown" and is not subject to additional analysis by BGAF.
3. Region of interest (ROI) extraction: At this point, each contig has a taxonomic classification. BGAF then checks the k-mers from each contig classified as a particular organism against that organism's Bloom filter and collects the k-mers not found in the filter along with their contig locations. Using this location information, BGAF assembles the filtered, overlapping k-mers into regions of interest (ROIs). BGAF then drops any ROIs less than 175 bp in length and submits the remaining ROIs to BLAST analysis.
4. BLAST analysis and scoring: BGAF uses BLAST to align each ROI against a selected NCBI mirror and our FUNYES list of common engineering sequence features. BGAF then retains and calls "engineered" any ROI that (a) matches an NCBI record with a suspicious keyword in its English language description such as "Vector", "Synthetic", or "Complementation Plasmid", (b) matches a sequence feature in our FUNYES list, or (c) is greater than 300 bp in length. In this work, however, we did not retain long ROIs without a BLAST match.

HMM. This genome analysis module uses Hidden Markov Models (HMMs) constructed from the aligned reference genomes of our target organisms to compute HMM scores for sample assemblies and BLAST ROIs with high scores. The training of an HMM for a new target organism involves several key steps:

1. Sequence alignment: First, we align chromosomal sequences from the reference genomes for the target organism undergo using the progressiveMauve[24] multiple sequence alignment tool with default parameters.
2. HMM model construction: Next, we construct an HMM model for each chromosome by computing transition and emission probabilities for every position, utilizing a preselected reference genome. In these models, each column of symbols in the alignment is represented by a frequency distribution of four nucleotides, including insertions and deletions as states. The model records

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

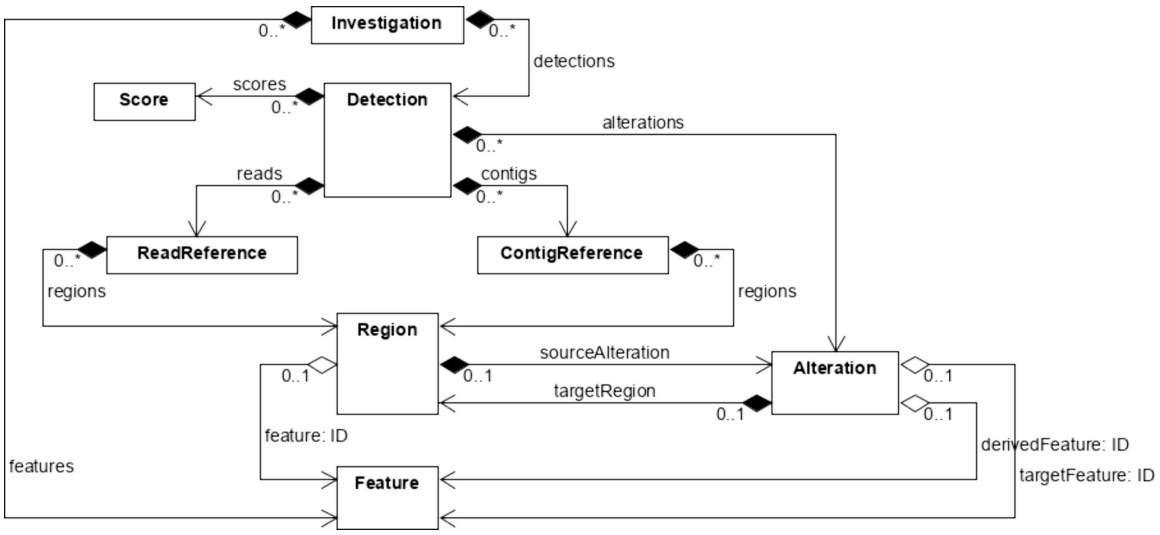


Figure 7. Diagram summarizing key classes and relationships in the GUARDIAN data model. Each box represents a different class of evidence, and each arrow represents a relationship between classes. The beginning of each arrow is a closed or an open diamond. A closed diamond indicates that a relationship is a strict "contains" relationship in which the object of the relationship does not have that relationship with any class instance other than the subject of the relationship (e.g., each instance of the Region class is contained by a single instance of the ReadReference class, ContigReference class, or Alteration class). An open diamond indicates that a relationship is a loose aggregation in which the object of the relationship is referenced by its unique ID and can be the object of other relationships as well (e.g., an instance of the Feature class can be the derived feature of one instance of the Alteration class and the target feature of another instance of the Alteration class).

both emission probabilities and the probabilities of transitioning from one state to another.

3. Vector exclusion list: To obtain organism-specific "vector exclusion lists", we BLAST sequences from the UniVec database against the reference genomes for our target organisms and add any matches to said lists.

Following training, analysis with the HMM module consists of the following steps:

1. Sequence alignment: We first align sample sequences against the reference genome for a target organism using the progressiveMauve multiple sequence alignment tool.
2. HMM scoring: Following alignment, we use the HMM model trained for the target organism to score all sample sequences. This process uses the sample sequence alignment as input to the HMM model, resulting in the multiplication of emission and transition probabilities and the calculations of a probability (HMM score) for each nucleotide position or insertion/deletion within the sample sequences. We do not penalize "deletion to deletion" transitions in the HMM models to account for lower sequencing coverage in some samples.
3. HMM peak scoring: Next, the HMM module detects peaks or regions with significant sequence deviations from the pretrained profile above an organism-specific threshold (which we determined via testing with data outside of the T&E data set analyzed in this work). Then, the HMM module computes an overall HMM peak score by integrating the area under all peaks.
4. BLAST analysis and final decision: Finally, the HMM module BLASTs the HMM peaks against the UniVec database. As previously described, matches containing sequences from the vector exclusion lists are filtered out. A sample sequence is then called "engineered" if its overall HMM peak score exceeds an organism-specific threshold, or if its number of significant BLAST matches against UniVec exceeds zero (Bit score >50, E-value >1e10, and vector coverage >65%).

N-Gram. This sequence analysis module uses n-gram language models constructed from the reference genomes of our target organisms to compute sequence entropy scores for sample assemblies and BLAST sequences with high scores. In this work, N-Gram used lists of reference genome FASTA files as inputs to train classic n-gram language models (LMs) with n = 14 or n = 13 using the SRI language modeling (SRILM) toolkit[25] or a custom script.

For analysis, N-Gram uses a list of LMs and FASTA files as input, with the latter preferably containing sequence contigs assembled from a sample. Raw reads can be used as input to N-Gram, but this may not be practical, except in cases when the number of reads is small enough to preclude genome assembly. For efficiency, input sequences should ideally be taxonomically classified using another module such as JHUARDIAN or BGAF, although N-Gram can optionally classify input sequences based on the minimum sequence entropy (that is, which organism's LM computes the least entropy for the input sequences). N-Gram performed the following analysis steps:

1. Entropy calculations: Given list FASTA files and candidate LMs for analysis, N-Gram generates base pair entropy scores for each combination of a file and LM.
2. ROI extraction: Given sequences and their base pair entropy scores, N-Gram extracts ROIs containing a high proportion of base pairs with entropy above an organism-specific threshold (also determined via testing with data outside of the T&E data set analyzed in this work).
3. ROI postprocessing: Finally, N-Gram can invoke various postprocessing steps for additional information relevant to engineering detection. For example, N-Gram can BLAST ROIs against UniVec, our FUNYES list of common engineering sequence features, and reference genomes for our target organisms, then filter out ROIs based on their matches (or lack thereof) to these

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

resources. In addition, N-Gram can use an LM trained on UniVec and FUNYES to compute entropies based on how different samples are from known engineering sequences, then combine these entropies with those based on differences from natural sequences to generate a likelihood ratio score that captures whether a sample appears engineered in addition to being unnatural. In this work, we filtered out N-Gram ROIs with high coverage BLAST matches to the reference genomes for our target organisms, and we called the remaining ROIs engineered if they had a BLAST match to UniVec or FUNYES.

GUARDIAN Data Model. Our data model represents and aggregates evidence of engineering in terms of two primary classes: Detections and Features (see Figure 7). Features represent identified sequence features of a potentially known function (e.g., pBAD promoter). By contrast, Detections can provide metadata on the location within a read or assembly contig where engineering has been detected and metadata on the supposed sequence alteration (inset, deletion, etc.) represented by the engineering. The engineering in question could be a known feature, but this is not a strict requirement (a sequence insert can be detected without knowing that it is the pBAD promoter).

We have implemented this data model as a JSON schema and made it available with documentation on GitHub as part of the Minimum Information for Detection of Engineering (MIDOE) repository (see the Software and Data Availability). This schema enables us to validate that engineering signatures detected by different modules are comparable and gives us the ability to access the metadata necessary to ensemble them. For ensembling, we have written a Python application named DetectionsToCSV that takes instances of GUARDIAN's JSON schema as input and produces an instance of the JSON schema that groups engineering signatures using the pairwise alignment procedure discussed at the beginning of this section. In addition to ensembling, DetectionsToCSV can produce a CSV spreadsheet summary of its output for quick review by an analyst.

T&E Results Analysis. We compared the groups DNA engineering signatures produced by GUARDIAN against insert signatures in the T&E samples by pairwise aligning all signatures' sequences in accordance with GUARDIAN's ensembling procedure described at the beginning of Methods. If at least one engineering signature in a GUARDIAN group matched an insert signature in a sample, then we called the sample a true positive (TP); otherwise, we called it a false negative (FN). If GUARDIAN produced any engineering signature for a sample containing no insert signatures, then we called it a false positive (FP); otherwise, we called it a true negative (TN). Metadata for T&E samples such as host organism, whether or not they contain an insert, their engineering signatures, and these signatures' sequences were obtained from the files generated by the FELIX T&E team and are listed in the Supporting Information.

In the 100 T&E sample we received, there are 1004 engineering signature representing 234 unique elements. Each T&E engineering signature has an ID (IF#) that is structured according the to the type of engineering it represents, including compound insertions and deletions, single element insertions and deletions, chromosomal inversions, transpositions and reassortments, single point mutations causing frameshifts, amber and ochre stop codons, base substitutions and transitions, and transformation of plasmids. For deletions and single-base changes, the T&E team annotated sequences on both sides of the change and designated them as "flank1" or "flank2", accordingly. The name of each IF element includes descriptors for any mutated gene, whether it generates a partial CDS, or whether it represents the insertion of a promoter, terminator, origin of replication, polyA-signal, protein tag, FRT, plasmid component, or "scar" left behind due to the type of sequence change.

■ ASSOCIATED CONTENT

Data Availability Statement

Source code and documentation for MIDOE are available on GitHub at https://github.com/raytheonbbn/midoe under the Apache License, Version 2.0. Evidence of engineering produced by GUARDIAN's modules is also available on GitHub from the same MIDOE repository. Sample sequencing data provided by the FELIX T&E team are at NCBI under the BioProject PRJNA607328. The reads are available from SRA and the assemblies are in GenBank. The accession numbers for these data are in the Excel file in the Supporting Information and are also available via the MIDOE repository.

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acssynbio.3c00398.

Excel file with sample metadata from FELIX T&E team, CSV file from FELIX T&E team linking sample IDs to engineering signature IDs, and FASTA file from FELIX T&E team with DNA sequences for engineering signatures headed by signature IDs (ZIP)

■ AUTHOR INFORMATION

Corresponding Author

Nicholas Roehner − *Raytheon BBN, Cambridge, Massachusetts 02138, United States;* orcid.org/0000-0003-4957-1552; Email: nicholas.roehner@rtx.com

Authors

Aaron Adler − *Raytheon BBN, Cambridge, Massachusetts 02138, United States;* orcid.org/0000-0002-7592-0763

Joel S. Bader − *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Brian Basnight − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*

Benjamin W. Booth − *Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States*

Jitong Cai − *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Elizabeth Cho − *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

Joseph H. Collins − *Department of Chemical Engineering, Worcester Polytechnic Institute, Worcester, Massachusetts 01609, United States*

Yuchen Ge − *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*

John Grothendieck − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115

Kevin Keating − *Department of Chemical Engineering, Worcester Polytechnic Institute, Worcester, Massachusetts 01609, United States;* orcid.org/0000-0001-6260-5572
Tyler Marshall − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Anton Persikov − *Department of Computer Science, Princeton University, Princeton, New Jersey 08544, United States*
Helen Scott − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Roy Siegelmann − *Department of Biomedical Engineering, Johns Hopkins University, Baltimore, Maryland 21218, United States*
Mona Singh − *Department of Computer Science, Princeton University, Princeton, New Jersey 08544, United States*
Allison Taggart − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Benjamin Toll − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Kenneth H. Wan − *Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States;* orcid.org/0000-0002-9203-1909
Daniel Wyschogrod − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Fusun Yaman − *Raytheon BBN, Cambridge, Massachusetts 02138, United States*
Eric M. Young − *Department of Chemical Engineering, Worcester Polytechnic Institute, Worcester, Massachusetts 01609, United States;* orcid.org/0000-0001-5276-2873
Susan E. Celniker − *Lawrence Berkeley National Laboratory, Berkeley, California 94720, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acssynbio.3c00398

Author Contributions

N.R., A.A., and F.Y. led GUARDIAN and analysis of the FELIX T&E results. N.R. designed the GUARDIAN data model and ensembling algorithm, and B.B. implemented them in MIDOE. J.G. developed the N-Gram module. T.M. developed Docker containers for all of GUARDIAN's modules. A.T. and H.S. developed the BED-DD module. B.T. developed the Targeted Search module. D.W. developed the BGAF module. J.S.B., E.C., J.C., Y.G., and R.S. developed the JHUARDIAN module. J.S.B., J.C., K.K., and E.M.Y. curated training data and performed SME review of T&E results. J.H.C., K.K., and E.M.Y. developed the Prymetime assembly pipeline. A.P. and M.S. developed the HMM module. K.H.W., B.W.B., and S.E.C. led the DNA sequencing, genome assembly, and annotation efforts to develop the FELIX T&E samples.

Notes

The authors declare no competing financial interest.
Approved for public release. Distribution is unlimited. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

■ ACKNOWLEDGMENTS

This research is based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) under Finding Engineering-Linked Indicators (FELIX) program contract #HR0011-15-C-0084. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. We thank the following individuals for providing samples for T&E: Becky Hess and Erin Bredeweg (PNNL), Jeff Chandler, Matthew Hopken, Antoinette Piaggio, and Mary Pantin-Jackwood (USDA NWRC), David Marciano and Olivier Lichtarge (Baylor College of Medicine), Pam Ronald (UCD), Javier Ceja-Navarro (NAU), John Gladden, Jay Keasling, Christopher Lawson, Hector Garcia Martin, and Vivek Mutlik (JBEI, LBNL), Michelle O'Malley (UCSB), Abby Kroken and Suzanne Fleiszig (UCB), and Matt Wargo (UV). We also thank Soo Park, Ryan Kenneally, Michael Neff, Quentin Lawrence, and Joanne Eichenberger of the Berkeley Drosophila Genome Project for cell growths, DNA isolation, and sequencing.

■ REFERENCES

(1) Pineda, M.; Moghadam, F.; Ebrahimkhani, M. R.; Kiani, S. Engineered CRISPR Systems for Next Generation Gene Therapies. *ACS Synth. Biol.* **2017**, *6*, 1614−1626.
(2) Wurtzel, E. T.; Vickers, C. E.; Hanson, A. D.; Millar, A. H.; Cooper, M.; Voss-Fels, K. P.; Nikel, P. I.; Erb, T. J. Revolutionizing Agriculture with Synthetic Biology. *Nat. Plants* **2019**, *5*, 1207−1210.
(3) Casini, A.; Chang, F.-Y.; et al. A Pressure Test to Make 10 Molecules in 90 Days: External Evaluation of Methods to Engineer Biology. *J. Am. Chem. Soc.* **2018**, *140*, 4302−4316.
(4) Lewis, G.; Jordan, J. L.; et al. The Biosecurity Benefits of Genetic Engineering Attribution. *Nat. Commun.* **2020**, *11*, 6294.
(5) Nielsen, A. A. K.; Voigt, C. A. Deep Learning to Predict the Lab-of-Origin of Engineered DNA. *Nat. Commun.* **2018**, *9*, 3135.
(6) Alley, E. C.; Turpin, M.; Liu, A. B.; Kulp-McDowall, T.; Swett, J.; Edison, R.; Von Stetina, S. E.; Church, G. M.; Esvelt, K. M. A Machine Learning Toolkit for Genetic Engineering Attribution to Facilitate Biosecurity. *Nat. Commun.* **2020**, *11*, 6293.
(7) Crook, O. M.; Warmbrod, K. L.; et al. Analysis of the First Genetic Engineering Attribution Challenge. *Nat. Commun.* **2022**, *13*, 7374.
(8) Clement, K.; Rees, H.; et al. CRISPResso2 Provides Accurate and Rapid Genome Editing Sequence Analysis. *Nat. Biotechnol.* **2019**, *37*, 224−226.
(9) Cancellieri, S.; Canver, M. C.; Bombieri, N.; Giugno, R.; Pinello, L. CRISPRitz: Rapid, High-Throughput and Variant-Aware In Silico Off-Target Site Identification for CRISPR Genome Editing. *Bioinform.* **2020**, *36*, 2001−2008.
(10) Collins, J. H.; Keating, K. W.; Jones, T. R.; Balaji, S.; Marsan, C. B.; Çomo, M.; Newlon, Z. J.; Mitchell, T.; Bartley, B.; Adler, A.; Roehner, N.; Young, E. M. Engineered Yeast Genomes Accurately Assembled from Pure and Mixed Samples. *Nat. Commun.* **2021**, *12*, 1485.
(11) 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 Res.* **2009**, *19*, 1117−1123.
(12) Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic Local Alignment Search Tool. *J. Mol. Biol.* **1990**, *215*, 403−410.
(13) McLaughlin, J. A.; Beal, J.; et al. The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering. *Front. Bioeng. Biotechnol.* **2020**, *8*, 1009.
(14) Beal, J.; Clore, A.; Manthey, J. Studying Pathogens Degrades BLAST-Based Pathogen Identification. *Sci. Rep.* **2023**, *13*, 5390.
(15) Cock, P. J. A.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. L. Biopython: Freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinform.* **2009**, *25*, 1422−1423.

(16) Wood, D. E.; Salzberg, S. L. Kraken: Ultrafast Metagenomic Sequence Classification Using Exact Alignments. *Genome Biol.* **2014**, *15*, R46.
(17) Lu, J.; Breitwieser, F. P.; Thielen, P.; Salzberg, S. L. Bracken: Estimating Species Abundance in Metagenomics Data. *PeerJ. Comput. Sci.* **2017**, *3*, No. e104.
(18) Langmead, B.; Salzberg, S. L. Fast Gapped-Read Alignment with Bowtie 2. *Nat. Methods* **2012**, *9*, 357−359.
(19) 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. *Bioinform.* **2015**, *31*, 1674−1676.
(20) Li, H.; Durbin, R. Fast and Accurate Short Read Alignment with Burrows−Wheeler Transform. *Bioinform.* **2009**, *25*, 1754−1760.
(21) Kolmogorov, M.; Yuan, J.; Lin, Y.; Pevzner, P. A. Assembly of Long, Error-Prone Reads Using Repeat Graphs. *Nat. Biotechnol.* **2019**, *37*, 540−546.
(22) Wick, R. R.; Judd, L. M.; Gorrie, C. L.; Holt, K. E. Unicycler: Resolving Bacterial Genome Assemblies from Short and Long Sequencing Reads. *PLoS Comput. Biol.* **2017**, *13*, No. e1005595.
(23) Marçais, G.; Delcher, A. L.; Phillippy, A. M.; Coston, R.; Salzberg, S. L.; Zimin, A. MUMmer4: A Fast and Versatile Genome Alignment System. *PLoS Comput. Biol.* **2018**, *14*, No. e1005944.
(24) Darling, A. E.; Mau, B.; Perna, N. T. progressiveMauve: Multiple Genome Alignment with Gene Gain, Loss and Rearrangement. *PLoS One* **2010**, *5*, No. e11147.
(25) Stolcke, A. SRILM: An Extensible Language Modeling Toolkit. In *Proceedings of International Conference on Spoken Language Processing* 2002.

https://doi.org/10.1021/acssynbio.3c00398
ACS Synth. Biol. 2024, 13, 1105−1115