Benchmarking Obfuscated Split Order Detection Methods[1]

Adam Jones
Imperial College London

Phil Palmer
safely.bio

Prateek Garg
Open Source Pharma

With
Apart Research

Abstract

Current DNA synthesis screening operates on individual oligonucleotides, leaving a critical gap: an adversary can split a hazardous gene across many short oligos that each pass screening, then assemble them post-delivery. This split-order evasion is especially concerning for benchtop synthesizers, which lack mandatory screening infrastructure. We present an open-source project for benchmarking and evaluating obfuscated split-order detection methods. Our pipeline includes (1) a parametric data generator producing realistic oligo pools from real NCBI genomic sequences, with configurable obfuscation including intron insertion, restriction-enzyme cassette insertion, synonymous codon substitution, and decoy oligo mixing; (2) six detection method implementations spanning exact-match approaches (k-mer hashing, Sourmash FracMinHash) and alignment-based approaches (BLAST, minimap2, MMseqs2); and (3) a standardised evaluation harness measuring accuracy and computational cost across difficulty axes. Using safe proxy viral genomes, we find that exact-match methods achieve perfect detection on unobfuscated pools but fail under codon substitution, while alignment-based methods, particularly minimap2 and MMseqs2, maintain robustness at practical speeds. The evaluation allows us to explore trade-offs between robustness of detection to obfuscation techniques, speed, and the number of false positives. All code and evaluation infrastructure are released openly.

[1] Research conducted at the AIxBio Hackathon, April 2026

1. Introduction

DNA synthesis screening is a foundational layer of biosecurity. Commercial providers screen orders against databases of sequences of concern (SOCs), but this system has a well-documented blind spot: split ordering. An adversary fragments a hazardous gene into short oligonucleotides that individually pass screening, orders them as a pool and assembles them post-delivery using techniques such as Gibson assembly.

This threat is particularly acute for benchtop DNA synthesizers, which are approaching the capability to produce viral-genome-length constructs and currently operate outside commercial screening frameworks. The Biosecurity Modernization and Innovation Act of 2026 (S.3741) mandates screening for commercial synthesis but does not adequately address benchtop devices. The IFP report on Securing Benchtop DNA Synthesizers identifies on-device split-order detection as a key technical mitigation. Edison, Toner & Esvelt (Nature Communications, 2026) empirically validated the threat by acquiring unregulated DNA fragments from dozens of providers sufficient to reconstruct 1918 influenza.

Despite the recognised importance of split-order detection, there is no standardised benchmark for evaluating methods. Our contributions are:

1. A parametric data generator producing realistic oligo pools from real NCBI sequences with configurable coverage, decoy ratios, and three biologically motivated obfuscation strategies.
2. Six detection method implementations spanning exact-match (k-mer hashing, Sourmash FracMinHash) and alignment-based (BLASTN, minimap2, MMseqs2) paradigms, benchmarked across difficulty axes.
3. An open-source, method-agnostic evaluation framework with a standardised result schema enabling reproducible comparison of current and future approaches.

2. Related Work

SecureDNA is a cryptographic screening platform detecting hazardous sequences in short windows ($\geq$30bp). Our k-mer method is conceptually similar, but SecureDNA screens individual sequences rather than reasoning about assemblability across an entire pool. IBBIS Common Mechanism uses HMM-based screening optimised for sequences above 150bp. Neither tool directly addresses pool-level split-order detection. Tayouri et al. demonstrated BLASTN-based approaches for detecting whether oligos could reconstitute a hazardous sequence; our BLAST implementation follows this line of work but situates it within a systematic benchmarking framework. The Microsoft/IBBIS/SecureDNA collaboration (Science, 2025) showed that AI-designed protein variants can evade screening, further motivating robust detection methods. Our work differs from existing work in providing robust detection algorithms that can capture typically-used obfuscation techniques like intron or restriction enzyme site intersections and codon switching, while also providing a benchmarking framework for six different methods, with customizable datasets along multiple difficulty axes.

3. Methods

3.1 Data Generation

Our parametric generator constructs oligo pools from real genomic sequences downloaded from NCBI GenBank. SOCs are safe proxy viral genomes (structurally representative but not controlled sequences); benign sequences are sourced from non-pathogenic organisms. A single configurable script controls the scenario: number of SOCs per pool, coverage fraction (1.0 = all tiles present), decoy ratio, and oligo length distribution (normally distributed with configurable mean and standard deviation).

Three obfuscation strategies model real-world evasion techniques, applied in a fixed pipeline order with each step respecting regions modified by previous steps to preserve biological validity:

- **Intron insertion:** Synthetic GT-AG introns at random positions, mimicking eukaryotic gene structure. Spliced out during expression but disrupts naive sequence matching.
- **RE cassette insertion:** Restriction enzyme recognition site pairs flanking nonce DNA. An attacker digests these out post-synthesis using the corresponding enzyme. Supports 15 common enzymes.
- **Synonymous codon substitution:** Codons replaced with synonymous alternatives preserving protein sequence and function while altering the DNA sequence, directly targeting similarity-based screening.

3.2 Detection Methods

We implemented six detection methods spanning two paradigms. All methods produce per-SOC scores between 0 and 1, thresholded at 0.5 for binary decisions, and inherit timing, memory measurement, and standardised JSON output from a shared base module.

Method	Speed	Benefits	Drawbacks
Exact k-mer	Fastest	Exact, no approximation	Zero tolerance for any sequence variation
Sourmash	Very fast	Memory-efficient at large DB scale	Same as exact k-mer + sampling noise at low coverage
Minimap2	Fast	Tolerates point mutations; single subprocess per pool	Short-read parameter traps; fragile on boundary-spanning oligos
MMseqs2	Moderate	Tolerates point mutations; scales best to large SOC DBs; native coverage/identity filtering	3 subprocesses per pool adds overhead
BLASTN-short	Slow	Gold standard; most sensitive seeding (word_size=7) for short reads	Per-pool database reload dominates runtime
BLASTN	Slow	Standard alignment baseline	Per-pool database reload; less sensitive than blastn-short for short oligos

Table 1. Detection methods with trade-offs. Exact-match methods (top two) are fast but brittle to sequence changes; alignment-based methods (bottom four) tolerate mutations at higher cost.

3.3 Evaluation Framework

The framework enforces strict separation between method execution and analysis. Methods write one JSON per pool following a standardised schema (scores, decisions, threshold, runtime, memory, parameters). Evaluation scripts join results against manifest CSVs to compute precision, recall, F1, and PR-AUC sliced by batch parameters. Adding a new method requires implementing only a single function; timing, memory tracking, and result serialisation are inherited automatically.

4. Results

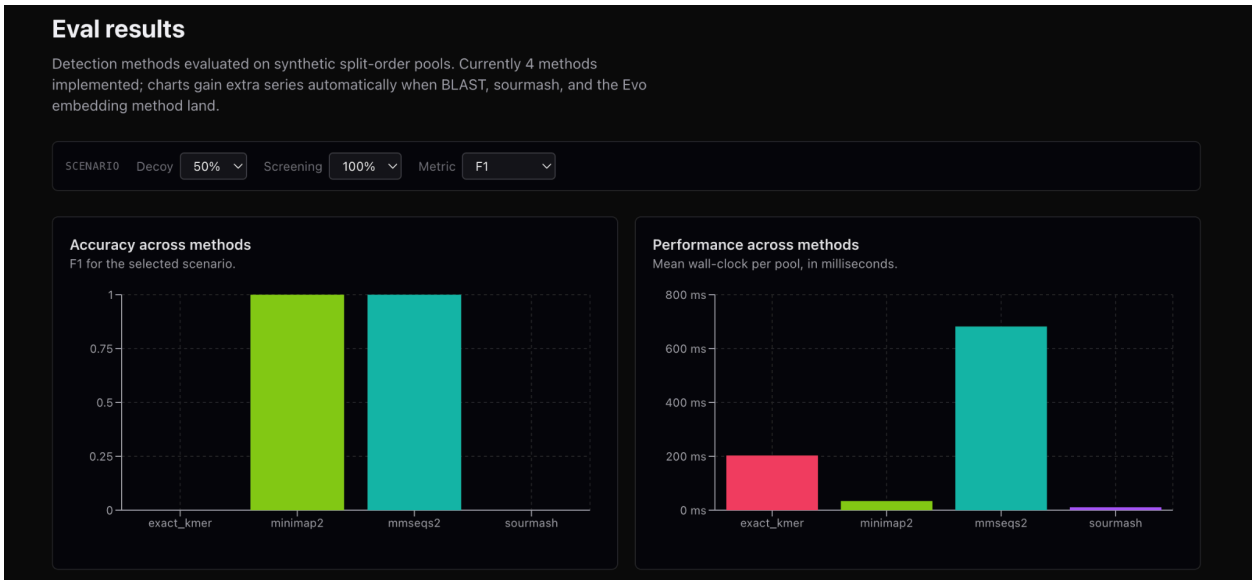

Fig 1.Accuracy of 4 different detection methods given obfuscation. Exact-match methods (exact k-mer and sourmash) are fast but brittle to sequence changes – their F1 scores drop to near zero with obfuscation; alignment-based methods (minimap2 and mmseq2) have near perfect F1 scores and tolerate mutations but mmseq2 has a much higher time cost.

We evaluated all six methods across difficulty axes using pools generated from real NCBI sequences.

Baseline detection. On unobfuscated pools with full SOC coverage and 50% decoy ratio, all six methods achieve perfect or near-perfect separation between SOC-containing and benign pools.

Coverage fraction. As SOC coverage decreases, detection scores drop proportionally for all methods. At 60% coverage, methods still correctly classify pools at the 0.5 threshold. Below ~50% coverage, exact-match methods begin to fail; alignment-based methods degrade more gracefully due to partial-hit accumulation.

Obfuscation robustness. This is where the methods diverge most sharply. Codon substitution causes exact-match methods (k-mer and Sourmash) to fail, because a single nucleotide change destroys all overlapping k-mers. Alignment-based methods degrade gracefully under codon substitution. Intron and RE cassette insertions present a subtler challenge: exact-match methods again fail, while alignment-based methods are more tolerant to sequence obfuscation. Under combined obfuscation, only alignment-based methods maintain useful detection.

Computational performance. K-mer hashing and Sourmash process pools in under 1 second. Minimap2 is the fastest alignment-based option (single subprocess per pool). MMseqs2 is moderately fast but incurs overhead from three subprocesses per pool; it may scale best to large SOC databases due to native coverage and identity filtering. BLAST is the slowest due to per-pool database reloading. For on-device benchtop screening, k-mer or Sourmash provides a fast initial screen; minimap2 offers the best speed/robustness balance.

Interactive demo. A live demo is available at https://split-order-detection.vercel.app/. It visually illustrates the split-order attack and shows detection methods producing verdicts, alongside comparison plots from the evaluation framework.

Taken together, the results show that the split-order detection problem is solvable at the single-pool level: for every obfuscation strategy we tested, at least one method family maintains useful detection. The critical variable is which speed/robustness trade-off is acceptable for a given deployment context – real-time on-device screening versus batch server-side analysis.

5. Discussion and Limitations

Discussion

Our results demonstrate that split-order detection is technically feasible even under intensive obfuscation: alignment-based methods maintain useful detection accuracy even when codon substitution, intron insertion, and RE cassette insertion are applied simultaneously. This is encouraging for the biosecurity community, as it means the screening gap is not fundamentally intractable. However, detection capability alone does not close the gap. The harder challenges are operational: implementing screening across vendors while maintaining privacy and intellectual property, correlating orders across providers and time windows, and navigating the governance constraints of shared screening infrastructure, particularly for benchtop devices operating outside existing regulatory frameworks.

Limitations

Our proxy SOCs are real viral genomes but not drawn from controlled-pathogen databases, limiting ecological validity. We do not address cross-vendor correlation. Our obfuscation strategies, while biologically motivated, may not cover all adversary techniques, including functional retention

with different sequences. However, functionally similar sequences can be added to the SOC and our methods will just map the analysis on that extended database. Scaling behaviour with larger databases would need investigation though. All work was completed within a hackathon timeframe and results should be treated as preliminary.

Future Work

Natural extensions include genomic language model embeddings (e.g. Evo 2) as a detection paradigm, expanded obfuscation strategies (gene refactoring), larger SOC databases including sequences with functional similarity, on-device benchmarking on representative benchtop hardware, and the cross-vendor correlation problem. The framework accommodates most of these without structural changes.

6. Conclusion

We have presented an open-source evaluation for obfuscated split-order detection methods in DNA synthesis biosecurity, with six implemented methods spanning exact-match and alignment-based paradigms. Our central finding is that split-order detection is technically tractable even under realistic obfuscation. However, the screening gap exists not only because of a lack of detection tools, but also because the operational infrastructure to deploy it does not yet exist. Cross-vendor order correlation, SOC database privacy, and regulatory coverage of benchtop devices remain the binding constraints. We have developed and can release all code and evaluation infrastructure to researchers to support continued progress on these challenges.

Code and Data

Information Hazard:

Part of our code-base may allow adversarial actors to generate obfuscated sequences and split-oligos for synthesizing sequences of concern, and hence the primary github repository has been made private but a sub-repository without the information hazard oligo creation code has been made public for the hackathon evaluation. Also, the Demo URL has been put behind a password “blueteam”. We encourage the evaluators to watch the ~5-minute Demo Video that has been linked below which takes one through the demo well.

1. Public Code repository: [GitHub link — https://github.com/PhilPalmer/split-screen.git]
2. Private Code repository [contains potential information hazard: https://github.com/PhilPalmer/split-order-detection/
3. Data/Datasets: Generated using real NCBI GenBank sequences via the included data generation pipeline. No controlled sequences are used or distributed.
4. Demo: [DEMO Video: https://drive.google.com/file/d/1_WS6thpFAJUW1fJK6heSqZs8vN17u7e-/view?usp=sharing
   Live Interactive Demo URL — https://split-order-detection.vercel.app/]

References

[1] SecureDNA Project. SecureDNA Documentation. https://securedna.org/
[2] IBBIS. Common Mechanism for DNA Synthesis Screening (commec). https://ibbis.bio/
[3] Edison, M., Toner, E., & Esvelt, K. (2026). Assembling unregulated DNA segments bypasses synthesis screening. Nature Communications.
[4] Institute for Progress. Securing Benchtop DNA Synthesizers. 2025.
[5] Biosecurity Modernization and Innovation Act of 2026, S.3741.
[6] OSTP. Framework for Nucleic Acid Synthesis Screening. April 2024.
[7] Microsoft, IBBIS, SecureDNA et al. (2025). Strengthening nucleic acid biosecurity screening against generative protein design tools. Science.
[8] Sherman, A. et al. (2026). Analysis of the Security Design, Engineering, and Implementation of the SecureDNA System. NDSS.
[9] Sentinel Bio. (2026). Why We're Doubling Down on Synthesis Screening.

LLM Usage Statement

We used Claude (Anthropic) as a coding partner for implementing the data generator, detection methods, and evaluation framework. Claude also assisted in draft generation and proofreading this report. Grok (xAI) was also used for research. All results and claims were independently verified. Architecture, design decisions, and scientific claims are the authors' own.