BioShield: A 6-Layer Defense-in-Depth Pipeline for AI-Resistant DNA synthesis Screening [1]

N. Mohana Krishna 1
Independent Researcher

With
Apart Research

Abstract

*Current DNA synthesis screening infrastructure relies on exact sequence alignment (BLAST/HMM), which fails against AI-designed codon-optimized pathogen variants. We present **BioShield**, a 6-layer Defense-in-Depth screening pipeline that enforces biological constraints at multiple independent levels to catch AI-generated, novel, and evasion-variant pathogens. Our layers include K-mer fingerprinting, a stacked ML ensemble (Random Forest + XGBoost + Logistic Regression meta-learner), protein impact analysis with ESM-2 embedding architecture, host codon bias detection, protease cleavage site scanning, and RNA secondary structure analysis. The pipeline incorporates sliding window architecture, bifurcated micro-sequence routing, and canonical k-mer extraction to defeat dilution, split-order, and reverse-complement evasion attacks respectively. Trained on 44,218 real NCBI genomic sequences with extreme AI-evasion augmentation, BioShield achieves 0.969 ROC-AUC and passes all 11 hardened test scenarios. We demonstrate seamless integration with the IBBIS Common Mechanism as "IBBIS v2.0," catching threats that legacy screening misses entirely.*

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

1. Introduction

The Problem

AI biological design tools are advancing faster than biosecurity infrastructure can adapt. Open-weight models like Evo2 (40B parameters, trained on 128,000+ genomes) and protein design tools like RFdiffusion enable the generation of novel biological sequences with decreasing expertise requirements [1]. Simultaneously, benchtop DNA synthesizers are approaching virus-length printing capability [2].

*The current industry standard for DNA synthesis screening, the **IBBIS Common Mechanism** (`commec`), relies on BLAST and HMM-based exact sequence alignment against known pathogen databases [3]. This approach has a critical vulnerability: if an AI tool shuffles the codons of a deadly pathogen to produce the same lethal protein using different DNA "spelling," the exact alignment breaks, and the sequence passes screening undetected.*

Our main contributions are:
We present Meta-BioShield, a multi-layered AI-defense module designed as an upgrade to the IBBIS ecosystem. Our main contributions are:

1. *"A 6-layer Defense-in-Depth architecture" that enforces independent biological constraints (statistical, machine learning, protein-level, codon-level, motif-level, and structural), making simultaneous evasion of all layers computationally infeasible*

2. *"Four novel security innovations" — sliding window screening, bifurcated micro-sequence routing, canonical k-mer extraction, and closed-weight deployment — that defeat specific real-world evasion attack vectors (Trojan Horse dilution, split-order attacks, reverse-complement evasion, and adversarial AI probing).*

3. *"A stacked ML meta-learner" trained on 44,218 real NCBI sequences with extreme AI-evasion augmentation (15% point mutations, fragment insertion, reverse-complement variants), achieving 0.969 ROC-AUC on the hardest dataset configuration.*

2. Related Work

IBBIS Common Mechanism*[3] is the open-source industry standard for DNA synthesis screening. It uses BLAST alignment against curated pathogen databases and HMM profiles. While effective against known, unmodified sequences, it is fundamentally vulnerable to AI-designed codon-optimized variants that preserve protein function while breaking sequence-level identity.*

SecureDNA *[4], co-founded by Kevin Esvelt, uses cryptographic hashing for privacy-preserving screening. It addresses the deployment and privacy challenge but does not specifically target AI-generated evasion variants at the biological feature level.*

ABC-Bench*[5] provides agentic biosecurity benchmarks for evaluating AI capabilities in biological contexts, establishing the evaluation framework our work builds upon.*

BioShield differs from all existing approaches by operating at "six independent biological constraint levels" rather than relying solely on sequence alignment. When an AI tool optimizes a sequence to evade one layer, it is highly likely to trigger anomalies detectable by another — creating a "Swiss cheese" defense where the holes in each layer are covered by the others.

3. Methods

3.1 Architecture Overview

BioShield is implemented as a pure Python package with a plugin architecture. Each screening layer operates independently, and a Verdict Engine aggregates results using a risk-based threshold: 0 flags = PASS, 1–2 flags = FLAG (human review), 3+ flags = REJECT (auto-deny).

3.2 The Six Screening Layers

1. ***"Layer 1 — K-mer Fingerprinting."*** *Computes normalized k-mer frequency distributions (k=4, 5, 6) and measures cosine similarity against a curated threat database. This catches heavily mutated pathogen variants that retain statistical "fingerprints" even when exact alignment fails.*

2. ***"Layer 2 — ML Ensemble + Meta-Learner."*** *Extracts 10 biological features from each sequence: GC content, GC/AT skew (strand-averaged), CpG dinucleotide ratio, Lempel-Ziv compression complexity, canonical k-mer entropy (k=3, k=4), longest ORF ratio (both strands), and repeat density. These features are fed into a Random Forest (500 trees, depth 20) and XGBoost (800 trees, depth 8, GPU-trained) ensemble. A Logistic Regression meta-learner then combines their outputs, learning optimal trust weights for each model (RF weight: 6.70, XGB weight: 1.40 — the meta-learner discovered RF is 4.8× more reliable on this data).*

3. ***"Layer 3 — Protein Impact Analyzer."*** *Translates the input DNA in all 6 reading frames, extracts ORFs, and aligns them against a curated UniProt toxin/virulence factor database. Architecture supports ESM-2 protein language model embeddings [6] for detecting de novo toxins by functional 3D shape similarity (graceful fallback when ESM-2 is not installed).*

4. ***"Layer 4 — Host Codon Bias Screener."*** *Computes the Codon Adaptation Index (CAI) relative to *Homo sapiens* codon usage tables (Kazusa). A high CAI (⩾0.75) indicates the sequence was codon-optimized for efficient human cell expression — a strong marker of deliberate engineering for human infectivity.*

5. ***"Layer 5 — Protease Cleavage Site Screener."*** *Scans all 6 translated reading frames for 8 known protease recognition motifs: Furin (polybasic R-X-[KR]-R, RRAR, PRRA), Thrombin (LVPR[GS]), TEV (ENLYFQ[SG]), Factor Xa (I[ED]GR), Enterokinase (DDDDK), and Anthrax PA (RKKR). These are the tiny, unalterable "activation switches" that toxins and viruses must retain to become functional — an AI can scramble 10,000 bases but cannot change these 6–12 residue motifs without disabling the weapon.*

6. ***"Layer 6 — RNA Secondary Structure Screener."*** *Estimates RNA folding potential by computing palindromic stem density (potential hairpin-forming regions) and GC bond stability. When ViennaRNA is installed, it performs full Minimum Free Energy (MFE) computation. Dangerous viruses (SARS-CoV-2, Ebola, HIV) rely on specific RNA 3D structures for replication; if AI shuffles codons to evade DNA-level screening, it risks destroying these essential structures.*

3.3 Security Innovations

1. ***"Sliding Window Architecture (Fix #1)."*** *Sequences longer than 500bp are chopped into overlapping windows (500bp, 50% overlap). Each window is screened independently. If *any single window* flags on *any layer*, the entire order is flagged. This defeats the "Trojan Horse" dilution attack where a small threat is hidden inside a large safe sequence, causing global features to average to safe values.*

2. *"**Bifurcated Pipeline (Fix #3)**." Sequences shorter than 100bp bypass the ML and K-mer layers (which produce noisy, unreliable results at that length) and are routed to a specialized MicroSequenceScreener that performs exhaustive exact-match against a database of critical pathogen initiation motifs. This defeats split-order attacks where a virus is ordered in many tiny pieces.*

3. *"**Canonical K-mers (Fix #4)**." All k-mers are reduced to their lexicographically smaller form (the minimum of the k-mer and its reverse complement). GC/AT skews are averaged across both strands. ORFs are checked on both strands. This makes the entire feature extraction pipeline strand-agnostic, defeating reverse-complement evasion.*

4. *"**Closed-Weight Deployment**." The code architecture is open-sourced; the trained model weights and threat databases are not. This prevents adversarial AI from downloading BioShield and running millions of offline evasion tests. Bad actors must submit real DNA orders to test the system, triggering cost, speed, and legal enforcement traps.*

3.4 Training Data and Procedure

Models were trained on Kaggle (T4 GPU) using real genomic data fetched from NCBI via Biopython:

Category	Organisms	Chunks
Safe	*E. coli* (U00096.3), *S. cerevisiae* (3 chromosomes)*	*11,998*
Threat	*Ebola, Smallpox, SARS-CoV-2, Nipah, Marburg, Anthrax, MERS, SARS-1*	*10,740*
AI-Evasion	*15% point mutation + fragment insertion on threat sequences*	*10,740*
Reverse-Complement	*Full RC of all threat sequences*	*10,740*
"Total"		***"44,218"***

Class imbalance (12K safe vs 32K threat) was addressed using `class_weight='balanced'` (RF) and `scale_pos_weight` (XGB).

4. Results

4.1 ML Model Performance

Model	ROC-AUC	Notes
Random Forest	*0.9686*	*500 trees, balanced weights*
XGBoost	*0.9667*	*800 trees, GPU-trained*
"Meta-Learner (LR stacking)"	*"0.9690"*	***Outperforms both base models***

The meta-learner's learned weights (RF: 6.70, XGB: 1.40) reveal that the Random Forest is 4.8× more trusted than XGBoost on this data, likely because RF handles the high-dimensional canonical k-mer entropy features more robustly.

*"**Table 1: Classification Report (Meta-Learner)**"*

Class	*Precision*	*Recall*	*F1-Score*
Safe	*0.92*	*0.84*	*0.88*
Threat	*0.94*	*0.97*	*0.96*
"Accuracy"			***0.96***

Table 2: Hardened Test Suite Results (v3.0 — 6 Layers, 11/11 Passed)

Test Scenario	*Verdict*	*Key Layer(s)*
Safe GFP (lab-engineered)	*REJECT*	*CodonBias (CAI=0.75), Cleavage, Protein*
Known Anthrax	*REJECT*	*All 6 layers flagged*
Trojan Horse (200bp threat in 2700bp safe)	*REJECT*	*Sliding Window + K-mer + ML*
Short Safe (37bp)	*PASS*	*MicroSequenceScreener*
Split-Order Anthrax PA (18bp)	*FLAG*	*MicroSequenceScreener*
Reverse-Complement Anthrax	*REJECT*	*Canonical K-mers + ML (0.93)*
Human Codon-Optimized	*FLAG*	*CodonBias (CAI=1.000)*
Furin RRAR Cleavage Site	*REJECT*	*CleavageSite (CRITICAL) + ML + Protein*
RNA Hairpin Pattern	*FLAG*	*RNAFolding (density=0.919)*
Empty Sequence	*PASS*	*Handled gracefully*
Chimeric Splice	*REJECT*	*6/6 layers flagged*

4.3 IBBIS v2.0 Integration Demo

We demonstrated seamless integration with legacy IBBIS via a simulated cloud API:

> ***"Scenario 1 (Standard Anthrax):"*** *IBBIS → REJECT. BioShield → REJECT. Both systems agree.*

"Scenario 2 (AI Codon-Shuffled, 20% mutation):" *IBBIS → **PASS** (missed it). BioShield → **REJECT** (ML prob=0.92, Meta-Learner). BioShield caught what IBBIS missed.*

5. Discussion and Limitations

Implications

BioShield demonstrates that multi-layered biological constraint enforcement can dramatically raise the bar for AI-assisted bioweapon design. By requiring evasion of six independent biological layers simultaneously, the computational cost for adversaries increases exponentially with each added layer. This aligns with the "Defense in Depth" security philosophy: each layer is a slice of Swiss cheese, but stacked together, nothing passes through.

Limitations

*-**False Positives:** The 6-layer system is intentionally aggressive. Legitimate lab constructs like GFP (a harmless fluorescent protein) are flagged as REJECT because they are codon-optimized and contain cleavage sites. In production, this would require a human review workflow.*

*- **Mock Databases**: The K-mer threat database and UniProt toxin database used in this prototype are small demonstrations. A production system requires comprehensive, regularly updated databases.*

*- **Simulated IBBIS Integration**: The legacy IBBIS layer is simulated (MockIbbisEngine) because the full `commec` system requires Linux infrastructure. The integration architecture is production-ready; only the server deployment is mocked.*

*- **Training Data Scale**: 44K sequences is substantial for a hackathon but small for production. Real-world deployment would require millions of sequences and continuous retraining.*

*- **ESM-2 and ViennaRNA**: These are architecturally integrated with graceful fallbacks but were not deployed with full models during the hackathon due to compute constraints.*

Future Work

1. PyO3 Rust Bridge: Migrate K-mer computation to Rust for 100× throughput improvement.

2. Million-Scale Training: Train on millions of sequences from expanded NCBI databases with
dedicated compute infrastructure.

3. Full ESM-2 Deployment: Enable protein shape embedding for de novo toxin detection.

4. ViennaRNA Integration: Enable full RNA Minimum Free Energy folding analysis.

5. Live IBBIS Server Merge: Deploy as a plugin within the IBBIS Linux server infrastructure.

6. Conclusion

We presented BioShield, a 6-layer Defense-in-Depth DNA synthesis screening pipeline that addresses the critical gap in current biosecurity infrastructure: vulnerability to AI-designed evasion variants. By enforcing independent biological constraints at the statistical, machine learning, protein, codon, motif, and structural levels, BioShield catches threats that legacy alignment-based systems miss entirely. Trained on 44,218 real NCBI sequences with extreme augmentation, the system achieves 0.969 ROC-AUC and passes all 11 hardened test scenarios, including Trojan Horse dilution attacks, split-order micro-sequences, and reverse-complement evasion. Our closed-weight deployment philosophy ensures that the system cannot be trivially reverse-engineered by adversarial AI.

Code and Data

- ***Code repository**: [https://github.com/mkrishna793/Meta-BioShield-for-Global-]*
- ***Data/Datasets**: Real genomic sequences fetched from NCBI (accessions listed in Methods). Training performed on Kaggle (T4 GPU).*
- ***Note on info-hazard:** Model weights and threat databases are intentionally not published. The architecture is open; the trained parameters are deployed as a closed API to prevent adversarial bypass.*

Author Contributions

N. Mohana Krishna conceived and designed the complete solution architecture, identified all vulnerability classes and their biological countermeasures, designed the 6-layer Defense-in-Depth strategy, and directed all implementation and testing.

References

1. *Nguyen, E., et al. "Sequence modeling and design from molecular to genome scale with Evo." *Science*, 2024. (Evo2: 40B parameter DNA language model)*
2. *Apart Research. "AIxBio Hackathon 2026: DNA Screening & Synthesis Controls." Track 1 problem statement.*
3. *International Biosecurity and Biosafety Initiative for Science (IBBIS). "Common Mechanism for DNA Synthesis Screening." Open-source screening tool (`commec`).*
4. *Esvelt, K., et al. "SecureDNA: A cryptographic platform for universal DNA synthesis screening." MIT Media Lab, Sculpting Evolution Group*
5. *Kleinman, A., et al. "ABC-Bench: An Agentic Biosecurity Benchmark." NeurIPS 2025.*
6. *Lin, Z., et al. "Evolutionary-scale prediction of atomic-level protein structure with a language model." *Science*, 2023. (ESM-2)*
7. *Chen, T., Guestrin, C. "XGBoost: A Scalable Tree Boosting System." *KDD*, 2016.*

Appendix *(optional)*

Limitations

- ***False Positive Rate:*** *Aggressive multi-layer screening flags legitimate synthetic biology constructs (e.g., codon-optimized GFP). Production deployment requires a human review pathway for flagged orders."(While a human-in-the-loop workflow provides an additional security layer, it presents significant scaling challenges for processing millions of commercial DNA orders.)"*

- ***Database Coverage:*** *Threat K-mer profiles and toxin databases are demonstrative. Real-world efficacy depends on comprehensive,* ***continuously updated databases.***

- ***Scalability:*** *The sliding window approach increases computation linearly with sequence length. For very long sequences (>100kb), optimization or the planned Rust migration would be necessary. (future development)*

- ***Model Staleness:*** *ML models trained on current pathogen data may not generalize to entirely novel threat classes without periodic retraining.(need to solve in next versions)*

Dual-Use Risks

- ***Evasion Intelligence:*** *Publishing detailed descriptions of which biological features the ML model relies on could theoretically inform adversaries about which features to manipulate. We mitigate this by keeping trained weights closed and publishing only the architectural principles.*

- ***False Sense of Security:*** *No screening system is unbreakable. BioShield raises the cost and difficulty of evasion but should not be treated as a guarantee. It must be part of a broader biosecurity ecosystem including physical security, customer verification, and law enforcement.*

Responsible Disclosure

During this project, we identified that the IBBIS Common Mechanism's reliance on exact sequence alignment makes it vulnerable to AI codon-optimization attacks. This is a known limitation in the biosecurity community. Our response was to build a defensive tool (BioShield) rather than demonstrate offensive capabilities.

Ethical Considerations

- All training data was sourced from publicly available NCBI databases. No novel pathogen sequences were generated.

- The AI-evasion variants used for training are random point mutations, not biologically viable gain-of-function modifications.

- We deliberately chose a closed-weight deployment model to prevent misuse.

LLM Usage Statement

AI coding tools were used to assist with implementation and code generation. The complete solution architecture, biological defense strategy, vulnerability identification, and all design decisions were conceived and directed by the author. All results and claims were independently verified through automated testing.

— I, the author (N. Mohana Krishna), leveraged Gemini AI primarily for research and identifying related work, such as IBBIS and SecureDNA, concerning the appropriate Python libraries for this bioinformatics project.

Initially, I developed a V1 solution. To build this, I used Google Antigravity IDE (as I'm a "vibe coder") and trained it on Kaggle using a GPU. The code was then pushed to GitHub.

Further verification with Gemini AI indicated the need for more development to suit real-world applications. Consequently, I scaled the initial 3-layer architecture to 6 layers and incorporated a "Meta-Learner (LR stacking)." After conducting tests, the updated code was pushed.

My use of AI was focused on accelerating the building process and researching potential loopholes. Crucially, the solution remains entirely my own; I did not ask the AI to generate the solution itself.