Unintended Creation or Injection of Antisense Promoter Motifs During Codon Optimization: A Cyber-Biosecurity Risk

Elad Carmi †, Roni Glikman †, Yuval Dorfan *

Posted Date: 24 February 2026

doi: 10.20944/preprints202602.1456.v1

Keywords: codon optimization; antisense promoter; cyber-biosecurity; motif injection; synthetic biology; DNA design; E. coli; translation efficiency; codon bias

 doi:10.20944/preprints202602.1456.v1

Article

Unintended Creation or Injection of Antisense Promoter Motifs During Codon Optimization: A Cyber-Biosecurity Risk

Elad Carmi [1,†], Roni Glikman [1,†] and Yuval Dorfan[1] *

[1] Faculty of Electrical Engineering, Holon Institute of Technology

[†] Equal contribution.

* Correspondence: dorfany@gmail.com

Abstract

Codon optimization is a cornerstone technique in synthetic biology and biotechnological production, aimed at enhancing heterologous protein expression through synonymous codon substitutions. While optimization traditionally focuses on forward-strand translation efficiency, its impact on the complementary DNA strand is not always examined carefully enough. In this study, we investigate whether codon optimization inadvertently introduces antisense motifs, specifically bacterial antisense promoter (e.g., 'TATAAT'), and whether such motifs can be silently injected into coding sequences on purpose without altering protein output. We developed a computational pipeline that (i) scans optimized sequences for antisense motifs, natural or synthetic unintended motifs; (ii) implements a silent injection algorithm that preserves amino acid sequence; and (iii) evaluates injection feasibility across a large genomic dataset. These components can also lead to useful scanning of synthetic sequences, way before they are ordered. It has the potential to save a lot of time and money that might be spent in wet labs that are using wrong sequences. Their experiments often fail due to predictable reasons, while these failures can be avoided using the software we developed, which is published here as an open source for academia and industry. In a dataset of 484,741 protein-coding sequences, only 4.8 % naturally contained the motif, yet 77.28 % of motif-free sequences permitted silent injections. We extend these findings with codon bias analysis, derive analytical bounds for injection complexity, and propose computational defenses. These results uncover a novel cyber-biosecurity vulnerability in DNA design pipelines, emphasizing the need for bi-directional screening in codon optimization tools.

Keywords: codon optimization; antisense promoter; cyber-biosecurity; motif injection; synthetic biology; DNA design; E. coli; translation efficiency; codon bias

1. Introduction

DNA synthesis has changed rapidly in the past few decades[1], opening new opportunities like synthetic biology[2], alongside new challenges. The field of molecular biology has seen significant progress in optimizing genetic sequences[3] Codon optimization exploits degeneracy in the genetic code to tailor DNA sequences for improved expression in heterologous hosts[4–6]. The method adjusts synonymous codon frequencies to match host codon usage bias, thereby increasing protein yield and translational efficiency[7–10]. Such approaches underpin major applications in gene therapy, vaccine development, and industrial biotechnology, including mRNA vaccines and recombinant protein production systems.

Historically, codon optimization algorithms prioritize metrics such as the Codon Adaptation Index (CAI)[11], GC content, and minimization of unfavorable mRNA secondary structure[12]. Advances include deep learning models that capture sequential codon context for refined optimization[13–15]. Despite these improvements, most tools remain insensitive to potential

 doi:10.20944/preprints202602.1456.v1

functional motifs introduced on the complementary strand, particularly those that could act as antisense promoters. We focus on *E. coli*, since it is a well-studied model organism[16, 17], just for the case study, but the same software tools and conclusions apply to all other organisms.

Promoter motifs such as the bacterial '−10' element 'TATAAT' are critical regulatory elements for transcription initiation[18]. When observed in reverse complement ('ATTATA')[19], they may function as unintended antisense promoters. Although motif engineering has been considered in some codon design contexts, the explicit risk of inadvertently creating antisense promoters and other motifs has not yet been systematically characterized. Moreover, many free and commercialized codon optimization tools are widely used as is, although they do not always scan for many relevant motifs. These tools' typical users, the average biologist, are often naïve and tend to trust the output sequence blindly.

The convergence of synthetic biology and cyber-security has given rise to a new interdisciplinary domain: cyber-bio-security[20–24]. Within this space, codon optimization emerges not only as a tool for improving protein expression but also as a potential vector for unintended regulatory behavior and malicious exploitation. As the capacity to synthesize and edit DNA becomes more accessible and codon optimization is increasingly performed using automated tools, the potential for errors or security loopholes grows. The rise of cyber-biosecurity highlights threats at the intersection of information security and biology. Synthetic DNA can encode harmful elements—not only biological agents but also computational payloads, like those used for SW storage[25]. While most research focuses on software and hardware vulnerabilities, the prospect of intentional manipulation of biological regulatory elements via sequence design remains understudied. Another important factor is the flourishing of Artificial Intelligence (AI)[9]. The usage of machine learning to detect anomalies in synthetic DNA can be very useful. By training classifiers on statistical properties of natural versus synthetic sequences, even subtle deviations introduced during optimization could be detected computationally. A key contribution of that work is the demonstration that synthetic DNA could be engineered to exploit vulnerabilities in bioinformatics software, including buffer overflows triggered during sequence parsing. Their proof-of-concept attack transformed a DNA sequence into a vehicle for remote code execution—blurring the line between biological and computational domains. From a methodological standpoint, their work reinforces the necessity of anomaly detection algorithms in bioinformatics pipelines.

At a broader conceptual level, Elgabry et al. [22] framed synthetic biology as a potential enabler of novel forms of cybercrime. Their review presented a criminological perspective on technologies such as gene editing, bio-malware, and antisense-driven payloads, outlining scenarios where biological material could be used to circumvent security protocols or mislead forensic analysis. They emphasized the importance of developing preemptive security measures grounded in both biology and criminology. Beyond identifying threats, some researchers have explored how biological principles can inform computational defense mechanisms[26]. They proposed a bio-inspired framework for intrusion detection in network security, in which digital data was encoded as amino acid sequences and analyzed using models adapted from molecular biology. Their approach leveraged the inherent pattern recognition capabilities of bioinformatics tools to enhance detection accuracy in cybersecurity contexts. Although not directly related to DNA manipulation, this methodological crossover is significant. It suggests that tools developed for motif detection and codon analysis in synthetic biology can be repurposed or adapted to other domains, and vice versa.

In this work, we address two questions: (i) Does codon optimization inadvertently generate antisense promoter motifs? (ii) Can such motifs be deliberately injected into coding sequences without altering the encoded protein? [Figure 1] We present a computational framework that examines both biological and security implications of codon redesign, supported by quantitative data and analytical modeling.

We present various simulations performed using codon-optimization tools available online, both open-source and commercial. In addition, we tested and calculated the probability of various motifs in DNA sequences. Our delivery is an application that contains several testing options for

 doi:10.20944/preprints202602.1456.v1

customers, including comparing sequences between different optimization tools, searching for problematic motifs, and performing attempts to inject promoter motifs into the complementary strand without changing the original protein sequence.

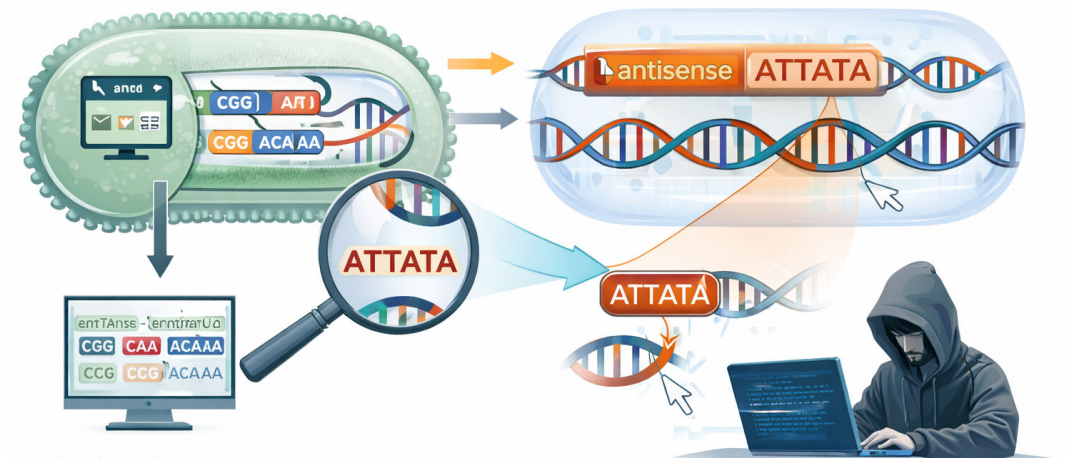

Figure 1. Web-based codon optimization SW can suffer from either unintended or silent injection of harmful motifs, like anti-sense promoters. The SW naïve user (or even a more sophisticated one) might not identify problematic DNA sequences .

2. Materials and Methods

2.1. The Promoter '-10' Motif: An Example Used for our Proof of Concept (POC)

Promoters play a critical regulatory role in gene expression by controlling when, where, and to what extent a gene is transcribed. Promoters typically contain specific motifs [17] recognized by transcriptional machinery. In bacteria, a promoter consists of:

- The -10 Region: A sequence positioned ~10 nucleotides upstream, which is crucial for DNA unwinding and formation of the open transcription complex.
- The -35 Region: A sequence located ~35 nucleotides upstream of the transcription start site, responsible for initial recognition by RNA polymerase sigma factor.

As a case study, we focus on the '-10' motif for various reasons. Expanding the results to longer promoters and/or enhancers is straightforward.

2.2. Dataset Choice and Preprocessing

We compiled a dataset of 484,741 coding sequences from *Escherichia coli* via the online database[27]. We focused on *E. coli* due to its well-characterized genome. Sequences shorter than 300 bp or containing ambiguous nucleotides were excluded. When focused on the classical '-10' motif, we looked for Motif-free sequences and excluded those containing the reverse-complement motif, i.e., 'ATTATA'.

2.3. Codon Optimization Framework: Analyzing Popular Tools' Output Sequences

Codon optimization alters the codon usage of a gene to align with the preferred codons of the host organism [28], thereby potentially improving protein production. Different amino acids have codon usage biases, so adjusting the gene's codons to match these biases can significantly improve expression efficiency. There are many types of optimization tools, and the algorithm by which each tool works is different. Some tools can only be used as a 'black box', meaning the customer cannot know what considerations were made in optimizing a protein for a specific organism. These 'black box' observations lead us to a limited set of conclusions since these tools often use random seeds,

 doi:10.20944/preprints202602.1456.v1

meaning the optimization results for a given protein vary with time. On the other hand, there are open-source tools. Our research utilized an open-source codon optimization. We used a custom Python-based tool named Codon Optimization QA. This tool incorporates codon usage tables for the model organism of choice and optimizes sequences based on a composite score including CAI, GC content, and synonymous codon distribution. Evaluation of codon optimization tools focused in this study on two optimization tools that have been selected for evaluation:

- Codon Transformer[15] – An open-source tool, which allows modifications to its original code if necessary.
- Codon Optimization of Vector Builder[29] – A proprietary tool with a closed-source interface, where the user inputs the protein to be optimized and specifies the target host organism.

A toy example we ran to compare these two tools is the expression of an insulin protein that the host organism is Homo sapiens[30], aiming to evaluate the differences between the optimized DNA sequences for *E. coli* produced by these two tools.

2.4. New Quality Assurance (QA) Open Source Software Tool for Codon Optimization

Our main contribution to the community is a new QA open source software tool, which has been developed for testing any synthetic DNA sequence that was produced by any codon optimizer application. During the research, tests and statistics were performed on various sequences. The tests can be performed by a script written in Python, consisting of two main parts, as drawn in Figure 2 and explained below:

a) Backend - This part includes the logic for the various tests, including:

- Manipulations on sequences.
- Software interface with optimization tools.
- Searching for promoter motif in different positions in sequences.
- Injection attempt of promoter motif in different positions in sequences.

b) Frontend – This part presents the product to the client in the form of a Server that opens and displays the results on the WEB based on Hyper Text Markup Language (HTML) and Cascading Style Sheets (CSS).

There are two ways to access the Web application:

- Users – Running the application via an '.exe' file.
- Developers – Running the application via Python.

The open source described in the SI implements an application that can be accessed using the two methods described above.

Since it is an open source and we envision a design process that will continue as a collaborative effort, it is important to highlight the design principles. As shown in Figure 2, the application is designed with the following emphasis:

- Modular structure and division between backend and frontend.
- For each option available on the home page of the Web, a separate Python script was written to enable a proper development and debugging process.
- For convenience, the developer who wants to run the application will also be required to run only one script - "app.py", which is responsible for the integration and combination of all the scripts written in the project.

After running the application, you can access the local host URL on the Web: http://127.0.0.1:5000/ . At this point, the application home page opens, which describes all existing functions that constitute the research results. Figure 3 shows the home page that will be opened when there is a login to the application. The user can compare DNA sequences, can search motifs in the full genome or in specific sequences. In addition, a promoter can be injected into a sequence of choice. The advanced option is generating a test file for such injection.

We focus below on three main functions that can be considered the core components of the algorithm. The first one is motif detection. The second one is about silent motif injections. The third one is a function to secure your lab from unintended or malicious injections.

 doi:10.20944/preprints202602.1456.v1

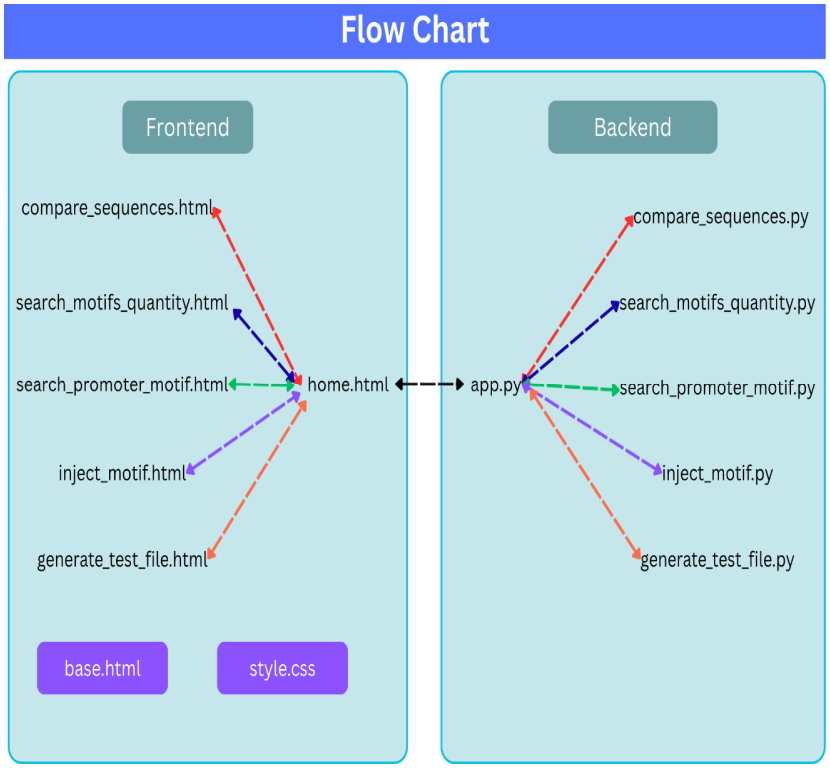

Figure 2. Software structure of the application. On the right side of the figure, we can see the Backend part and its Phyton scripts. On the left side, we see the Frontend part with its functions.

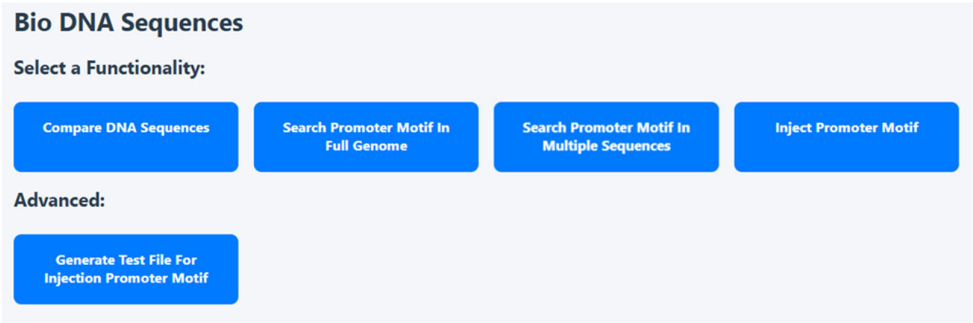

Figure 3. Current homepage of the web application: Four basic functionalities and a single advanced option. .

2.4.1. Antisense Promoter Motif Detection Algorithm

We use the motif detection function for two main purposes. Firstly, explore natural DNA sequences. It enables us to see if there is a natural evolutionary pressure against certain sequences that should be avoided for any reason. Secondly, scan optimized sequences and detect problematic motifs like anti-sense promoter. It enables us to test various optimization tools, prevent the usage of problematic sequences, and predict their impact in advance. As an example, we scanned for the 'ATTATA' motif using a sliding 6-mer window. Motif frequencies were calculated per sequence and across the genome to establish baseline occurrence. Our SW is available with some high-level description in the SI part.

2.4.2. Motif injection algorithm: Silent injection probability

 doi:10.20944/preprints202602.1456.v1

This tool tests the option of injecting a promoter motif into DNA sequences without changing the original protein sequence. The output refers both to the percentage of success and to the success for each specific sequence. It analyzes in which sequences the injection attempt was successful, and in which cases and locations this happened for those sequences. Injection simulations were conducted using an algorithm that iteratively substitutes synonymous codons in a target window to produce a reverse complement matching the target motif. We evaluated all possible codon combinations that encode the same amino acid chain while forming 'ATTATA' on the antisense strand.

2.4.3. Motif Defense Module: A software Tool to Protect Your Lab

To detect potential injections, we implemented a screening function that analyzes whether synonymous substitutions could feasibly produce antisense motifs. This tool flags sequences at risk and estimates the number of substitutions required for successful injections. Based on the software provided, we can also suggest alternative sequences in the case of motif injection.

3. Results

We present the results of our algorithms for large-scale data we found on the website. These results reveal insights regarding natural sequences as well as demonstrate their efficiency in the case of synthetic sequences.

3.1. Natural Occurrence of Antisense Promoter Motifs and Injection Feasibility in Motif-Free Sequences

The first test focused on the natural occurrence of antisense promoters. It enables to identify for each motif the evolutionary pressure against each motif and can lead to a better understanding of codon bias in various organisms. Among the 484,741 coding sequences analyzed, 23,057 (4.76%) contained the reverse-complement motif 'ATTATA' (Table 1). These sequences were excluded from subsequent injection tests. The motif's rarity supports the hypothesis that antisense promoters are naturally avoided in coding DNA. All other motifs in the table were detected much more often and very close to the theoretical random calculation. Please note that the motif ‘ATACCT’ can also lead to ‘-10’ motif, which is aligned with its low probability.

In the remaining 461,684 motif-free sequences, silent motif injection was achievable in 356,797 cases (77.28%) through synonymous substitutions. Three simple examples of such injections for SW validation are shown in Table 2.

Table 1. Scanning 484,741 proteins (546,462,945bp) for 6-mer motifs. The -10 antisense motif found in much lower probability than randomly generated motifs. One of the random motifs (‘ATACCT’) could also serve as a -10 antisense motif.

The Sequence	Probability by nucleotides ($P \sim 10^{-4}$)	Probability by sequences ($P \sim 10^{-1}$)
ATTATA (motif)	$4.64 \cdot 10^{-5}$	$4.76 \cdot 10^{-2}$
GCCATC	$3.02 \cdot 10^{-4}$	$2.73 \cdot 10^{-1}$
CATTGC	$4.27 \cdot 10^{-4}$	$3.54 \cdot 10^{-1}$
TGGCAT	$1.22 \cdot 10^{-4}$	$1.26 \cdot 10^{-1}$
ATACCT	$8.92 \cdot 10^{-5}$	$8.85 \cdot 10^{-2}$
CGATCG	$1.85 \cdot 10^{-4}$	$1.67 \cdot 10^{-1}$

 doi:10.20944/preprints202602.1456.v1

Table 2. Example of successful injections of promoter motif in the complementary strand. The DNA encodes a short peptide used as an SW validation case. All three insertions are found as expected. First one requires only one change in the 3rd nucleotide of the first codon 'AAC' ➔ 'AAT'. The other two insertion cases also generate the -10 motif in the antisense strand. The second case requires two changes, and the third case only one synonymous change. Generated test sequence (SW validation).

Predicted DNA	Protein Sequence	Expected Case
AACTATATGATCATTGCTTTATAT	NYMIIALY	[(case 3: start in codon 1), (case 1: start in codon 4), (case 2: start in codon 6)]

Injections

Original DNA	Modified DNA	# Insertions	Changes	Existing Motif	Cases
AACTATATGATCATTGCTTTATAT	AATTATATGATTATAGCATTATAT	3	[('Insertion_1', 2, 'C', 'T'), ('Insertion_2', 11, 'C', 'T'), ('Insertion_2', 14, 'T', 'A'), ('Insertion_3', 17, 'T', 'A')]	FALSE	[3, 1, 2]

3.2. Comparison with Random k-Mer Frequencies

We calculate the probability of each injection using the following procedure, demonstrated here for a specific 6-mer motif. To determine the probability of finding the anti-sense promoter motif, 'ATTATA', within a randomly generated sequence composed of amino acid codons. The probability calculations are based on the following naïve assumptions:

- All amino acids appear with equal probability in the sequence.
- No specific codon usage bias.

To find the probability of obtaining the motif randomly, it is necessary to determine how this sequence can be formed across different codon alignments. There are three frame shifts to be analyzed:

- Case 1: Two Complete Codons: 'ATT' – 'ATA'
- Case 2: Across Three Codons: '_ _ A' – 'TTA' – 'TA _'
- Case 3: Across Three Codons: '_ AT' – 'TAT' – 'A _ _'

For each of these cases, it is required to calculate the probability based on amino acid distributions.

For case 1:

'ATT' is a codon for Ile (I). 'ATA' is also a codon for Isoleucine Ile (I). Because according to codon usage bias Ile has three codons (ATT, ATC, ATA), so the probability of selecting 'ATT' or 'ATA' is 1/3 each. Since we assume that the probability of getting each amino acid is equal, then the probability for each amino acid is 1/20. Therefore, the overall probability in this case will be calculated as follows:

$$(1) \quad P_{Case1} = P(ATT, ATA) = P(ATT) \cdot P(ATA) = \left(\frac{1}{20} \cdot \frac{1}{3}\right) \cdot \left(\frac{1}{20} \cdot \frac{1}{3}\right) = 2.78 \cdot 10^{-4}$$

For case 2:

The amino acids with a codon ending with 'A' are listed in Table 3. We can now calculate the probability of a codon ending with 'A':

$$(2) \quad P(A_{end}) = \frac{1}{20} \cdot \left(\frac{1}{4} + \frac{1}{2} + \frac{1}{4} + \frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3} + \frac{1}{6}\right) = \frac{47}{240}$$

Next probability to calculate is for 'TTA', which is a codon for Leu (L). Because according to codon usage bias Leu (L) has 6 codons, the probability of selecting 'TTA' is:

$$(3) \quad P(TTA) = \frac{1}{20} \cdot \frac{1}{6} = \frac{1}{120}$$

 doi:10.20944/preprints202602.1456.v1

The amino acid that starts with 'TA' is only Tyr (Y). Since we assume that the probability of getting each amino acid is equal, then the probability will be:

(4) $P(TA_{start}) = \frac{1}{20}$

Therefore, the overall probability in this case would be calculated as follows by considering the three probabilities found above:

(5) $P_{case2} = P(A, TTA, TA) = P(A) \cdot P(TTA) \cdot P(TA) = \frac{47}{240} \cdot \frac{1}{120} \cdot \frac{1}{20} = 1.63 \cdot 10^{-4}$

For case 3:

The amino acids that end with 'AT' are listed in Table 4. Based on this table we calculate the probability:

(6) $P(AT_{end}) = \frac{1}{20} \cdot \left(\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2}\right) = \frac{1}{10}$

The codon 'TAT' is for Tyr (Y). Since Leu (L) has 2 codons, the probability of selecting TAT is:

(7) $P(TAT) = \frac{1}{20} \cdot \frac{1}{2} = \frac{1}{40}$

Lastly, the amino acids that start with 'A' are shown in Table 5. It yields the following calculation:

(8) $P(A_{start}) = \frac{1}{20} \cdot \left(\frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \frac{1}{4} + \frac{1}{3}\right) = \frac{31}{240}$

Therefore, the overall probability in this case would be calculated as follows by considering the three probabilities found above:

(9) $P_{frameshift[0]} = P(AT, TAT, A) = P(AT) \cdot P(TAT) \cdot P(A) = \frac{1}{10} \cdot \frac{1}{40} \cdot \frac{31}{240} = 3.23 \cdot 10^{-4}$

The overall probability would be to test the union of the three cases:

(1) $P_{total} = \frac{2.77 \cdot 10^{-4} + 1.63 \cdot 10^{-4} + 3.23 \cdot 10^{-4}}{3} = 2.54 \cdot 10^{-4}$

For comparison, the theoretical probability of randomly finding a six-nucleotide sequence assuming an equal distribution of nucleotides is:

(11) $P_{random} = \left(\frac{1}{4}\right)^6 = 2.44 \cdot 10^{-4}$

We can now revisit the results of Table 1 for various 6-mers. Our control analysis using random 6-mers revealed that sequences like 'GCCATC' and 'TGGCAT' occurred at frequencies up to 3× higher than 'ATTATA' and very close to the random calculation. It confirms the underrepresentation of certain motifs and likely selective pressure against antisense promoter formation. The only random 6-mer that is relatively close to 'ATTATA' is 'ATACCT', which has only 2 times higher probability. This specific sequence seems to create a sequence close enough to '-10' site promoter motif for *E.coli*.

Table 3. The amino acids that end with 'A'. The number of all optional codons for each amino acid is shown in the second column. The third column counts how many of them end with 'A'. The probability in the last column is the division of the previous two.

 doi:10.20944/preprints202602.1456.v1

Name Of Amino Acid	Optional Codons	Codons that end in A	Probability
Gly (G)	4	1	1/4
Glu (E)	2	1	1/2
Ala (A)	4	1	1/4
Val (V)	4	1	1/4
Arg (R)	6	2	1/3
Lys (K)	2	1	1/2
Thr (T)	4	1	1/4
Ile (I)	3	1	1/3
Gln (Q)	2	1	1/2
Pro (P)	4	1	1/4
Leu (L)	6	2	1/3
Ser (S)	6	1	1/6

Table 4. The amino acids that end with 'AT'. The number of all optional codons for each amino acid is on the second column. The third column counts how many of them end with 'AT'. The probability in the last column is the division of the previous two.

Name Of Amino Acid	Optional Codons	Codons that end with AT	Probability
Asp (D)	2	1	1/2
Asn (N)	2	1	1/2
His (H)	2	1	1/2
Tyr (Y)	2	1	1/2

Table 5. The amino acids that start with 'A'. The number of all optional codons for each amino acid is on the second column. The third column counts how many of them start with 'A'. The probability in the last column is the division of the previous two.

Name Of Amino Acid	Optional Codons	Codons that start with A	Probability
Arg (R)	2	1	1/2
Ser (S)	2	1	1/2
Lys (K)	2	1	1/2
Asn (N)	2	1	1/2
Thr (T)	4	1	1/4
Ile (I)	3	1	1/3

4. Discussion

Our findings reveal a significant and underrecognized vulnerability in codon optimization pipelines. While these tools aim to improve gene expression by refining synonymous codon usage, they unintentionally enable the creation of functional motifs on the antisense strand, such as bacterial

 doi:10.20944/preprints202602.1456.v1

promoter-like sequences. The widespread feasibility of silently injecting 'ATTATA' across coding regions, with no effect on protein output, underscores this threat.

Biological implications include unintended transcriptional activity, interference with native gene regulation, and synthetic noise in engineered circuits. From a cyber-biosecurity perspective, this opens the door to adversarial attacks where malicious motifs are embedded into synthetic constructs — a form of sequence-level 'backdoor' akin to steganography in software systems.

Current design tools often ignore complementary strand effects. A defense module that can be easily implemented based on our detection algorithm should be incorporated into bio foundry pipelines. Future work can easily expand to other motif types (e.g., riboswitches, terminators), eukaryotic systems, and automated prevention mechanisms using AI-guided codon selection.

We envision the open source code provided here becoming the basis of a much bigger set of QA tools that will be used by every lab before ordering DNA to decrease the probability of experimental failures. In addition, this set of tools can assist biologist understanding evolutionary pressure against certain motifs, leading to a better understanding of the genetic code.

5. Conclusions and Future Research Directions

Codon optimization has become indispensable in synthetic biology, yet our study highlights that it also presents hidden risks. Antisense promoter motifs such as 'ATTATA' can be introduced without altering protein sequence in over 77% of motif-free sequences. We propose a framework for motif detection, injection simulation, and defense screening strategy. Integrating such tools into standard codon optimization workflows is essential to safeguard biotechnology and basic biological research from inadvertent or malicious motif insertion.

Future research directions based on the open source code provided include the following:

1. Dive deeper into understanding the genetic code and various motifs with probabilities very different than random (both much higher and much lower).
2. Developing more QA tools to minimize the potential damage of synthetic DNA sequences that might encode surprises for biologists using these tools.
3. Identify other bio-informatics tools that should be examined within this framework to ensure their high quality and minimize their damage potential.

Supplementary Materials: The supporting information is described and can be downloaded. Details in the SI document.

Author Contributions: Conceptualization, Y.D.; methodology, all; software, E.C. and R.G.; validation, all; formal analysis, all; investigation, all; resources, all.; data curation, all; writing—original draft preparation, all; writing—review and editing, all; visualization, all; supervision, Y.D.; project administration, all; funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Israel Science Foundation (ISF), HIT research internal funds, the Ariel University–HIT Joint Research Grant, the Ministry of Innovation, Science and Technology (MOST), the Planning and Budgeting Committee (PBC/VATAT), the Directorate of Defense R&D (MAFAT), and Tevuna.

Data Availability Statement: The data supporting the findings of this study include publicly available protein and gene annotation data obtained from the UniProt database, as well as genomic sequences retrieved from the NCBI database. In addition, synthetic DNA sequences were generated programmatically for validation and testing purposes. No sensitive or personal data was used. All scripts and algorithms developed for data generation, motif insertion, and motif analysis were created by the authors as part of the research and are available from the corresponding author upon reasonable request. No additional datasets were generated.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

 doi:10.20944/preprints202602.1456.v1

References

1. Carlson R. 2009. The changing economics of DNA synthesis. Nat Biotechnol 27:1091–1094.
2. Cameron DE, Bashor CJ, Collins JJ. 2014. A brief history of synthetic biology. Nat Rev Microbiol 12:381–390.
3. Elgabry M, Nesbeth D, Johnson SD. 2020. A Systematic Review of the Criminogenic Potential of Synthetic Biology and Routes to Future Crime Prevention. Front Bioeng Biotechnol. Frontiers Media S.A. https://doi.org/10.3389/fbioe.2020.571672.
4. Watts A, Sankaranarayanan S, Watts A, Raipuria RK. 2021. Optimizing protein expression in heterologous system: Strategies and tools. Meta Gene 29:100899.
5. Condon A, Thachuk C. 2012. Efficient codon optimization with motif engineering. Journal of Discrete Algorithms 16:104–112.
6. Angov E. 2011. Codon usage: Nature's roadmap to expression and folding of proteins. Biotechnol J 6:650–659.
7. Tuller T, Carmi A, Vestsigian K, Navon S, Dorfan Y, Zaborske J, Pan T, Dahan O, Furman I, Pilpel Y. 2010. An Evolutionarily Conserved Mechanism for Controlling the Efficiency of Protein Translation. Cell 141:344–354.
8. Goodman DB, Church GM, Kosuri S. 2013. Causes and Effects of N-Terminal Codon Bias in Bacterial Genes. Science (1979) 342:475–479.
9. Quax TEF, Claassens NJ, Söll D, van der Oost J. 2015. Codon Bias as a Means to Fine-Tune Gene Expression. Mol Cell 59:149–161.
10. Plotkin JB, Kudla G. 2011. Synonymous but not the same: the causes and consequences of codon bias. Nat Rev Genet 12:32–42.
11. Masłowska-Górnicz A, van den Bosch MRM, Saccenti E, Suarez-Diez M. 2022. A large-scale analysis of codon usage bias in 4868 bacterial genomes shows association of codon adaptation index with GC content, protein functional domains and bacterial phenotypes. Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms 1865:194826.
12. Mo O, Zhang Z, Cheng X, Zhu L, Zhang K, Zhang N, Li J, Li H, Fan S, Li X, Hao P. 2025. mRNAdesigner: an integrated web server for optimizing mRNA design and protein translation in eukaryotes. Nucleic Acids Res 53:W415–W426.
13. Han X, Shao X, Liu S, Shi Z, Huang R, Chu H, Zhang H, Wang R, Li H, Liao X, Cheng J, Jiang H. 2025. DeepCodon: A deep learning codon-optimization model to enhance protein expression. BioDesign Research 7:100042.
14. Karaşan O, Şen A, Tiryaki B, Cicek AE. 2022. A unifying network modeling approach for codon optimization. Bioinformatics 38:3935–3941.
15. Fallahpour A, Gureghian V, Filion GJ, Lindner AB, Pandi A. 2025. CodonTransformer: a multispecies codon optimizer using context-aware neural networks. Nat Commun 16:3205.
16. Welch M, Govindarajan S, Ness JE, Villalobos A, Gurney A, Minshull J, Gustafsson C. 2009. Design Parameters to Control Synthetic Gene Expression in Escherichia coli. PLoS One 4:e7002.
17. Browning DF, Busby SJW. 2004. The regulation of bacterial transcription initiation. Nat Rev Microbiol 2:57–65.
18. Zhou X, Feng R, Ding N, Cao W, Liu Y, Zhou S, Deng Y. 2025. Deep learning guided programmable design of Escherichia coli core promoters from sequence architecture to strength control. Nucleic Acids Res 53:gkaf863.
19. E DJ, M DA, J PM, T WJ. 2010. Widespread Antisense Transcription in Escherichia coli. mBio 1:10.1128/mbio.00024-10.
20. Anjum N, Alshahrani H, Shaikh A, Mahreen-Ul-Hassan, Kiran M, Raz S, Alam A. 2025. Cyber-Biosecurity Challenges in Next-Generation Sequencing: A Comprehensive Analysis of Emerging Threat Vectors. IEEE Access 13:52006–52035.
21. Peccoud J, Gallegos JE, Murch R, Buchholz WG, Raman S. 2018. Cyberbiosecurity: From Naive Trust to Risk Awareness. Trends Biotechnol 36:4–7.

 doi:10.20944/preprints202602.1456.v1

22. Elgabry M, Nesbeth D, Johnson SD. 2020. A Systematic Review of the Criminogenic Potential of Synthetic Biology and Routes to Future Crime Prevention. Front Bioeng Biotechnol Volume 8-2020.
23. Puzis R, Farbiash D, Brodt O, Elovici Y, Greenbaum D. 2020. Increased cyber-biosecurity for DNA synthesis. Nat Biotechnol 38:1379–1381.
24. Murch RS, So WK, Buchholz WG, Raman S, Peccoud J. 2018. Cyberbiosecurity: An Emerging New Discipline to Help Safeguard the Bioeconomy. Front Bioeng Biotechnol Volume 6-2018.
25. Doricchi A, Platnich CM, Gimpel A, Horn F, Earle M, Lanzavecchia G, Cortajarena AL, Liz-Marzán LM, Liu N, Heckel R, Grass RN, Krahne R, Keyser UF, Garoli D. 2022. Emerging Approaches to DNA Data Storage: Challenges and Prospects. ACS Nano 16:17552–17571.
26. Ibaisi TAL, Kuhn S, Kaiiali M, Kazim M. 2023. Network Intrusion Detection Based on Amino Acid Sequence Structure Using Machine Learning. Electronics (Basel) 12.
27. DB source: Database. https://www.uniprot.org/uniprotkb?query=Escherichia+coli+%28strain+K12%29&facets=reviewed%3Atrue%2Cmodel_organism%3A83333. Retrieved 6 February 2026.
28. Chakravarty PR. 2023. What is Codon Bias: Implications in Gene Cloning.
29. VectorBuilder. 2025. Codon Optimization Tool. https://en.vectorbuilder.com/tool/codon-optimization.html.
30. U. Consortium. 2025. UniPort entry P01308. https://rest.uniprot.org/uniprotkb/P01308.fasta.