Do sequence obfuscation technologies present a biothreat?

Abstract

DNA synthesis is a large and growing industry. To prevent the illegitimate synthesis of hazardous pathogens, companies screen their sequence orders. Recent techniques have been shown to obfuscate the sequences of synthetic genetic circuits. Could such techniques enable the evasion of screening? I consider how two obfuscation techniques (scrambling and camouflage) might evade three families of screening approaches (basic bioinformatics, random adversarial thresholds, and functional genomics). Using theoretical models and exploratory simulations I find that the scrambling of pathogen genomes is generally infeasible with current technology, but conjecture that linear advances in the technology might enable evasion of random adversarial thresholds. Camouflage also appears generally infeasible for the obfuscation of hazardous genomes. I then assess the key drivers and likely trajectories for the future of sequence obfuscation, and point to several developments that might disrupt our security. They are: intellectual property in synthetic biology that is embedded in very short sequences; norms of common use of obfuscation; large advances in CRISPR ease and efficiency; obfuscation in situ; arbitrarily orthogonal recombination; seamless recombination; and diffusion of genetic recoding. In general, imaginative and collaborative red-teaming is encouraged.

Introduction

DNA synthesis is a rapidly growing industry that significantly helps biomedical research.[1] [2] Simultaneously, these services lower technical barriers to the procurement and subsequent misuse of highly dangerous pathogens.[3] [4] [5] To prevent bad actors accessing the genes of dangerous pathogens via these companies, screening approaches were outlined by the U.S. Department of Health and Human Services[6] and put into practice by many synthesis companies, most notably as part of the International Gene Synthesis Consortium.[7] These generally work by screening customers against specific lists (e.g. the State Department's Debarred List, or Specially Designated Nationals) while also comparing the requested genetic sequence against the Regulated Pathogen Database and other reference databases.[8] These comparisons usually involve the use of basic bioinformatics tools, but more sophisticated approaches are being developed, such as 'random adversarial thresholds' by SecureDNA[9] and 'functional genomics' by the Intelligence Advanced Research Projects Activity (IARPA)[10]. Last year, the Nuclear Threat Initiative and World Economic Forum called for a common, global model for DNA screening.[11] More recently, California passed the first law requiring synthesis companies to screen their customers and orders.[12] Such efforts are promising and needed, and motivate efforts to identify and resolve vulnerabilities in existing screening approaches before they are cemented into less nimble institutions such as laws or global agreements.

[1] Carter and Friedman 2015
[2] Carter and DiEuliis 2019
[3] DiEuliis, Carter and Gronvall 2017
[4] Carter and Friedman 2015
[5] Kobokovich et al. 2019
[6] Department of Health and Human Services, 2010
[7] International Gene Synthesis Consortium; 2017
[8] Ibid.
[9] Gretton et al., n.d.
[10] King, 2021
[11] Nuclear Threat Initiative and World Economic Forum. 2020.
[12] West and Gronvall, 2021

One unexplored vulnerability concerns the increasing sophistication and accessibility of sequence obfuscation techniques.[13] The development of such techniques is motivated legitimately by the need to protect the intellectual property instantiated in the genomes of synthetic biological products. To this end, scientists have shown in-vivo proof-of-concept techniques for hiding the structure and functions of synthetic genetic circuits.[14] Using multiple, nested sets of site-specific recombinases to excise or reverse parts of the genome, a user can 'scramble' the genome, rendering its true structure and function opaque without the 'decryption key' of knowing what specific recombinases to apply in which order.[15] Similarly, users can use steganographic techniques to 'camouflage' the true synthetic circuit amongst false dummy-components that are then subtracted away via a specific CRISPR-Cas9 'decryption key'. These methods allow a user to obscure the end-function and structure of a DNA sequence while it is being ordered, synthesized, stored, and transported. New obfuscation techniques like genetic recoding are on the horizon, and advances in existing obfuscation techniques could increase their threat potential.

This paper will first describe three major families of screening approaches: basic bioinformatics, random adversarial thresholds, and functional genomics. Then it will describe two current techniques to obfuscate sequences - scrambling and camouflage - and assess whether such methods could feasibly be used to obfuscate a hazardous sequence from each screening approach. Then the paper will consider the future of obfuscation as a biosecurity threat, including what changes to the threat landscape should be concerning to those trying to reduce global

[13] Purcell et al. 2018
[14] Ibid
[15] Ibid

catastrophic biological risks, and what foreseeable forces could drive those changes. Finally, the paper will draw implications for the screening problem in general, and explain the management of information hazards involved in the project.

Screening approaches

Basic bioinformatics

Comparing a query sequence to a database is a basic procedure in bioinformatics. Comparisons on the basis of similarity or homology can provide evidence of species identity, evolutionary history, mutations from a wild type, and resulting protein structure and function. One family of comparison methods is sequence alignment. Sequence alignment mostly works by looking for places where parts of the query sequence match sequences in the database.

Quantitative sequence alignment uses dynamic programming, which relies on breaking down and solving recursive sub-problems. Exact matches are often too specific to be very useful, so alignment algorithms also use a scoring matrix - a table of values corresponding to particular substitutions of nucleotides or amino-acids. Some algorithms also use gap penalties - negative values corresponding to the size of an insertion or deletion needed for an alignment. An example of dynamic programming for alignments is the Smith-Waterman algorithm, which is guaranteed to find the optimal alignment for a given scoring system.

However, it's often impracticably slow to go through the scoring algorithm for every window of the sequence against every window in a large database. So in practice alignment tools tend to be heuristics - ways for finding high scoring alignments quickly before subjecting them to a more rigorous algorithm. One such heuristic tool is BLAST, which stands for Basic Local Alignment Search Tool.[16]

[16] Altshul et al. 1990

Here is how BLAST works. BLAST first removes low-complexity regions - parts of a sequence that are only made up of a few elements, which therefore contain less useful information and cause more false positive matches. Then BLAST considers subsequences or words of a particular length *k* - called a *k*-mer, and lists all the *k*-mers in the query sequence. It then considers all possible *k*-mer permutations. For example, given 20 possible amino acids and a word-length of *k*, there are 20^k possible words. Comparing a query *k*-mer to each possible word, it scores the comparisons according to a scoring matrix.
High-scoring comparisons above some threshold score *T* are searched for in the database. When there is a high-scoring match in the database, this is used as a seed to extend the comparison forward and backward in the database sequence, until the cumulative score begins to decrease. Some versions of BLAST also incorporate gaps in the comparisons (e.g. BLAST2 or gapped BLAST) with accompanying penalties. Other versions improve accuracy by using position-specific scoring systems (e.g. PSI-BLAST).

Extended matches above the threshold are called high-scoring segment pairs (HSPs), and multiple HSPs in the same sequence can be joined together. For database entries with a HSP score above some threshold score *S*, a more rigorous algorithm (like the Smith-Waterman) is run between it and the query. This score threshold also translates into an e-value, which is the number of times that score could occur by chance. Finally, the algorithm can report all the database entries and their E-values for some user-defined e-value threshold.

The Guidance from the U.S. Department of Health and Human Services recommends local alignments for screening orders, and suggests BLAST as a popular option. They suggest a sliding 200 basepair (bp) window (that is, using as a query sequence the nucleotides from position 1-200, then 2-201, and so forth). They further suggest a ‘Best Match’ approach, where the highest scoring database entry for each 200bp

window is compared to the list of hazardous pathogens and toxins, often drawn from existing lists of dangerous pathogens, like the Federal Select Agent List or the Australia Group.[17]
Subsequent reports on this approach have found a 'hit'-rate of around 5% of orders requiring expert human follow-up. Overall, 0.7% of orders were a 'red hit', meaning both flagged by the 'Best Match' approach and having >80% homology to known pathogenic sequences.[18]

There are other tools in bioinformatics that can be used to compare a query sequence with a database. Hidden Markov models use stochastic unobserved states to represent how two sequences could be aligned. 'Alignment-free' methods rely on informational features of the query sequence other than strict sequence comparison, such as the frequencies of particular *k*-mers or nucleotides, k-mer fingerprint phylogeny trees, the number of spaced word matches, and the mutual information between sequences. These approaches don't appear to be generally used in initial screening, but may form part of the toolkit of human experts following up apparent hits.

Random adversarial thresholds approach[19] [20]

SecureDNA is developing an alternative approach to basic bioinformatics using 'random adversarial thresholds', aiming to improve specificity and automatability. SecureDNA chooses a number of short particular subsequences ('windows') of a hazard, which are usually 19 amino acids or 57 base pairs long. This length is intended to be so short that it would be very difficult to assemble a larger pathogenic sequence from parts that are smaller than 57 bp.

They then search for exact matches of the 57-mers in the DNA order. To counter an adversary using minor mutations to avoid exact matches, SecureDNA also creates a database of a significant number of 'functional variants', using a number of methods of variant predictors. To prevent false positives,

[17] Department of Health and Human Services, 2010
[18] Carter and Friedman 2015
[19] Gretton et al. N.D.
[20] Baum et al. N.D.

SecureDNA removes any computed functional variants that also occur in unrelated, non-hazardous sequences. The screening algorithm looks for exact matches between the query and this database of functional variants.

An intelligent adversary might try to introduce significant mutations in every 19 amino acids, and thus in every possible window. However, for each window w_i, the adversary's mutated sequence could result in one of three outcomes: it could be matched to the variant database, raising a red flag; it could evade detection with an accompanying fitness cost c_i; or it could evade detection with no fitness cost. To access a working hazardous pathogen, an adversary would need to limit the overall fitness cost beneath some threshold, which practically would require achieving the 'evasion with no fitness outcome' in a majority of windows.

SecureDNA defines a 'random adversarial threshold' R as the probability of detecting an adversary who has perfect knowledge of fitness costs but lacks knowledge as to which windows of a sequence are assessed and what variants of each window are in the database. Fitness costs can't be perfectly predicted, so the screening algorithm can't just list all minimally functional variants of one window (which could achieve $R = 1$). Instead, an 'optimal' screening approach could pick windows that are least amenable to substitution and include variants of this window that have the lowest fitness costs. Then it could proceed to the next least-amenable window, and so forth. In this way, the smallest number of variants introduce the greatest difficulty to the adversary.

SecureDNA ran a 'red-teaming' exercise to test the security of its approach. They used a bacteriophage as a harmless stand-in for a hazardous sequence. Across 4 proteins, 9 windows were chosen on the basis of their low number of possible 'neutral' substitutions. The defensive team computed 10^6 functional variants for each window using a basic substitution matrix and sampling method. The attacking team launched 21,000 attacks at each window.

Variants were defined as functional if their fitness was at least 0.05 that of the wild-type. This was chosen as a level that would reduce the fitness of the most infectious pathogen - measles - to a level where it would only just be able to spread in an unvaccinated population.

The detection of attacks varied significantly across windows. In the most defensible window, 92% of attacks were detected. Most other windows had detection rates of 40-50%. Because an adversary's order would conceivably have to evade all windows simultaneously, SecureDNA concluded their overall detection rate in the exercise to be 99.96%.

In the real world, adversaries wouldn't know which windows are defended, and so would have to introduce mutations every 19 amino acids to have confidence in evading detection. Furthermore, databases could be as large as 10^9 variants. Finally, defenses can be adaptive, reactively targeting window and variant selection in response to detected attempts, while also changing variant prediction methods and window selections to prevent foreknowledge of defensive measures.

Functional genomics

The third family of screening protocols we will consider is 'functional genomics'. IARPA is funding five teams in a program called Fun GCAT, which stands for 'Functional Genomics and Computational Assessment of Threats'.[21] For a given query sequence each team aims to analyse what organism it came from, what function it serves, and how dangerous it is. While each team's approach is different, they generally use a combination of specifically designed algorithms, machine learning, and specially annotated databases. Specific designs have not been published, except for SignatureScience's SeqScreen program, which is open source.

[21] King 2021

SeqScreen aims to functionally annotate short oligonucleotides for pathogenicity. It is an ensemble machine learning model trained on manually labelled sequence data. SeqScreen appeared to perform with high accuracy on quite short sequences, and may also be better able to classify novel pathogens, at least relative to approaches based on taxonomy databases.[22]

Current obfuscation techniques

Sequence obfuscation is a family of emerging techniques that aim to make synthetic circuit designs inscrutable to those who can read the circuit's sequence but who lack the private decryption key. Initially, we might expect some screening protocols to have difficulty identifying hazardous sequences that have been obfuscated.

To understand current obfuscation techniques, we have to consider their application to synthetic genetic circuits. What are synthetic genetic circuits? Synthetic genetic circuits are an abstract way of thinking about genetic design, inspired by the logical circuits in electrical and mechanical engineering.[23] Circuits in this sense occur in natural genomes, designed by natural selection and serving particular functions within the cell. Synthetic genetic circuits are 'devices' designed by biologists to take inputs, perform particular functions, and provide outputs, usually within a particular host cell, such as *E. coli*.

Circuit components are more primitive systems made up of interacting genes and transcription factors. Example components include switches, oscillators, Boolean logic gates, memory stores, filters, sensors, and actuators. Synthetic circuits usually depend on careful combinations of interacting components. Feasible near-term applications of synthetic circuits could be detection of a drug or toxin, a trigger for the

[22] Balaji et al. 2021
[23] Jusiak et al. 2015

release of a therapeutic agent, or the manufacturing of a biofuel. More complex circuits and functions appear to be both feasible and desirable.

The ‘topology’ of a circuit is its overall network - its set of specific components and how they link together. The topology of the circuit is potentially valuable intellectual property, but someone with access to the genetic sequence could potentially read off this topology. Published obfuscation techniques aim to disguise and obstruct the reconstruction of circuit topology from sequences. When current obfuscation techniques ‘encrypt’ the sequence, they also make them non-functional. As such, obfuscation is only intended to protect the sequence’s intellectual property in situations such as product storage, transit, or synthesis by an external provider. Whether obfuscation could be achieved while the product is in use is an interesting and speculative question, reserved for a later section**.**

There are two major techniques for obfuscation with demonstration in the literature. They are scrambling (or ‘encryption’) and camouflage (or ‘steganography’).

Scrambling with recombinases

Scrambling is a method of encrypting the design of synthetic gene circuits by rearranging parts of the sequence. A demonstrated form of scrambling is using site-specific recombinases. These recombinases are naturally occurring enzymes that recognise a particular section of DNA marked on each end by particular subsequences (‘sites’); the recombinase can then invert or excise that section, depending on the orientation of the sites. There are 11 different site-specific recombinases that are orthogonal - that is, that do not interfere with each other - though more could be discovered. These multiple sets of recombinases

can overlap, such that 'decrypting' the sequence (getting the sequence back to its original, functional topology) requires applying the right set of recombinases in the right order.[24]

Here is an outline of a scrambling process. For a given original sequence on paper, choose sections to be inverted or add decoy sections to be excised. For each such section add the sequences for the required recombinase recognition sites. Then, still on paper, invert or excise those sections to create a new version of the sequence. Again, pick parts to be inverted or excised, and repeat. This can be done iteratively. At the end, you will have a 'scrambled' sequence - a sequence for which it is difficult for others to recognise the original structures or patterns (e.g. circuit topology and function). When you want to reconstruct your original sequence, you can apply the right recombinases in reverse order, undoing your on-paper transformations, and obtaining your functional circuit.

Through this method you could send a DNA order that is scrambled such that the synthesizing company could print it, but would have difficulty interpreting the order's true design and function. It would also restrict access to the circuit design, where actors with access to the physical product in storage or transit would need to also have the recombinase 'decryption key'.

Purcell et al. (2018) notes that the permutations of orders from using 4 recombinases is:

$$\sum_{k=1}^{k=4} \frac{n!}{(n-k)!} = 64$$

where n = total number of recombinases used and k = number of recombinases used in a possible decryption. Due to redundancies in permutation orders, they find that when using one set of sites for each

[24] Purcell et al. 2018

recombinase the number of unique sequences that can arise from using *n* recombinases is usually slightly less than 2^n.

In Purcell et al.'s example of scrambling a genetic AND-gate using 4 recombinases, an attacker would have to permute 16 possible circuit designs, many of which are plausible set ups of the promoters and genes. Sophisticated designers could contrive to have incorrect circuits to look plausible, and for the correct circuit to look messy or unlikely.

Scrambling in this way is limited by the number, orthogonality, and efficiency of recombinases. Currently there are 11 orthogonal site-specific recombinases, though more could be discovered and improved over time. However, this pattern might be disrupted if scrambling were to be combined with assembly methods. This would allow separate pieces of the overall sequence to be ordered and scrambled independently, which would allow the re-use of recombinases. Then the pieces could be stitched together. While there is added difficulty and cost for each assembly step, each separation essentially multiplies the number of orthogonal recombinases available, so it may be worth it in some use cases.

Our central question is: could scrambling obfuscate hazardous sequences from screening protocols, and thereby present a biothreat?

Scrambling to evade basic bioinformatics

A baseline for screening sophistication is offered by basic bioinformatics in general and BLAST in particular. BLAST is widely known, accessible, and free. To assess if scrambling could evade BLAST, we will first consider theoretical predictions, and then empirically test those predictions using a simplified model.

Recall BLAST's variables, where *k* is the length of a subsequence of the query, and *T* is a cut-off for high-scoring segment pairs to seed extended comparisons that are then subjected to a provably optimal alignment algorithm. Theoretically, to prevent being flagged by a BLAST-based screening protocol, a scrambling process would need to interrupt every *k*-mer that scores above a threshold *T* and links to a hazard sequence, for each 200bp window. How many interruptions are required?

We can make simplifying assumptions to model this problem. First, assume the attacker is 'scrupulous', and wants to interrupt every possible *k*-mer that could score over *T*. Some sections of a hazardous sequence might appear innocuous and so not require much (or any) scrambling, such as housekeeping genes. However, these sections are unlikely to be a majority of the sequence, so the effect on scrambling requirements is small. Furthermore, ambiguous subsequences can still register as hits - they are a common cause of false positives. An attacker who wanted to have high confidence of evading detection would want to scramble these ambiguous sections too.

Second, the use of recombinases requires the insertion of recombinase recognition sites. The use of recombinase-based excisions can restore inactive gene segments into a functional gene, suggesting that in such cases sites do not subsequently interfere with coding function.[25] However, Purcell et al. 's demonstration of recombinase scrambling consistently placed sites outside the regions of genes and transcription factors, and studies of recombinase sites in mouse genetics found that they disrupted coding function half of the time.[26] As such, practical scrambling of a pathogen genome would likely have to avoid placing sites within coding regions. This greatly increases the difficulty of obfuscation design, and greatly increases the likelihood of leaving intact *k*-mers that could be detected by screening. To make the

[25] Prorocic & Stark, 2013
[26] Goodwin et al. 2019

infeasibility case stronger, we will make the generous assumption of a scrambling method that is 'seamless' - that is, that leaves no sites remaining in the unscrambled sequence.

With these assumptions in mind, we can take a sequence (genome) of length *L,* and a detection method that relies on intact subsequences of length *k* in either direction. What is the minimum number of recombinase functions required to ensure that no original *k*-mer in *L* is intact?

Recombinase functions include inversions and excisions. Excisions are not an efficient use of recombinase functions. You can introduce a decoy element to be excised, but this would only break up one *k*-mer. This simplifies the problem to finding the minimum number of inversions required to disrupt every *k*-mer. An inversion reverses the order of a designated subsequence of the genome. An inversion could break up at most 2 k-mers - by their end points being inside separate k-mers. *k*-mers that are entirely inside an inversion are not interrupted - as the detection method considers sequences in both directions.

Given these features, the most efficient use of recombinase functions is to use inversions placed just before the end of a complete *k*-mer, at *(k*-1*)*. As you proceed through the sequence from the (*k*-1) position this pattern repeats, with an inversion needed after a further (*k*-1) positions. This thus proceeds by *R* steps of *(k-1)* length, where *R* is the number of inversions required. At the halfway point of the genome, the corresponding end-sites of the inverted subsequences meet. At this point, you need *R(k-1)* to be at least within half a step-length of the halfway point, which would completely prevent intact *k*-mers:

$$R(k-1) \geq \frac{L}{2} - \frac{(k-1)}{2}$$

Rearranging:

$$R \geq \frac{L}{2(k-1)} - \frac{1}{2}$$ (Eq. 1)

Using the assumptions of scrupulousness, seamless scrambling, and maximum efficiency, we arrive at a theoretical estimate of the minimum number of recombinase steps required to evade BLAST. We can input parameters favourable to the attacker to find specific numbers.

We can take the length of the sequence to be scrambled $L = 1700$, the smallest RNA virus genome. Take $k = 11$, the default word-size of BLAST. Given these values, the number of required inversions $R \geq 84.5$.

This means that 85 separate recombination events must be performed on the sequence to retain the original function. This is technically infeasible. First, there are only 11 orthogonal recombinases. Second, assuming 90% efficiency for every recombinase, the collective efficiency of the process would be $0.90^{85} = 0.013\%$.

Using the default word size of BLAST for our k value may be misleading - perhaps some score thresholds may only be breached by high-scoring segment pairs using longer word seeds. To find functional values of *k*, exploratory empirical testing was performed using a Python-scripted program that introduced a user-defined number of inversions with quasi-random positions and lengths. Samples of varying sequence length were randomly selected from several different viruses, and subjected to BLAST analysis after an increasing number of inversions. Using this method we can observe the number of inversions required to evade BLAST identification for that sample, or *R*. We already take *L* to be the shortest genome, or 1700bp. And we can rearrange Equation 1 for *k*:

$$k \geq \frac{L}{2R+1} + 1$$

Putting it all together, for sequence samples that were 200bp long (the size of the screening window recommended by the IGSC) testing suggests that the functional *k*-value was around 9 bp. Functional *k*-values grew shorter with longer samples, and there was significant variance. More systematic testing could be performed to achieve more reliable values. For our purposes, we can make our *k*-value more reliable by considering it to be between 5 and 13bp long.

Returning to equation 1 and using $k = 5$ and the same 1700bp genome, we find the minimum number of inversions is $R \geq 212$. Using $k = 9$, $R \geq 105.75$. Using $k = 13$, $R \geq 70.3$. All three values suggest an unreasonable amount of recombination events are required to obfuscate the shortest genome.

We can make this case more robust by calculating the minimum length *k*-mer that can be consistently disrupted, given a user implements all 11 currently known orthogonal recombinase systems on the shortest genome. Or in algebraic terms: what is *k* when $L = 1700$ and $R \leq 11$?
We find $k \geq 74.9$.

This means that using all available recombinases in the most efficient way on the shortest genome would still leave intact *k*-mers of up to 75 bp in length. Intact *k*-mers of this length are readily amenable to being picked up as high-scoring segment pairs in BLAST, and so are very unlikely to evade this basic bioinformatic screen.

Furthermore, the most efficient use of recombinases (inversions at every (*k-1*) position) is unlikely to be an effective encryption method, given it's predictability. Improving the encryption would involve varying

the position of inversions and using excisions, both of which increase the number of recombination events required.

A countervailing force noted in the preceding section is the potential for genomes to be ordered separately and stitched together. This combination process was used by necessity in the synthesis of horsepox, where the 212kbp genome was ordered as separate overlapping 30kbp segments, which were then assembled together.[27] Separate genome fragments could be scrambled independently, as recombinase recognition sites could be used repeatedly without interference.

With respect to our model, splitting the relevant genome into *N* separate parts approximately multiplies the set of independent recombinases by *N*, while the total number of inversion transformations required remains approximately the same.

While this is a potential worry, scrambling still appears infeasible. For the shortest genome, the highest k-value of 13, and using all 11 recombinases, an attacker would still need to split the genome up into more than 6 pieces. The total number of recombination events would remain 71, so a generous estimate of efficiency would equal 0.90^{71} or 0.05%, and this would be further multiplied by the imperfect efficiency of the assembly process. While hard to assess objectively, the difficulty of such an evasion becomes comparable to the difficulty of synthesizing whole constructs on their own, nucleotide by nucleotide, avoiding screening altogether.

Relaxing our assumption of seamless recombination strengthens the infeasibility case for a scrambling attack against BLAST. Recombination recognition sites likely interfere with coding regions, meaning the attacker cannot be confident of reconstructing a functional sequence without leaving large, telltale subsequences intact. Recombination recognition sites also take up space - simple sites can be 30bp, while

[27] Noyce, Lederman, & Evans 2018

complex ones can be up to 200bp.[28] Even assuming every site is 30bp, the minimum number of recombination events would require:

(2 sites per recombinase * 30bp * 71 recombinations) = 4260bp.

Which is 2.5 times larger than the smallest viral genome itself.

Finally, while BLAST may act as an initial, automated screen, suspicious hits are followed up by human experts. This protocol brings a much larger array of bioinformatics tools to bear on analysing the sequence. As discussed in the preceding section, these tools exploit a number of features of a sequence beyond word alignment. Because these features use statistical measures of more granular features of the sequence, the effective value of *k* in our model is reduced, such that recombination-based disruption of those *k*-mers becomes near impossible.

Targeted follow-up could also identify recombinase recognition sites and permute some inversions or excisions. The number of possible permutations may make it intractable to retain the original sequence. For screening purposes however, it would suffice to try some permutations and reconstruct pieces of the original sequence that are longer than *k* and form a high-scoring alignment with a hazard.

Scrambling to evade random adversarial thresholds

Scrambling appears insufficient to obfuscate hazardous sequences from the tools of basic bioinformatics. Because of this, one might think that more sophisticated screening protocols would naturally defeat scrambling attacks, but this assumption is too quick. Screening protocols trade off a number of variables in their design, such that improvements in specificity or automatability may open new vulnerabilities. We

[28] Prorocic & Stark, 2013

can shed light on this by considering scrambling attacks on a random adversarial threshold (RAT) approach.

RAT predicts sets of functional variants of hazard sequences to counter the potential threat of adversaries using subtle mutations to evade exact-matching screening.[29] These functional variants are generated using a variety of protein prediction software, which also generate their predicted fitness costs. Importantly, these variants are predicted in a functional form - as they would exist in a working organism. A theoretical vulnerability occurs because scrambling could allow an attacker to order 'variants' of a fragment that would not appear to be functional, and would not exist in the RAT database. The scrambled variant would only regain function when it is unscrambled by the appropriate recombinase set, but after that point could function normally as the hazard sequence.

In a general sense, this problem arises because the variant generating algorithm assumes a particular test for functionality, and scrambling offers a form of functionality that exists outside that test - that is, the functionality that can be gained through alterations post-synthesis.

Evading RAT with scrambling is not trivial, however. Scrambling 'variants' would need to be obtained for every possible window of 57bp. We can return to Equation 1 from the previous section, and use a k-value of 57. For the shortest genome of 1700bp, the number of inversions required would be $R = 15$, which would still be infeasible given current recombinase technology. Splitting the genome into 2 pieces would still require a minimum of 15 recombination events, but would then only require 8 independent recombinases, which is currently feasible. Evading RAT with scrambling still appears difficult and unlikely even with generous assumptions, but does appear to be theoretically possible.

[29] Gretton et al., N.D.

If this vulnerability is genuine, further scrutiny of this potential problem might be warranted, which could lead to either reassurance or adaptation to the attack type. It is fortunate that the RAT approach seems readily amenable to respond and close the vulnerability. Scrambled variants could be added to the variant databases, with variant predictors modified to include the possibility of scrambling. However, a key question is whether the space of variants made possible by scrambling is too large to be adequately sampled and defended by the RAT database.

A SecureDNA manuscript on the RAT approach describes a red-team exercise where libraries of 10^6 variants per fragment window were sufficient to defeat 99.96% of attacks.[30] It also noted that libraries of 10^9 variants per hazard were affordable. If SecureDNA were to use 100 windows per hazard, there would be 10^7 variants available per window. If the number of windows was kept the same as the demonstration (at 9), there would be 1.11×10^8 variants per window.

Assume a scrambling attack is both scrupulous and seamless, as in the basic bioinformatics model. First, an attacker would need to interrupt every 57-mer, requiring 15 recombination events, as calculated above. We can break down the combinatorics of the problem by considering where the attacker might position the start of an inverted subsequence. Our model of 15 recombination events assumed the efficiency-maximising position of (k-1), but this would be predictable and easily detected. Instead, the attacker could position the start of an inversion in any position within the subsequence 1…(k-1). For simplicity let's assume that the attacker's conflicting aims for unpredictability and efficiency converge such that the average position of the start of an inversion is roughly the middle of such a choice-set, or $\frac{(k-1)}{2}$

This would lead to an effective k-value of $\frac{57}{2}$ or 28.5.

With this new k-value and a genome of 1700bp, the number of inversions R = 30.4, or 31 inversions.

[30] Ibid.

The number of unique sequences that can be generated by using R recombinases is generally slightly, but not significantly, less than 2^R.[31] Given the use of 31 inversions to scramble a 1700bp genome, the number of unique sequence states is therefore around 2.1E9. This number is large enough to pose a problem for RAT databases, even if the unique states were distributed non-repetitively among windows. This number also doesn't include the possibility of using scrambling in conjunction with non-synonymous mutations, which could result in an intractably large variant set.

This vulnerability only emerges from considerations of short genomes and a number of recombinations that surpass current feasibility. Nevertheless, it merits closer attention and analysis. More generally, it suggests the lesson that more sophisticated security systems can be relatively superior on many dimensions, while simultaneously opening new vulnerabilities.

Scrambling to evade functional genomics

We do not know the precise schematics of screening protocols arising from IARPA's Fun GCAT, with the exception of SeqScreen.[32] Thinking about how scrambling attacks might apply to functional genomics is therefore quite speculative. Empirical testing of SeqScreen with scrambled sequences would be valuable, but fell outside the limits of this project.

First principles suggest some interesting points, however. First, functional genomics protocols may be vulnerable in a similar way to RAT protocols. Both depend on sophisticated understanding and predictions of biological function, but such knowledge can be overly specific. They analyse sequences and their environment, using the knowledge gained by considering sequences as usual, and environments

[31] Purcell et al. 2018
[32] Balaji et al. 2021

as usual. But sophisticated attackers might leverage unusual interactions, such as how an environment might change a sequence itself, as through the application of recombinases. Recombination events can transform a sequence from nonsense, considered in the usual context, to a hazard, in the unscrambled context. Knowledge-bases that are insufficiently imaginative will fail to foresee and screen for novel forms of known dangers.

This principle is cashed out quite specifically when considering the machine-learning components of functional genomic protocols. The use of machine learning in screening is extremely promising, but care must be taken. Machine learning algorithms are often at the mercy of their training data; when the training data is systematically biased or incomplete, the resulting algorithms can be too. Seemingly dependable systems can be thrown by an unprecedented datum. Scrambled sequences may be one such example.

Fortunately, as in RAT, machine-learning approaches are readily adaptable. Generating scrambled hazard sequences for training data is trivial. Algorithms could leverage fairly obvious data patterns, such as recognising recombinase recognition sites and inverting or excising the enclosed sequence, or they could divine more subtle and unexpected trace traits that scrambled hazards embed.

But while the algorithm has great capacity to defend these attacks, it relies on its human overseers recognising and amending gaps in the data supplied to it. Scrambling is only one way an attacker could leverage the unexpected; rigorous red-teaming may turn up more.[33] Perhaps security approaches that use machine learning components should be like the (now outdated) human-computer 'centaur' teams in chess: computers for the tactics, humans for the strategy.

[33] Zhang and Gronvall 2018

Camouflage with CRISPR

The second obfuscation technique is ‘camouflage’ using CRISPR, a form of steganography. In the protection of integrated circuit designs, physical dummy components can mislead a potential thief looking at the circuit from the top-down using microscopy. A similar method is suggested by Purcell et al. (2018) for synthetic circuit designs.

Purcell et al. (2018) ‘camouflage’ a synthetic circuit design (in this case a bi-stable switch) by introducing false dummy components into the sequence. This is so a competitor with access to the sequence would only be able to deduce a messy, nonfunctional circuit. Then, CRISPR gene editing tools can be targeted to remove or repress particular circuit components. To ‘decamouflage’ and restore the original functional circuit, you would need the molecular ‘decryption key’ - that is, to know the proper guide regions for CRISPR to target.

Purcell et al. suggest this obfuscation method is superior to scrambling because it is reversible, continuous, and exploits a larger space of potential solutions. It is reversible because the CRISPR-based repression can be transient or removable. It is continuous in that it allows the circuit’s output of some incorrect solutions to be continuous with the output of the true circuit. Incorrect solutions can also be designed to appear plausible, with the true solution made to appear implausible. The space of possible solutions is defined by the number of potential subcircuits, which is 2^n where n is the number of genes in the circuit. The limiting factor to camouflage is the number of dummy components that can be successfully repressed with CRISPR.

Camouflage to evade basic bioinformatics

Could camouflage be used to evade screening based on basic bioinformatic tools? We have some real-world evidence that it could. In discussing a potential cybersecurity vulnerability in synthesis screening, Puzis et al. (2020) mention using CRISPR-based camouflage-like obfuscation to order a hazardous subsequence from a synthesis company that is a member of the IGSC.

It should be noted that the hazardous subsequence was very short. Revealed in a separate paper, the hazardous subsequence was an α-conotoxin called PeIA.[34] Assuming the order used the full prepropeptide of this toxin, the sequence was 120bp long.[35] As the relevant screening window was the standard 200bp, the red-team attack consisted of splitting the 120bp into at least 2 pieces, and embedding them in a surrounding sequence that would lead to a higher 'Best Match' organism that was not a select agent. With the order cleared for printing, the attacker could then construct the hazardous toxin via CRISPR-Cas9 mediated deletion of the intervening camouflage section. The order was stopped before printing and the company was informed.

While easy access to toxins does pose some danger, it is dwarfed by the risk of access to whole viruses on the Select Agent program. Would such efforts enable access to substantially longer hazards?

Consider a screening protocol that uses basic BLAST and a 'best match' approach. To evade this protocol, a camouflage transformation would need to introduce decoy elements sufficient to change the highest scoring classification from a hazardous genome for every possible 200 bp window. Estimating the ratio of 'camouflage' basepairs from 'hazard' basepairs is difficult, as it would depend on the relative

[34] Farbiash and Puzis, 2020
[35] McIntosh et al. 2005

scoring possibilities of each fragment, and how fragment scores could overtake their neighbours within the sliding 200 bp window.

We can get a very rough estimate by considering maximal and minimal scores, given a particular scoring system. BLOSUM-62 is the standard scoring matrix in BLAST. In BLOSUM-62, the highest score possible for comparing two amino acids is 11, which results from identically placed tryptophan. The lowest possible score for an identity is 4, which is given for the identically-placed of alanine, isoleucine, leucine, serine, or valine. Scoring matrices also have negative scores, for particularly mismatched amino acids. Since seeds stop when the score begins declining, we can simplify by only considering high-scoring seeds.

Assume a camouflage section that scores 11 in every place. Note the generosity of this assumption - it would occur in chains of tryptophans alone. Assume a hazard sequence that scores 4 in every place. This would only occur if the chain consisted solely of alanine, isoleucine, leucine, serine, or valine. Given these unrealistic assumptions that favour camouflage, what is the minimum ratio of camouflage to hazard in a 66 amino acid (200bp) window where the camouflage score remains higher than the hazard score?

We can combine this ratio of scores ($11*C > 4*H$) with the fact that the hazard sequence and camouflage sequence must sum to the size of the window ($C+H=66$). Solving, we find that the hazard sequence must be less than 48 amino acids (in a 66 amino acid sequence), using maximally generous assumptions.

This requirement means that the hazard sequence must be broken up into pieces that are shorter than 144bp. While feasible for short toxins, this requirement is severe for larger hazardous constructs. The smallest genome of 1700bp would need to be broken up into 12 pieces to achieve this, which would then require 11 necessary CRISPR-guided deletions. Longer constructs and realistic score ratios compound the difficulty and inefficiencies, likely rendering such an attack infeasible.

A camouflaged sequence would also be extremely vulnerable to the same panoply of follow-up bioinformatics tools as in the scrambling case. In particular, metagenomic methods would be well-suited to identify hazards within camouflaged sequences.

Camouflage to evade random adversarial thresholds

To consider whether camouflage attacks could evade RAT, we can apply a similar model. Instead of score thresholds within sliding 200bp windows, camouflage would need to prevent exact matches to 57-mer variants. First, because the attacker would need to avoid all possible intact 57-mers to be confident of evasion, they would need to introduce camouflaging subsequences at least every 56bp. Since the RAT database includes millions of variants per window, merely changing the last basepair or amino acid is not sufficient.

An attacker would have to change the final five amino acids to hide the 57-mer in the resulting 20^5 (3.2E6) combinatorial space. This would lead to only using 42bp hazard fragments, each of which would be separated by a 15+bp camouflaging fragment. As in the preceding section, the difficulty and inefficiency of such a severe fracturing render the attack infeasible.

Camouflage to evade functional genomics

Intuitively, functional genomics would likely be able to leverage fragments smaller than 144bp (those relevant to BLAST) or even 42bp (those relevant to RAT) analysis. In fact, SeqScreen directly tested its application to fragments as short as 34bp, and found excellent performance (Balaji et al.) As such, functional genomic systems are unlikely to be evaded by a feasible camouflage attack.

Future obfuscation technology

Forecasting the future of biotechnology

Forecasting future technological developments is hard.[36] Forecasting biotechnology is particularly hard, because innovations in biotechnology are often adapted from discoveries in the natural world. This means that qualitatively new techniques can arrive quickly and the capabilities of the field can advance discontinuously.

Two claims justify attempts to forecast biotechnology. First, some continuous trends are trackable, and by extrapolating them we can say sensible things about the possible futures they imply. Second, by analysing potential problems in advance, we may be able to recognise indicators of problematic developments early. These indicators might be recognisable even if they are discontinuous with previous advances. With early recognition, we may be able to build safeguards proactively or influence the direction and shape of the nascent technologies.[37]

Forecasting the future of sequence obfuscation

The preceding sections argue that current techniques for obfuscating circuits probably won't be able to obfuscate pathogens from current screening protocols. How might this change in the future?

By contrasting synthetic circuits and pathogen genomes and their purposes, we find theoretical expectations that circuits should be more amenable to obfuscation than genomes. The design of synthetic genetic circuits is still a nascent stage, but generally they are made up of well-understood and well-known components that encompass functional genes or transcription factors. Circuits are designed by choosing

[36] Thomas, 2001
[37] Bostrom, 2002

and grouping particular components together such that the designer can combine them to make more complex functions. The key information to be obfuscated is the high-level organisation of these components. They aim to make the overall organisation of the sequence ambiguous to an unauthorised sequence reader, so that hard-earned designs are not distinguishable from similar but inferior comparators. They are not usually trying to mask the identities of these basic components.

Genomes, in contrast, embed information about their identities across a range of structural levels encoded in their sequence. An actor looking to evade screening must obfuscate the sequence to such a degree that none of this identifying information is retrievable. When this information can be embedded in features ranging from the ratios of individual nucleotides or short *k*-mers to whole genes and gene networks, throughout most of a sequence, obfuscation faces a far harder objective.

Obfuscation through scrambling can also interfere with coding functions, so (consistent with Purcell et al.'s demonstrations), recombinase recognition sites are likely to be placed outside the functioning genes and transcription factors - outside the primitive components. While this pattern is sufficient for obfuscating circuit topology, the basepair length of these primitive components is large enough that a hazard genome's identity would be recognisable if they are left intact.

This contrast in obfuscation aims is reassuring, but this situation may not be stable. Obfuscation techniques aim to protect intellectual property instantiated in the large scale features of sequences; if intellectual property is believed to reside in more granular features of a sequence, then methods of disguising shorter and shorter fragments (e.g. 30-100bp) might become increasingly sought after. If this occurred, there would be a greater overlap in the obfuscation requirements of designs and genomes, and correspondingly a greater worry of malicious pathogen obfuscation.

Why might the intellectual property of synthetic biology become so granular? Some products can be functional with very short lengths, such as primers, toxins, or short peptides. While synthetic biology is driving towards a library of standardised parts[38] it's possible that, if the field goes awry, parts might become increasingly proprietary. In such a case, designers might want to hide even the primitive components that circuits are made up of. This would induce a worrying overlap between the obfuscation targets of legitimate designers and malicious actors.

The field's direction also matters for norm setting. Obfuscation remains a little-used technology. That means that even in the case of a successfully obfuscated order, detection of obfuscation itself would be grounds for suspicion and follow-up. Requiring obfuscation to also be undetectable - to be true steganography - would dramatically decrease it's technical possibility.

We can see this pattern at work in scrambling. As genomes are scrambled, they do not come to look like other, unrelated genomes. The space of possible sequences is too vast for such a pattern to occur coincidentally or to be orchestrated intentionally. Instead, as scrambled genomes look less and less like themselves, they look more and more like biological nonsense. This facilitates the detection of scrambling. But reacting to obfuscation with suspicion and scrutiny depends on scrambling being rare; if it became widely used or normalised, attending to scrambled sequences would be an increasingly onerous and unpopular task.

Norms also affect innovations in obfuscation. Current techniques only obscure the sequence when it is being ordered, synthesized, stored, and transported. If the synthetic biology product is performing its function, it's sequence will be functional and therefore readable. Traditional protections of proprietary devices are possible, such as keeping a genetically engineered drug-producing bug in a secured

[38] Baldwin 2012

pharmaceutical lab. However some synthetic biology products will likely have to perform their functions out in the world, and in those situations the sequence design would be vulnerable to being read.

Such a situation might push innovation towards developing a form of obfuscation that would work 'in situ'. That is, obfuscation that would allow the sequence to remain functional, but could not be sequenced legibly. In situ obfuscation would likely be qualitatively novel, highly valuable, and a potential screening problem. Schematics would depend on the specifics of current and future sequencing approaches and their potential countermeasures. This makes it hard to speculate, and specific analysis is outside this project's scope. However, innovation often follows market incentives, and technological forecasts often fail via a failure of imagination. As such, in situ obfuscation should not be dismissed, but noted as a security concern that could be on the horizon.

Advances in scrambling and camouflage

Scrambling is currently limited by the number of orthogonal recombinases. Synthetic biology is increasing in publications, synthesis volume, and venture capital (Ord 2020). It seems likely that with this increasing effort more orthogonal recombinases and recognition sites will be discovered and refined. Considering our model of scrambling attacks however, this trend does not appear very troubling. The infeasibility of scrambling attacks is not sensitive to small linear additions to the set of recombinases. The required number of required steps and vanishing efficiency rates suggest that scrambling will remain infeasible even if the number of orthogonal recombinases increased by several times, an unlikely event if recombinases continue to be discovered and refined one at a time.

What would be more troubling is if scrambling escaped the orthogonality requirement. Splitting genomes, scrambling the pieces independently, and then assembling them is one way of loosening the bonds of

orthogonality, bought at a cost of the increased technical difficulty and inefficiency of assembly. A more speculative worry is if the orthogonality requirement is transcended altogether. Orthogonality is a crucial factor in the design of synthetic circuits more generally (Jusiak et al.), so progress in this area is well motivated. What should concern us is a recombination system that is *arbitrarily orthogonal*, where recombinations could be performed with linear efficiency costs and low risk of crosstalk. Such a system could deliver the number of inversions and excisions required to heavily scramble sequences, resulting in a customer-encryption method that screening protocols would have difficulty defending.

Even this concern has limitations however. A heavily scrambled sequence would still likely appear like biological nonsense, and could trigger closer scrutiny. Such scrutiny might also be triggered by the identification of a large (according to some threshold) number of recombinase recognition sites. Then, a sophisticated screening system could test the early branches of the permutation tree, searching for any fragmentary clues of a hazard. This technical solution might be dependent on a normative one - it is reliant on being in a world where such scrambling is not common practice.

A final consideration for the future of scrambling is the possibility of seamless scrambling. Our simplified model assumed such seamlessness, and still suggested the method was infeasible. In the real world, recombinase recognition sites can disrupt coding regions and take up space. As such, they can serve as simple indicators that a sequence is likely scrambled, and point to possible permutations. Should a recombination system be developed that leaves no traces in the sequence, our model should become less reassuring, and our easy footholds for the screening problem would vanish.

Camouflage is dependent on CRISPR, or, more generally, precise and reliable methods of genetic alteration. Previous sections argued that while adding dummy components is helpful for disguising circuit designs, it isn’t well-suited to disguising pathogen genomes. Puzis et al. 's (2019) actual use of the method to evade an IGSC company’s screen is notable and deserves attention, but we should keep in mind that the

relevant hazard was an extremely short toxin. If CRISPR or a similar system became efficient and reliable enough to introduce a very large number of dummy subsequences, then camouflage may become a potential evasion method for larger hazards. Increasingly sophisticated screening methods use shorter and shorter sections as clues of identity, and this trend increases the required amount of dummy subsequences. It's unclear if such a method is easier than just synthesizing the construct independently. As such, security-minded people should continue closely watching the growing ease, efficiency, and reliability of CRISPR.

Quantum computing

Predicting what new breakthroughs will emerge in synthetic biology is confounded by the natural world's inventiveness and our ignorance of it. We have better insight on our tools for navigating the biological world, and one development we can dimly foresee is the potential impact of quantum computing. One of the major applications expected by quantum computing experts is the simulation of quantum systems, including biomolecules for the development of pharmaceuticals.[39] [40] This may allow a much faster exploration of the space of possible sequences, with unclear security repercussions. It is even less clear what, if any, impact such simulations would have on the balance between sequence screening and evasion.

Quantum computing is also likely to allow greatly optimised search functions, and this could be usefully applied to the problems of large databases and vast possible searches faced by sequence alignment. However, such an application would require cheap, routine access to quantum search algorithms, which is an unlikely possibility for several decades.

[39] de Wolf 2017
[40] Nielsen 2019

The final consideration of quantum computing concerns it's most discussed and concrete impact: breaking standard encryption methods. Several widely used cryptography methods rely on using very hard mathematical problems, such as factoring large numbers. Quantum algorithms (e.g. Shor's quantum-factoring algorithm) threaten to break such foundations, sparking the fervent interest of intelligence agencies and the proactive development of 'quantum-resistant cryptography'.

Given that scrambling is referred to as 'encryption', one might expect post-quantum cryptography to be a significant factor in the future of sequence obfuscation. On intuition however, this seems unlikely. Biologically instantiated sequences are a much sludgier medium to perform encryption and decryption on compared to clean, weightless digital strings. The kinds of cryptography that are relevant to quantum breaking and resistance are high-speed, high-volume, and highly reliable, and they probably couldn't be performed successfully on biological sequences. A similar intuition rejects the speculative possibility of quantum-computing based encryption for sequences - sequences are probably the wrong kind of object to embody such schemes.

Genetic recoding

Expanding the natural genetic code is a promising avenue of synthetic biology.[41] By re-assigning redundant codons, developing a quadruplet codon scheme, or by introducing new, non-natural nucleotides, biologists could enhance DNA transcription and translation mechanisms to implement non-canonical amino acids, greatly expanding the range of synthetic genetic designs and applications. These efforts are currently a nascent field, but advances are probable. Such advances are also likely to introduce new screening evasion methods.[42]

[41] de la Torre and Chin, 2021
[42] Kobokovich et al. 2019

Sequences that use non-natural nucleotides would not be legible to current DNA synthesis suppliers, and are more likely to be independently synthesized. However, should non-natural bases become a more common technique, such that commercial synthesis suppliers include them in their services, effective screening seems difficult. Using non-natural bases requires deoxynucleoside triphosphate analogues, while the use of non-canonical codons relies on new aminoacyl-tRNA synthetases (aaRS) and tRNA pairs, as well as an expanded ribosomal polymerisation capability.[43] If these other parts of the system are standardised, interpretation of orders that include non-natural bases may be possible.

Non-canonical codons that re-assign redundant codons or use quadruplet codons may be more difficult to screen. Orders using them will appear much less unusual than those using non-natural bases, but they will likely remain detectable as biological oddities deserving of closer scrutiny. Deriving their end-function will be difficult, as it will be contingent on the specific biochemical context it is used in. However, standardisation of aaRS and tRNA pairs could again aid interpretation.

A speculative worry is an order that uses non-canonical quadruplet codons to encode a hazard, but whose triplet codon interpretation suggests a safe construct. In such a case ordinary screening software would suggest a normal DNA order, but in the user's carefully constructed biochemical environment its quadruplet transcription and translation would result in a dangerous pathogen. Orchestrating such a scheme would be a technical nightmare, and impossible with current technology. Even in the future, we might expect that an actor with such sophistication would not require external synthesis services, rendering screening moot. Nevertheless, we do not know how genetic recoding will develop and diffuse through the community, so red-teaming creatively and proactively should be considered as recoding advances.[44]

[43] de la Torre and Chin, 2021

[44] Zhang and Gronvall 2018

Other protections of sequence intellectual property

The future of synthetic biology need not solely pose problems to screening; it can generate solutions too. The key driver of obfuscation techniques in the first place is the protection of intellectual property instantiated in sequences. In the future such intellectual property may not need protection, or may be protected in non-obfuscatory ways.

In general, developing stringent intellectual property protections is contingent on the nature of the field and how it develops. Scientific norms of openness and commonality of resources may prevail over commercial incentives. If use of the intellectual property is recognisable and well-policed, there is much less incentive to go to extreme lengths to ensure secrecy. Genetic engineering already bears tractable signs of authorship unintentionally.[45] [46] Intentionally designed and implemented authorship tags in the sequences themselves could be subtle, specific, and relatively indelible . Strategic use of devices like copyright traps or watermarking could enable the protection of intellectual property without a reliance on secrecy, dissolving the potential tension between the incentives of the synthetic biology industry and biosecurity.

Implications for the screening landscape

Perfect security is an impossibility. In the real world security is always traded off against other values, with the trading negotiated by a set of heterogeneous actors with particular incentives and agendas. Current screening of DNA synthesis orders is imperfect and incomplete in substantial ways, despite the good intentions, intellectual sophistication, and hard effort of a range of actors from government, academia, and industry. In parallel with these good faith motives however, there also exist motives of

[45] Alley et al. 2020
[46] Wang et al. 2021

providing reassurance and escaping liability - motives satisfied by the performative aspects of synthesis screening, or 'security theatre'.[47]

The burden of screening is already increasing as a relative cost, and advances in obfuscation threaten to markedly increase this burden in the future. If this happens, the aims satisfied by security theatre will significantly decouple from the aims of genuine biosecurity. While obfuscation may pose a technical challenge, the greater challenge derives from how obfuscation could expose the shaky structure of current screening.

Note on information hazards

Appropriate management of information hazards in biosecurity is not captured by a simple openness-secrecy axis.[48] Nuances of this project point to such complexities. Critically examining security protocols for vulnerabilities carries risk. Expert advice early in the project gave confidence that there was a low chance of exploitable vulnerabilities in current protocols with current obfuscation techniques. An exception was the camouflage vulnerability shown by Puzis et al. (2020), but their publication in *Nature Biotechnology* and application to a very short hazard suggested that the benefits of continued analysis outweighed the risk. Vulnerabilities discussed in the paper were generally theoretical and based on possible future developments. Exploring them served three purposes. First, to inform the foresight of other actors in noticing and analysing trends and breakthroughs in synthetic biology. Second, to point out that increasing sophistication and performance in some domains may open new potential vulnerabilities, which might be responded to early. Third, to seed imaginative red-teaming efforts, which can (with proper collaboration and structure) perform a central role in robust security systems.

[47] Schneier 2013
[48] Lewis et al. 2018

Conclusion

Sequence obfuscation appears infeasible using currently available techniques on currently used screening protocols. Linear advances in scrambling techniques might present problems to the random adversarial threshold approach, but this possibility deserves greater scrutiny. Future drivers and directions of sequence obfuscation are hard to predict, but some warning signs can be sketched to inform future observers. These are: intellectual property in synthetic biology that is encoded in very short sequences; norms of common use of obfuscation; obfuscation in situ; arbitrarily orthogonal recombination; seamless scrambling; and diffusion of genetic recoding. While the overall message is reassuring, vulnerabilities in the future are possible and significant. Imaginative and collaborative red-teaming appears to be a useful tool in the biosecurity arsenal.

Acknowledgements

Thanks to the staff and fellows of the Stanford Existential Risk Initiative for funding, support, advice and encouragement. Thanks to Dr. Michael Montague for incredibly useful guidance, advice, and interesting chats. Thanks also to the Future of Humanity Institute and Dr. Cassidy Nelson, who supported me on a related but unfinished project in 2020 which gave me a greater understanding of the screening problem, and Dr. Greg Lewis of same for mentioning steganography as a potentially interesting area to look into.

References

1. Alley, E. C. et al. 2020. “A machine learning toolkit for genetic engineering attribution to facilitate biosecurity.” *Nature Communications* https://doi.org/10.1038/s41467-020-19612-0

2. Altschul, Stephen F., Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. 1990. "Basic Local Alignment Search Tool". Journal Of Molecular Biology 215 (3): 403-410. doi:10.1016/s0022-2836(05)80360-2.
3. Balaji, Advait, Bryce Kille, Anthony D. Kappell, Gene D. Godbold, Madeline Diep, R. A. Leo Elworth, and Zhiqin Qian et al. 2021. "Seqscreen: Accurate And Sensitive Functional Screening Of Pathogenic Sequences Via Ensemble Learning". *Arxiv.* doi:10.1101/2021.05.02.442344.
4. Baldwin, Geoff. 2012. *Synthetic Biology -- A Primer:Revised Edition*. Imperial College Press.
5. Bartoszewicz, Jakub M, Anja Seidel, Robert Rentzsch, and Bernhard Y Renard. 2019. "Deepac: Predicting Pathogenic Potential Of Novel DNA With Reverse-Complement Neural Networks". *Bioinformatics*. doi:10.1093/bioinformatics/btz541.
6. Baum et al. N.D. “Cryptographic Aspects of DNA Screening” *SecureDNA*. https://www.securedna.org/download/Cryptographic_Aspects_of_DNA_Screening.pdf
7. Bostrom, Nick. 2002. "Existential Risks: Analyzing Human Extinction Scenarios". 9 *Journal of Evolution and Technology.*
8. Carter, Sarah, and Diane DiEuliis. 2019. "Mapping The Synthetic Biology Industry: Implications For Biosecurity". *Health Security* 17 (5): 403-406. doi:10.1089/hs.2019.0078.
9. Carter, Sarah, and Robert Friedman. 2015. "DNA Synthesis And Biosecurity: Lessons Learned And Options For The Future". *J. Craig Venter Institute*. https://www.jcvi.org/research/dna-synthesis-and-biosecurity-lessons-learned-and-options-future

10. de la Torre, Daniel, and Jason W. Chin. 2020. "Reprogramming The Genetic Code". *Nature Reviews Genetics* 22 (3): 169-184. doi:10.1038/s41576-020-00307-7.
11. de Wolf, Ronald. 2017. The Potential Impact of Quantum Computers on Society. *Arxiv.* https://arxiv.org/pdf/1712.05380.pdf
12. Department of Health and Human Services, 2010, "Screening Framework Guidance for Providers of Synthetic Double-Stranded DNA" https://www.phe.gov/s3/law/syndna/Documents/syndna-guidance.pdf
13. DiEuliis, Diane, Sarah R. Carter, and Gigi Kwik Gronvall. 2017. "Options For Synthetic DNA Order Screening, Revisited". *Msphere* 2 (4). doi:10.1128/msphere.00319-17.
14. Farbiash, Dor, and Rami Puzis. 2020. Cyberbiosecurity: DNA Injection Attack in Synthetic Biology. *Arxiv*. https://arxiv.org/pdf/2011.14224.pdf
15.
16. Goodwin, Leslie O., Erik Splinter, Tiffany L. Davis, Rachel Urban, Hao He, Robert E. Braun, and Elissa J. Chesler et al. 2019. "Large-Scale Discovery Of Mouse Transgenic Integration Sites Reveals Frequent Structural Variation And Insertional Mutagenesis". Genome Research 29 (3): 494-505. doi:10.1101/gr.233866.117.
17. Gretton et al. N.D. "Random adversarial threshold search enables specific, secure, and automated DNA synthesis screening" *SecureDNA*. https://www.securedna.org/download/Random_Adversarial_Threshold_Screening.pdf
18. Jusiak et al. "Synthetic gene circuits", in *Synthetic Biology* by Meyers R.A. (ed.). Wiley.
19. King, Melissa. 2021. "Functional Genomic And Computational Assessment Of Threats (Fun GCAT)". *Iarpa.Gov*.

https://www.iarpa.gov/index.php?option=com_content&view=article&id=752&Itemid=354.

20. Kobokovich, Amanda, Rachel West, Michael Montague, Tom Inglesby, and Gigi Kwik Gronvall. 2019. "Strengthening Security For Gene Synthesis: Recommendations For Governance". *Health Security* 17 (6): 419-429. doi:10.1089/hs.2019.0110.
21. IGSC Harmonized Screening Protocol v2.0. International Gene Synthesis Consortium; 2017. https://files.codexdna.com/docs/IGSC_Harmonized_Screening_Protocol.pdf
22. Lewis, Gregory, Piers Millett, Anders Sandberg, Andrew Snyder‐Beattie, and Gigi Gronvall. 2018. "Information Hazards In Biotechnology". *Risk Analysis* 39 (5): 975-981. doi:10.1111/risa.13235.
23. McIntosh, J. Michael, Paola V. Plazas, Maren Watkins, María E. Gomez-Casati, Baldomero M. Olivera, and A. Belén Elgoyhen. 2005. "A Novel A-Conotoxin, Peia, Cloned From Conus Pergrandis, Discriminates Between Rat A9α10 And A7 Nicotinic Cholinergic Receptors". Journal Of Biological Chemistry 280 (34): 30107-30112. doi:10.1074/jbc.m504102200.
24. Millett, Piers, and Andrew Snyder-Beattie. 2017. "Human Agency And Global Catastrophic Biorisks". *Health Security* 15 (4): 335-336. doi:10.1089/hs.2017.0044.
25. Nielsen, Michael. 2019. “Quantum computing for the very curious”. *Quantum country*. https://quantum.country/qcvc
26. Noyce, Ryan S., Seth Lederman, and David H. Evans. 2018. "Construction Of An Infectious Horsepox Virus Vaccine From Chemically Synthesized DNA Fragments". *PLOS ONE* 13 (1): e0188453. doi:10.1371/journal.pone.0188453.

27. Nuclear Threat Initiative and World Economic Forum. 2020. “Biosecurity Innovation and Risk Reduction: A Global Framework for Accessible, Safe and Secure DNA Synthesis” *World Economic Forum: Insight Report.*

28. Prorocic, Marko M. and William .M. Stark. 2013, “Site-specific recombination” in *Brenner’s Encyclopedia of Genetics* (Second Edition).

29. Purcell, Oliver, Jerry Wang, Piro Siuti, and Timothy K. Lu. 2018. "Encryption And Steganography Of Synthetic Gene Circuits". *Nature Communications* 9 (1). doi:10.1038/s41467-018-07144-7.

30. Puzis, Rami, Dor Farbiash, Oleg Brodt, Yuval Elovici, and Dov Greenbaum. 2020. "Increased Cyber-Biosecurity For DNA Synthesis". *Nature Biotechnology* 38 (12): 1379-1381. doi:10.1038/s41587-020-00761-y.

31. Schneier, Bruce. 2013. *Beyond Fear.* New York: Springer Science+Business Media.

32. Thomas, J.M. 2001. "Predictions". Notes And Records Of The Royal Society Of London 55 (1): 105-117. doi:10.1098/rsnr.2001.0128.

33. Wang, Qi, Bryce Kille, Tian Rui Liu, R. A. Leo Elworth, and Todd J. Treangen. 2021. "Plasmidhawk Improves Lab Of Origin Prediction Of Engineered Plasmids Using Sequence Alignment". Nature Communications 12 (1). doi:10.1038/s41467-021-21180-w.

34. West, Rachel, and Gigi Kwik Gronvall. 2020. "California Shows The Way For Biosecurity In Commercial Gene Synthesis". Nature Biotechnology 38 (9): 1021-1021. doi:10.1038/s41587-020-0667-0.

35. Zhang, Lisa, and Gigi Kwik Gronvall. 2018. "Red Teaming The Biological Sciences For Deliberate Threats". *Terrorism And Political Violence* 32 (6): 1225-1244. doi:10.1080/09546553.2018.1457527.