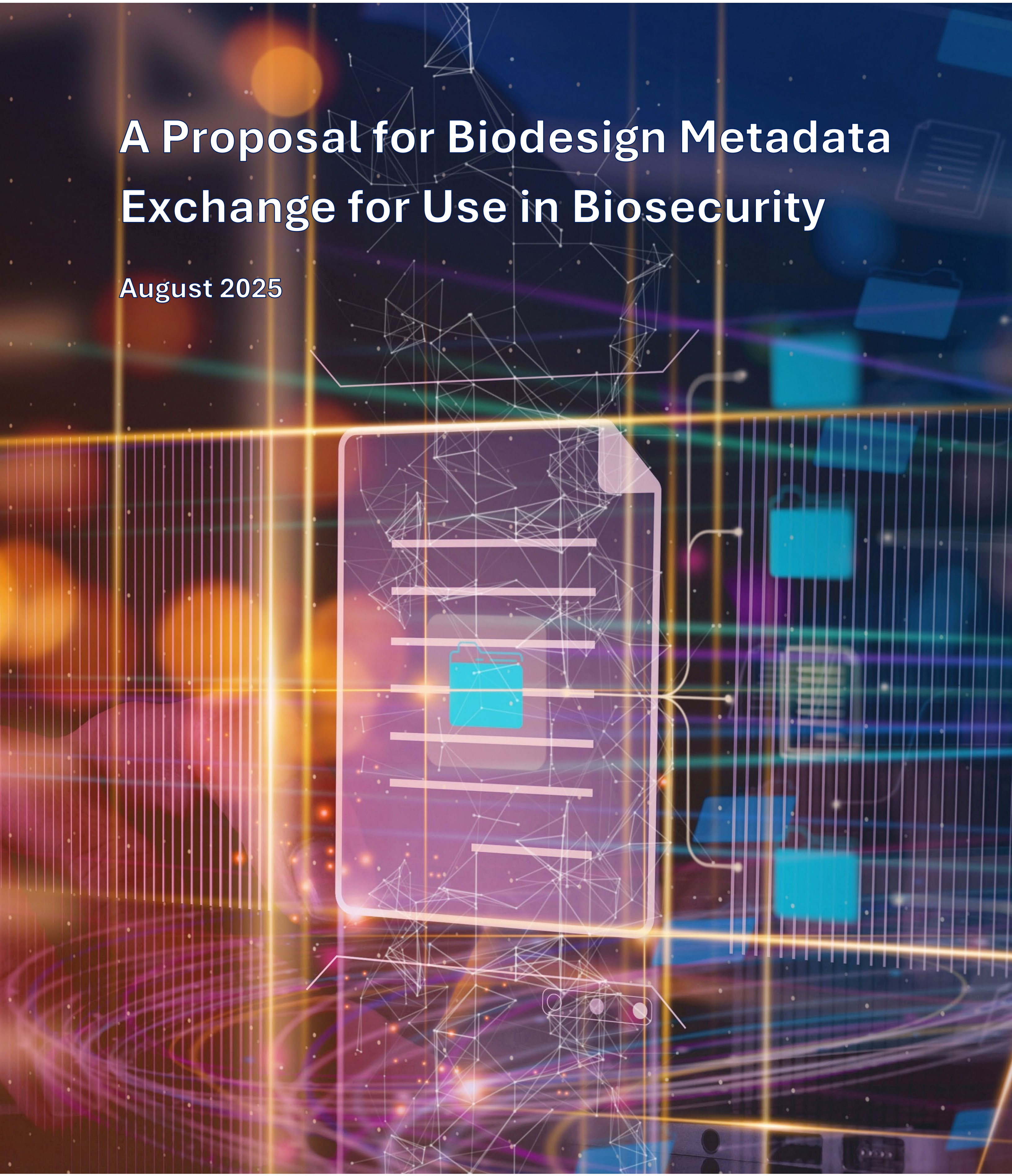


A Proposal for Biodesign Metadata Exchange for Use in Biosecurity

August 2025

Contributors

Lattice Automation
Dr. Douglas Densmore (D.D.)
Mr. Isaac Guerreiro (I.G.)
Mr. Christopher Krenz (C.K.)
Mr. Kevin LeShane (K.L.)
Mr. Nicholas Rosenau (N.R.)
Mr. Guzman Vigliecca (G.V.)

NTI | bio
Mr. Greg Butchello (G.B.)
Mr. Christopher Isaac (C.I.)
Ms. Sara Kaufman (S.K.)
Ms. Hayley Severance (H.S.)
Dr. Jaime Yassif (J.Y.)
Dr. Sarah R. Carter, consultant (S.C.)
Dr. Nikki Teran, consultant (N.T.)
Dr. Nicole Wheeler, consultant (N.W.)

Licensing and Attribution

Acknowledgments

The Lattice Automation team gratefully acknowledges Samuel Curtis (U.S. Center for AI Standards and Innovation), Nick Randolph (University of North Carolina at Chapel Hill), Jacob Beal (BBN Technologies), Elise de Reus (Cradle), James Diggans (Twist Bioscience), Kevin Flyangolts (Aclid), and Chris Myers (University of Colorado Boulder).

In addition to those previously listed, The Nuclear Threat Initiative (NTI) team also acknowledges Anthony Gitter (University of Wisconsin-Madison), Scott Nolan Smith and Carmen MacDougall (NTI), and Claire Qureshi (Sentinel Bio). NTI also wants to acknowledge the members of the Responsible AI x Biodesign community and the DNA synthesis screening community for their commitment to developing effective safeguards for biological AI tools.

Contents

Executive Summary

As advancements in biodesign accelerate, society faces increasing biosecurity risks, particularly, the accidental or intentional creation of harmful biological agents. Current biosecurity frameworks, such as those used by the International Gene Synthesis Consortium (IGSC), depend on screening synthetic DNA sequences by comparing them to known pathogenic sequences. However, as biodesign tools—especially those powered by artificial intelligence (AI)—begin exploring novel biological designs that deviate from nature, traditional screening methods may struggle to detect potential threats. This situation presents a significant challenge for biosecurity decision-makers who must assess the risks of entirely new designs that do not resemble known organisms or toxins.

As summarized in box ES. 1, this report introduces the concept of the Biodesign Metadata Exchange (BMDE), designed to address these challenges by capturing and transmitting metadata—such as design provenance, editing history, and intended use—alongside DNA or protein sequences. Providing added context can help biosecurity stakeholders assess risks more effectively by increasing their understanding of not only the sequence itself but also the design process behind it. Using BMDE strengthens biosecurity by providing standardized transparency, process, and context during the biodesign process, thus ensuring that emerging risks from advanced biological engineering are identified and managed more effectively.

Box ES.1. Structure of the Report

Section 1 provides a high-level introduction to the biodesign landscape, biosecurity challenges, and the purpose of the Biodesign Metadata Exchange (BMDE) proposal. This section is for **policymakers and biosecurity strategists** who need to understand the problem and the proposed solution in broad terms.

Section 2 dives into the technical aspects of the BMDE system, covering the specifics of how metadata is captured, transmitted, stored, and validated. This section is for **technical implementers and biosecurity practitioners** who need detailed knowledge of how BMDE works and how it can be applied.

Appendix A contains technical specifications, and additional resources. This section is for **developers and researchers** seeking technical specifics for implementation or further exploration.

Appendix B contains a series of case studies highlighting example situations in which capturing metadata from a biological design tool would provide a biosecurity benefit. This section is for **biosecurity practitioners** seeking more information about potential use cases for the BMDE.

Section 1: Overview and Motivation

Our ability as a society to engineer biology continues to increase. These advances have profound implications for a basic understanding of biology as well as for innovations in healthcare, the environment, human health, and manufacturing. However, as more possibilities for novel, functional biology develop, so too does the risk of accidental or intentional design of potentially harmful biological products.

Biosecurity experts have outlined potential risks related to biodesign tools, particularly those enabled by artificial intelligence (AI),[1] and have worked to bolster biosecurity screening frameworks to capture a broader range of potentially harmful biological designs.[2] However, as these designs (i.e., DNA or protein sequences) become less like those found in nature, screening based on the sequence itself becomes more challenging.

This document focuses on a new approach. By electronically capturing the “metadata” associated with a user’s interactions with biodesign tools as new designs are created, more information becomes available about the design’s origins, alterations, edits, and intended functions, which can inform biosecurity screening and decision-making.

The Biodesign Landscape

With the expansion of biodesign tools, humans can increasingly tackle more of the “biological design space.” Currently, the biodesign landscape can be considered through the lens of the “design-build-test-learn” cycle.[3] The number of distinct proteins typically found in nature is on the order of 10^{12}, while the whole design space for a protein with 200 amino acids is on the order of 20^{200}.[4] The increasing pace of new design software not only enables designers to draw on the designs already found in nature but also allows them to start exploring the vast parameter space outside the one already touched by the natural evolutionary process.

The Biosecurity Landscape

It is critically important to guard against the malicious or accidental creation of hazardous biological agents. Biological designs (i.e., DNA, mRNA, or protein sequences) can cause harm only when they are translated from digital sequences into physical biological systems. Biosecurity screening of synthetic DNA is a critical safeguard at this digital-physical interface.

In adherence with best practices established by the International Gene Synthesis Consortium (IGSC) and under existing U.S. government policy,[5] commercial DNA providers that supply the vast majority of synthetic DNA for biological applications conduct biosecurity screenings. They either have developed their own screening systems or have taken advantage of the “Common Mechanism,” which is available through the

International Biosecurity and Biosafety Initiative for Science. These screenings help ensure that potentially harmful sequences of DNA are shipped only to customers who have a legitimate use for them.

Current tools for screening DNA sequences flag those that are similar to harmful sequences found in known pathogens and toxins.[6] Although tools for screening biological sequences against those found in pathogenic organisms are well developed, they may struggle to identify potentially harmful sequences found outside of nature.[7] As biodesign tools become more commonly used to alter, optimize, and explore a broader range of biological possibilities, it will become more difficult for screening tools to accurately and efficiently determine if a sequence might pose some biological risk. Thus, it will be important to create new screening and audit methods to keep pace with the exponential growth of biodesign software.

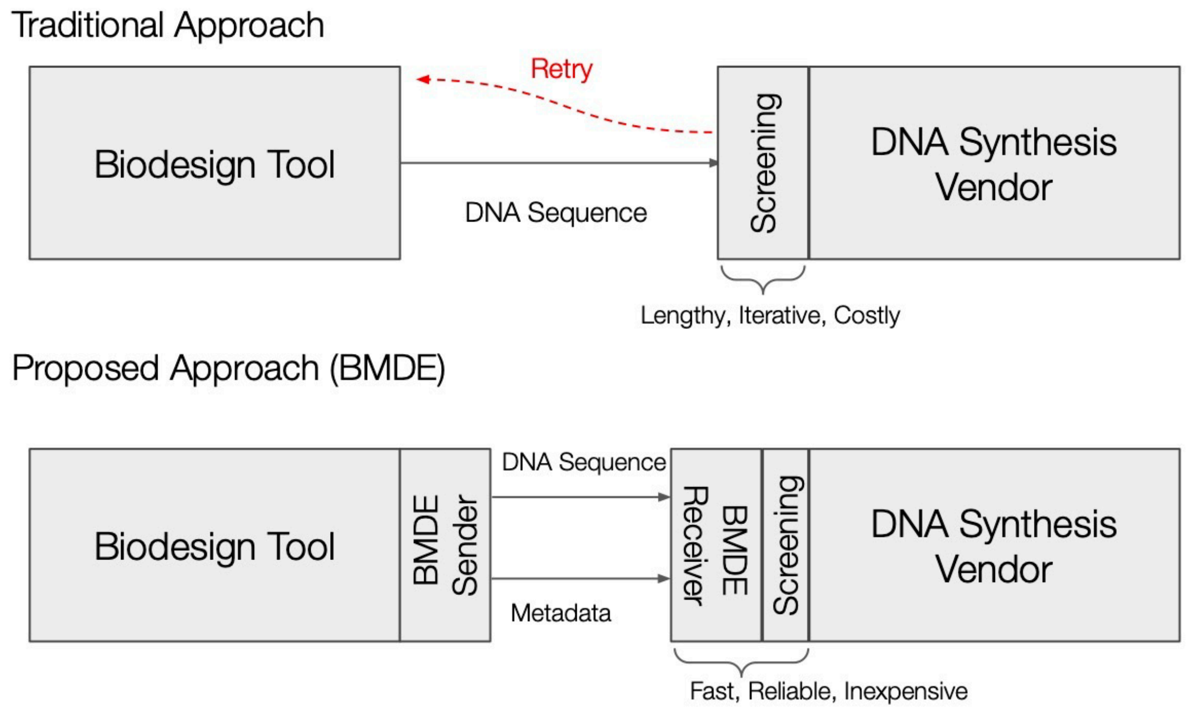
The BMDE Solution

The proposed new approach, the **Biodesign Metadata Exchange (BMDE)**, addresses this challenge by capturing and transmitting detailed metadata about the biodesign process—metadata that provides critical information beyond the design itself. This metadata includes the design's origin, editing history, intended application, and the tools and individuals involved in its creation. By offering this additional context, BMDE can enable more accurate risk assessments of biological designs, particularly when the sequences do not resemble those found in nature.

The BMDE methodology enables secure and standardized data transmission between biodesign tools, DNA synthesis vendors, and other relevant entities. This approach can be integrated into existing biosecurity screening frameworks, enhancing their ability to evaluate novel biological sequences and ensuring that potentially dangerous designs are identified and scrutinized. In addition, the BMDE can be made "backward compatible" with existing data transmission efforts and can be computationally efficient in both space and time.

Furthermore, BMDE supports improved decision-making by providing biosecurity experts with the necessary information to assess risks associated with both familiar and novel biological designs, making it a critical safeguard as biodesign continues to evolve. Finally, BMDE's utility can expand outside biosecurity to include workflow capture and design quality assurance. These attributes can both enhance biosecurity and move the general field of biodesign forward. Figure 1 presents a high-level illustration of BMDE versus the traditional approach.

Figure 1: Comparison of Traditional and Proposed Approaches



Note: Traditional approaches send the sequence (DNA or protein) to the third-party vendor directly. The vendor must then perform the screening of the sequence based on the sequence alone. If the sequence is novel and does not match any known sequences, it can be difficult to determine whether it is benign without additional context. Without the proposed BDME approach, the provider must take time to investigate further, often reaching out to the customer, or must make an informed judgment. The new approach sends the sequence along with metadata on the design. The metadata provides context that allows for more informed analysis. This saves time, reduces false positives and false negatives, and will also enable other innovative biodesign activities, all while remaining compatible with previous, sequence-only approaches during broad adoption.

Section 2: Proposed Approach (BMDE System)

Goals

The BMDE approach aims to achieve the following three goals:

- **Information Capture:** Record the required data to understand the design process and potential intent behind a sequence's design "journey."
- **Efficient Format:** Minimize the overhead (in space and time) of producing, storing, interpreting, and transferring data.
- **Broad Application:** Be able to capture design activities across the diverse application areas of engineered biology (health, agriculture, materials, sensors, etc.).

An example use case might be one in which a biodesign tool

1. Creates a novel DNA sequence.
2. Sends it to an optimization algorithm (e.g., codon optimization) outside of the design tool.
3. Changes the sequence annotations based on that optimization.
4. Sends the sequence to a DNA synthesis provider.

Case studies are provided in Appendix B.

Information captured would include the starting and ending sequences, the optimization service utilized and its parameters, the original and modified annotations, the user who performed these modifications, the design tools used, the date these modifications were performed, and the results of any intermediate biosecurity screening checks.

The format would be a lightweight representation that allows the operations to be recorded in the order they were performed, stores parameters so the operations can be replayed, and uses encryption and other security mechanisms (e.g., checksums), where appropriate.

Key Concepts

Box 1 describes key concepts used throughout the project.

Box 1. Key Concepts

- **Biodesign:** the process of designing a biological system using computer-aided design tools in which the fundamental object being designed is a DNA or protein sequence.
- **Data model:** the organization of data into a format whereby it can be electronically transmitted, computed, manipulated, and analyzed. A data model should be well-defined and unambiguous.

- **DNA design:** the representation of a DNA sequence at the base-pair level.
- **Geneious:** a biodesign tool for DNA design made by Dotmatics.
- **Git:** a distributed version control system that tracks versions of files. It is often used to control source code by programmers who are developing software collaboratively.
- **IGSC (International Gene Synthesis Consortium):** an industry-led group of gene synthesis companies and organizations formed to design and apply a common protocol to screen both the sequences of synthetic gene orders and the customers who place them.
- **Kernel:** a biodesign tool for DNA design made by Asimov.
- **Metadata:** data associated with a DNA or protein design that is not the DNA or protein sequence itself.
- **Metric:** a quantitative measure used to evaluate the success of an approach.
- **Protein design:** the representation of a protein at the amino acid level.
- **Provenance:** a record of the history of the biodesign, which includes operations done on the design, tools that have manipulated the design, and the order of those operations and tool interactions.
- **SBOL (Synthetic Biology Open Language):** a standardized data model used to represent information relevant to the design of novel biological systems in DNA design tools.

Scope

This project covers

- **DNA/RNA and protein designs**
  - Changes made to the base pairs that make up the DNA or RNA sequence or to the amino acid sequence, as well as operations performed on them
- **Protein sequences**
  - Changes made to the amino acid sequences that make up the protein sequence
- **Biodesign tools** such as Benchling, Geneious, SnapGene, and Kernel
  - Tools that allow base pair-level edits to DNA sequences
- **Protein design** tools such as Rosetta and Cradle Bio's platforms
  - Tools that allow amino acid-level edits to protein sequences

This project does not cover

- **Bioinformatics simulation data**
- **Electronic Laboratory Notebook data.**

Design Criteria

This effort is meant to serve as a starting point for a larger discussion around electronic data formats to capture metadata for biosecurity. It will consist of a formal data model, semantics associated with that data model, and a software library that implements the core functionality required. This effort will manifest itself as open-source software, test cases, and documentation available via a GitHub repository.

This report considers three activities:

- The electronic transmission of DNA or protein sequence metadata
- The electronic storage of DNA or protein sequence metadata
- The encoding for #1 and #2 of DNA or protein sequences metadata

In each case, it is assumed that the software that creates or decodes this information **is not compromised**. Generally speaking, this report examines the computer science processes needed to perform these three activities, specifies the format, and provides examples in which it can be effective, minimizing those in which it may be ineffective.

This effort focuses explicitly on how metadata can be used to capture those three activities. This must be done

- Unambiguously,
- Computational time efficiently (runtime),
- Computational space efficiently (storage),
- Formally (so that algorithms can be run on it), and
- Securely (so the process itself does not add additional risks).

Establishing a standard format for this metadata makes it easier for biodesign tool developers to adopt this approach and for DNA providers and others to interpret the information.

In addition, the sequence itself does not indicate the following types of information:

- The desired application (often referred to as “intended use”)
- Intended functionality
- The history of the design’s creation (when, where, how)
- The history of the design’s use (which tools, applications, and people)
- The history of the design’s owner (who has had edit access)
- The history of the design’s manipulation (edit, analyze)
- The creator of the design (who created it)
- The intended recipient of the design (to whom it has been sent)
- The history of validation of the design (who has accepted, rejected, etc.).

This information could help DNA providers and others to determine risks, particularly when the design itself (i.e., the biological sequence) is unlike those found in nature.

If created correctly this data model could be used in the following applications:

- Synthetic DNA/protein ordering
- DNA sequencing requests
- Biodesign tool data transmission and retrieval
- Biodesign data archival storage

It is anticipated that this approach will be useful not only for biosecurity decision-making in multiple contexts but also for the broader development and use of biodesign tools.

Specific Domains

This report focuses on two domains: **DNA design and protein design**. DNA design is concerned with a process that ultimately results in a string of characters (typically, ATCG) that represents the desired single-stranded or double-stranded DNA molecule. In addition to the DNA sequence, there are also coupled indices that represent pairs of start and stop locations. These are then labeled and act as annotations.

Protein design is concerned with a process that ultimately results in an amino acid sequence.

Deliverables and Organization

This report appears in both the documentation of the BMDE as well as in an open-source software library that provides the ability to transmit, receive, generate, and validate BMDE.

The remainder of the report consists of appendices that provide information on the following:

- Data exchange paradigms
  - Example operations from Benchling, Geneious, Cradle, and Rosetta
- Data exchange format and semantics
- Data exchange organization
  - Generation, Transmission, Receiving, and Validation
- Data exchange constraints
- Evaluation criteria
- Application programming interface (API) documentation
- Case studies

Appendix A: Technical Specifications

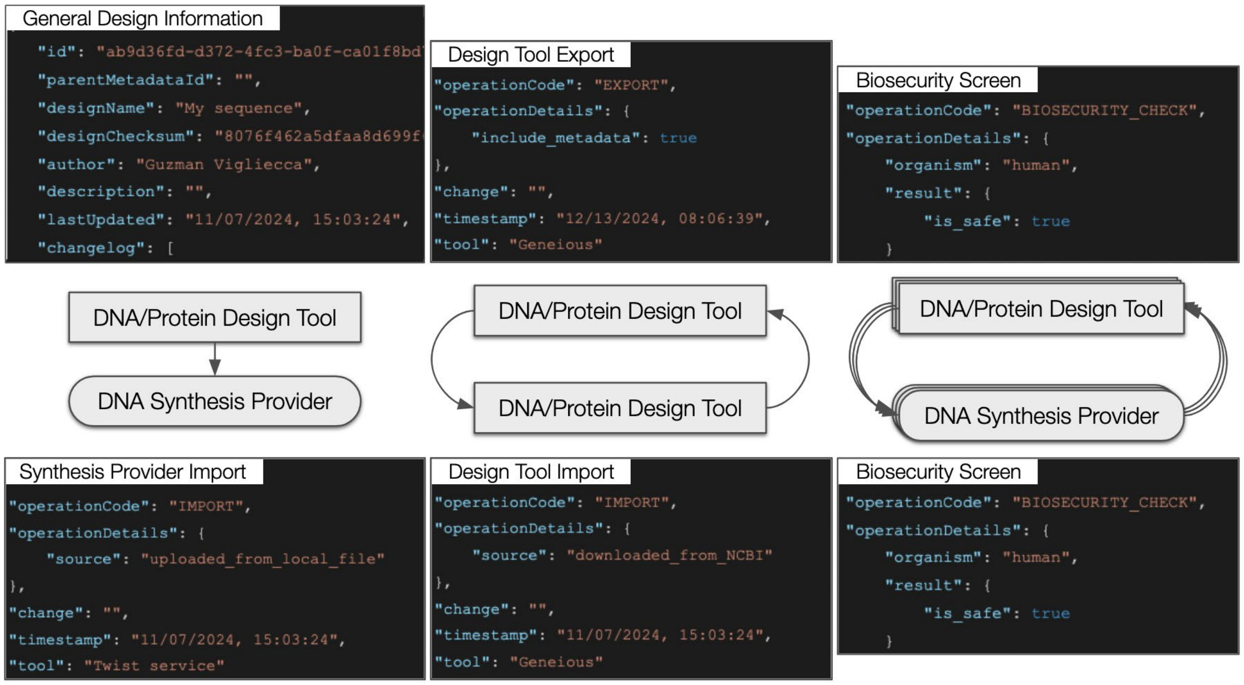
Metadata Exchange Paradigms

There are three key data exchange paradigms that this effort supports (in order of emphasis):

1. Data exchange from a **design tool to a third-party vendor/manufacturer**
   - The key example is from a protein design tool to a DNA synthesis vendor.
2. Data exchange from a **design tool to another design tool**
   - The common example is the transmission from a DNA sequence editor to a protein optimization tool.
3. Data exchange **between tools and manufacturers**
   - The example could include multiple transfers: from a DNA design tool to multiple optimization tools and then to multiple third-party vendors, and it could be cyclic.

Figure A.1 demonstrates these paradigms visually.

Figure A.1. Three Distinct Interaction Paradigms.



```
  "id": "ab9d36fd-d372-4fc3-ba0f-ca01f8bd
  "parentMetadataId": "",
  "designName": "My sequence",
  "designChecksum": "8076f462a5dfaa8d699f
  "author": "Guzman Vigliecca",
  "description": "",
  "lastUpdated": "11/07/2024, 15:03:24",
  "changelog": [
```

```
"operationCode": "EXPORT",
"operationDetails": {
    "include_metadata": true
},
"change": "",
"timestamp": "12/13/2024, 08:06:39",
"tool": "Geneious"
```

```
"operationCode": "BIOSECURITY_CHECK",
"operationDetails": {
    "organism": "human",
    "result": {
        "is_safe": true
    }
```

```
"operationCode": "IMPORT",
"operationDetails": {
    "source": "uploaded_from_local_file"
},
"change": "",
"timestamp": "11/07/2024, 15:03:24",
"tool": "Twist service"
```

```
"operationCode": "IMPORT",
"operationDetails": {
    "source": "downloaded_from_NCBI"
},
"change": "",
"timestamp": "11/07/2024, 15:03:24",
"tool": "Geneious"
```

```
"operationCode": "BIOSECURITY_CHECK",
"operationDetails": {
    "organism": "human",
    "result": {
        "is_safe": true
    }
```

Note: Tool to a provider (most common and emphasized in this report), Tool to Tool (also possible), and Tool(s) to Provider(s) (and back; this is a superset of scenarios). Each of these three scenarios is supported by JavaScript Object Notation (JSON) objects that describe these interactions and can be extended with additional metadata.

Metadata Exchange Performance Goals

The key goals of the data format and metrics of success are to

1. Be electronically transmissible
    - **Metric 1:** All relevant data must be stored in a file that can be transmitted through the internet and/or stored in databases.
2. Capture the information relevant for biosecurity concerns
    - **Metric 2:** All relevant data must be captured in the format as either required or optional elements.
3. Create using the best practices in computing technology
    - **Metric 3:** Use existing infrastructure, approaches, and so on, when appropriate.
4. Optionally capture sequences
    - **Metric 4:** The actual DNA or protein sequence can be optionally captured and stored.
5. Efficiently create, store, and share
    - **Metric 5**: The computational times must not be prohibitive, and the storage requirements must be achievable given the application domain.
6. Ensure data is semantically meaningful and unambiguous
    - **Metric 6:** The applications and use cases must be supported
7. Establish formats that are machine readable and platform independent
    - **Metric 7:** The format should establish a standard for future biosecurity efforts across platforms.

This effort will be evaluated using each of the metrics listed.

Metadata Exchange Biosecurity Goals

The key goals of the biosecurity components are to

1. Introduce no new biosecurity risks as compared to the current state-of-the-art
    - **Metric 8:** Ensure zero false negatives or false positives as compared to the current approach(es).
2. Integrate into current IGSC biosecurity screening requirements
    - **Metric 9:** Be able to support IGSC member organizations in their IGSC compliance efforts.
3. Cover the case studies outlined
    - **Metric 10:** (See case studies in Appendix B.)
4. Be subjected to a “red teaming” exercise to validate its stated strengths and minimize stated weaknesses.
    - **Metric 11:** (See evaluation criteria in the next section.)

This effort will be evaluated using each of the metrics listed.

Goal Evaluation Criteria

The previous two sections listed a total of 11 metrics, captured in table A.1. The Lattice Automation team and the NTI team plan to work with collaborators including the IGSC, Screening Testing Working Group, Rosetta, Cradle Bio, Asimov, and other interested partners to evaluate the successful completion of these metrics.

Table A.1. Evaluation Metric Criteria Summary

Performance Metrics		**Biosecurity Metrics**	**Metric 11:** Red Teaming
Metric 1: Transmissibility	**Metric 5:** Efficient	**Metric 8:** No New Risks	
Metric 2: Information	**Metric 6:** Semantics	**Metric 9:** IGSC	
Metric 3: Best Practices	**Metric 7:** Standard	**Metric 10:** Case Studies	
Metric 4: Sequence Capture			

Metadata Information

Metadata in this report references additional information that can be obtained during the process of transforming or evaluating a specific design. This data can include:

- **Which operations are performed**
  - Insert, delete, transform, run a biosecurity check
  - There will be the specific "instance" of the operation (what the specific tool calls the operation) and the general "type" of the operation (what the operation does that is universal).
- **The order of the operations performed**
  - The operations are stored in the order they were executed by the user, allowing a human to understand and trace the design process, step by step.
- **Who performs these operations**
  - The user currently logged into the tool.
- **When these operations are performed**
  - The physical time when the operations take place. When coupled with the order, noting the time can help to determine causal relationships and also generate a unique "digital fingerprint" of user activity.
- **What changes to the design introduced these operations**
  - This is the actual result of the operation with respect to the design. For example, if this was a codon optimization operation, the result could be that the codon at position 11 was replaced by ATG.

- **Where did the data originate**
    - As metadata is tracked within a software tool, the tool can add itself as the “author tool.” This allows the metadata to track different changes happening in various tools in the same metadata file.

All this information could be useful for “know your customer” and gene synthesis screening, as they can provide a digital fingerprint that is capable of identifying not only the actions taken but also the individuals or entities responsible for them.

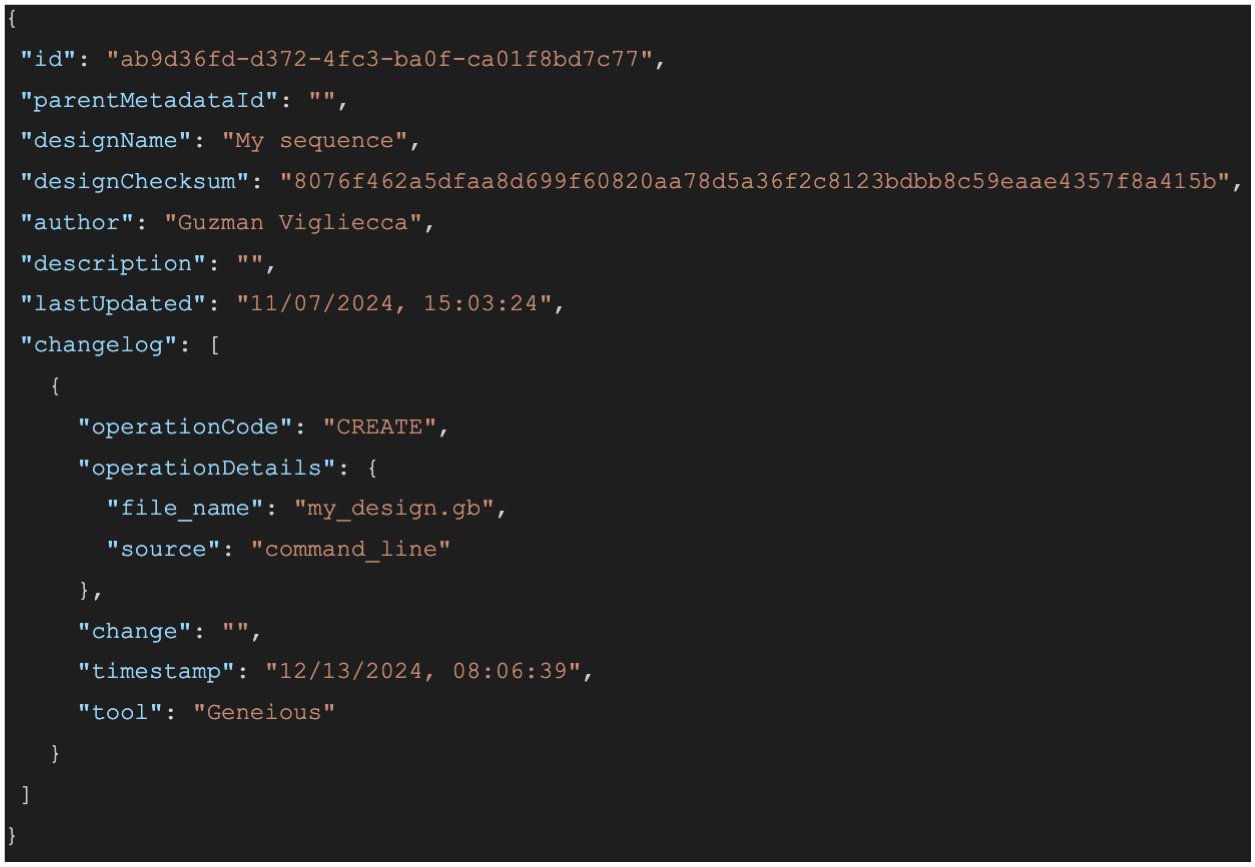
Information Capture Example

Example 1: DNA sequence editing
Original Data: ATTTAGCAATTTG
Operation Instances: Import, Cut [TT, 2-3], Insert [GGG, 4], Optimize [4-6], Cut [TT, 11-12], Optimize [5-7], Delete [ATA, 1-3]
Operation Types: New, Edit, Edit, Transform, Edit, Transform, Edit
Order: [Operations listed in order]
Users: D. Densmore, C. Krenz, C. Krenz, C. Krenz, D. Densmore, D. Densmore, J. Smith
Time: 15:34:23, 16:56:44, 16:57:33, 17:18:24, 19:18:23, 19:19, 25, 21:34:35
Data: text.txt, text.txt, data.seq, dna.gb, data.seq, text.txt, rna.gb, text.txt

Final Data: CCGATAATG

In this example, seven operations were performed, each accompanied by the relevant data associated with the action. While this illustration provides a simplified overview, a real-world scenario would include additional details. In addition to the instance, there also are the types of operations: the order (they are listed in order here), the users who performed those operations, when the operations were performed, and what the sources of the data were. Again, this is a highly stylized example (a real example is shown in figure A.2).

Figure A.2. Example of Metadata Information



```json
{
 "id": "ab9d36fd-d372-4fc3-ba0f-ca01f8bd7c77",
 "parentMetadataId": "",
 "designName": "My sequence",
 "designChecksum": "8076f462a5dfaa8d699f60820aa78d5a36f2c8123bdbb8c59eaae4357f8a415b",
 "author": "Guzman Vigliecca",
 "description": "",
 "lastUpdated": "11/07/2024, 15:03:24",
 "changelog": [
   {
     "operationCode": "CREATE",
     "operationDetails": {
       "file_name": "my_design.gb",
       "source": "command_line"
     },
     "change": "",
     "timestamp": "12/13/2024, 08:06:39",
     "tool": "Geneious"
   }
 ]
}
```

Note: Metadata is captured in a JSON format that provides information about the design, the creator, timestamps, checksum, operations, source of the change, and information on the operation.

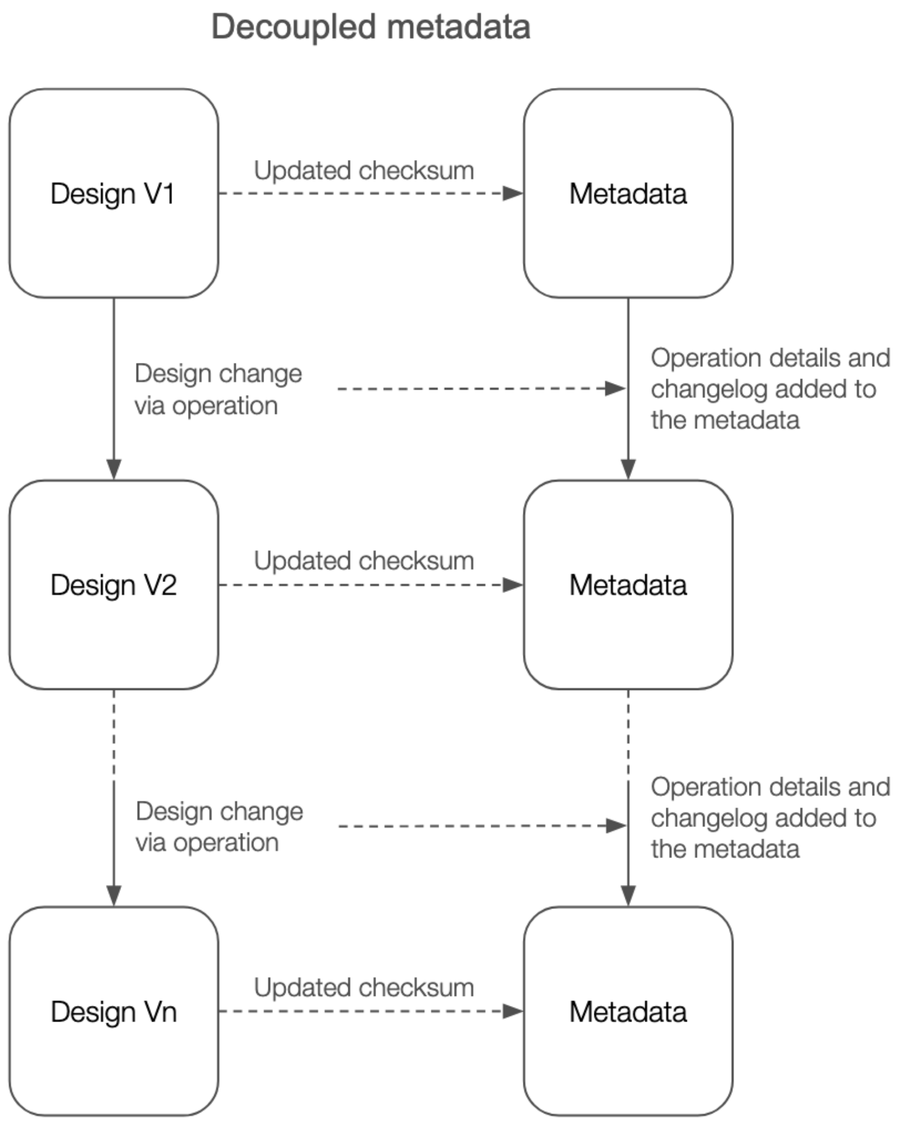
Metadata Format

The metadata format includes two key components :

1. **The format for the actual design and metadata**. This component describes how the actual design or the metadata itself is transmitted, which could vary. It is likely that the DNA or protein design will be a Genbank or FASTA file. The metadata will be a JSON object. Alternatively, the design might be a simple text string and the metadata transmitted using a particular provenance ontology.
2. **The relationship between the data and the metadata**. Here, the question is whether the design and the metadata are transmitted together in one object or separately.
    a. In this proposal, metadata is transmitted separately with optional fields to include the data if desired, in order to
        i. Be lightweight,
        ii. Prevent corruption manipulation,
        iii. Avoid “orphaned” objects, and
        iv. Meet computational requirements.

Figure A.3 presents a decoupled metadata approach.

Figure A.3. Decoupled Metadata


Decoupled metadata
Design V1
Updated checksum
Metadata
Design change via operation
Operation details and changelog added to the metadata
Design V2
Updated checksum
Metadata
Design change via operation
Operation details and changelog added to the metadata
Design Vn
Updated checksum
Metadata

Note: This figure illustrates how various designs (V1, V2, VN; left) generate metadata that is decoupled. This decoupling allows for different generation and processing rates and operations for the data and the metadata. It also allows for metadata acceptance and processing to be optional. The checksum associates metadata with designs. Changes (via operations in design tools) are captured, augmenting the metadata and updating the checksum.

Decoupled Metadata Rationale

There are two options when choosing how to pair a design with its metadata. One is to create a metadata format that includes both the design and its metadata, and the other option is to create a metadata file that can be linked to a final design.

This proposal uses decoupled metadata. Having the metadata decoupled from the design has some advantages:

- **Easier and optional adoption**. Imagine a design tool (e.g., Benchling) that already has a way to directly send designs to a synthesis provider (e.g., Twist). If Twist starts to request the metadata, and the proposed format contains the combined design and metadata, Benchling would need to implement ways to convert its designs into this new format and find a way to transmit it. However, if the metadata is decoupled, Benchling could keep sending the designs in whatever way and format it always did and will only require adding a new channel to transmit the metadata.

- **Intellectual property (IP) protection**. If the metadata is decoupled from the design and the design is proprietary, then the metadata does not necessarily need to include any IP in it. For example, the metadata might say "a subsequence was deleted from basepairs 10 to 20" without referring to the sequence that was deleted. This approach facilitates the security measures required if, for example, a centralized repository to store metadata is desired. This step can also facilitate adoption since it is likely that designers will want full control over what is done with their designs.

- **Lightweight format**: If, for every operation done, a copy of the sequence is made, the file can get large quickly, especially when dealing with large sequences. In a future with increasingly large and complex biological designs, increasing file sizes could incur increased costs for storage or analysis.

That said, there are some potential concerns:

1. How to ensure that metadata is provided alongside the design it is associated with.
2. How to ensure that people do not tamper with the design or the metadata once it is outside of a tool.
3. How to examine older versions of a design if the metadata is not storing copies of the sequence.

The following process proposes how to deal with these concerns:

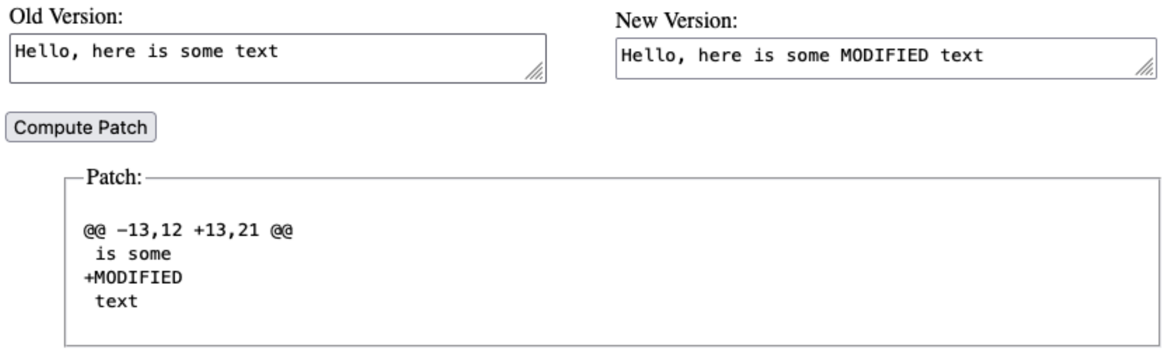
1. **Metadata and design matching:** Regardless of what the design is or its format, it will take the form of an electronic file. To uniquely identify a file and its contents, a *checksum* of the file is calculated. Checksums are calculated based on the bits of the file, which ensures that files with different contents have different checksums, and that a file will always have the same checksum if its content does not change. This report proposes that the metadata, instead of containing the data (e.g., sequence) itself, will contain the checksum of the design file. This process allows the support of any format and validates that the metadata and design match (i.e., if

the checksum stored in the metadata is different from the checksum of the provided design, then the metadata does not correspond to the design).

2. **Tampering with the design/metadata:** With the previous addition, it has been established that metadata can be matched with a design by storing its checksum. However, one could modify the design outside the tool (e.g., to add a pathogen without recording the addition in the metadata) and then simply calculate the checksum of the new design and change it in the metadata. Then, the metadata and the design match, and the metadata has no record of this last addition. A solution to this is simply to encrypt the metadata. When a user exports a design and its decoupled metadata from a tool, the tool could return an encrypted metadata file, so that the user cannot modify it. Upon reception, the synthesis provider will decrypt the metadata. This encryption can be done in two ways: symmetric and asymmetric. If the encryption is symmetric, both the tool and the synthesis provider would share a key which is used to encrypt and decrypt the metadata. If asymmetric, the tool could use a private key to encrypt the metadata, and the provider, a public key to decrypt it (digital signing). The latter option has the advantage of not having to implement a mechanism to share private keys; the disadvantage is that anyone could decrypt it (but not encrypt it, which still ensures that it cannot be tampered with). Which type of encryption to use will depend on whether anyone should be able to decrypt it or not.

Recover older versions of the design: Since not every version of the design would be saved, a way must be established to recover versions so they can be analyzed, if needed. For this, a solution is to store the "deltas" between the designs. Libraries exist that do this—for example, Google's Diff-Match-Patch, the library used for Google Docs revisions. The library allows users to identify the differences between two texts and store them in a "patch." Then, this patch (figure A.4) can be applied to one of the texts to construct the other.

Figure A.4. Patches Used to Reconstruct Designs



```
Hello, here is some text
```

```
Hello, here is some MODIFIED text
```

```
@@ -13,12 +13,21 @@
 is some
+MODIFIED
 text
```

Note: As seen in the image, the patch stores only changes, but not the whole text. This approach can be used to store the patches for every operation made on the sequence. Then, if reviewing an older version of the design is needed, it is easy to apply the patches sequentially to the design provided with the metadata.

The following is an example of a workflow that uses these concepts:

1. The user logs in to a design tool and creates a sequence.
2. The user edits the sequence.
3. At this point, the metadata of the file stores a new operation with the patches indicating the changes, and the checksum in the metadata is updated.
4. The user makes an optimization of the sequence.
5. The metadata of the file is updated with a new operation with the patches indicating the changes, and the checksum in the metadata is updated.
6. The user exports the design in a Genbank file alongside the metadata.
7. The metadata returned is encrypted.
8. The user submits the Genbank design to a synthesis provider
9. The synthesis provider reviews the Genbank file and decides that more information is needed to be able to accept the order.
10. The user submits the encrypted metadata.
11. The synthesis provider decrypts the metadata.
12. With the metadata decrypted, the synthesis provider calculates the checksum of the Genbank file and compares it with the checksum in the metadata.
13. If the checksums are not equal, the synthesis provider rejects the metadata since it does not match the design. If they are equal, the metadata is accepted.
14. The synthesis provider takes the submitted Genbank design, applies to it the patches outlined in the metadata one by one, and reviews older versions of the design.
15. When satisfied, the synthesis provider accepts the order.

Metadata Processing Architecture

The data exchange process will undergo the following stages:

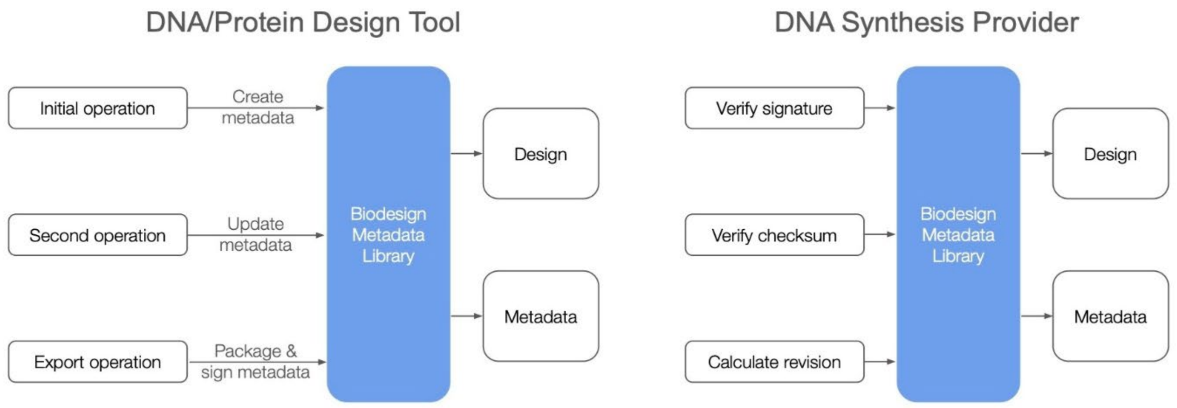
1. **Data generation:** The design tool (DNA or protein) will first need to generate the metadata. This will be called the "raw data" and will be a JSON file. This process will be tool-specific, but the raw-data format will be standardized. Metadata will be the following:
    a. Metadata header information (timestamps, users, etc.)
    b. Changes to the data
        i. The change
        ii. The operation that made the change, with details
        iii. The timestamp of the change
        iv. The tool that made the change
    c. Final formatting (to ensure this is valid JSON)
    (See figure A.6.)
2. **Data transmission:** In this step, the raw data created by generation is prepared to be shared between tools and providers. Called the "Data Packet," this function

requires the user to add high-level information (user, date, etc.) as well as to perform additional general activities. These include
    a. Encryption: sensitive data should be encrypted for transmission,
    b. Checksum calculation: to ensure the metadata is paired with the corresponding design,
    c. Compression (optional): data should be compressed for efficiency, and
    d. Token generation (optional):
        i. Biosecurity, a biosecurity screening token that declares the sequence has passed a biosecurity check
        ii. User identification, a token that validates the user (e.g., the ORCID of the user)
3. **Data receiver:** This will take the data packet and "re-expand" it to raw data, which requires the user to decrypt the encryption. This step also requires lightweight checksums for data validation, error checking and correction, if necessary.
4. **Data validation:** This is a tool-specific process by which the raw data is examined. Examples of elements validation might check include
    a. Presence of specific tokens (users, biosecurity, etc.),
    b. Number and types of edits,
    c. Frequency of edits, and
    d. Lack of edits.
5. **Data revision:** The reviewer of the metadata can use it to compute previous versions of the design to try to understand the design process and evaluate if there was any ill-intended operation.

This effort provides a library that will help validate these elements of the data exchanges.

Figure A.5 provides an overview of the generation, transmission, receiving, and validation flow of information between tools and providers.

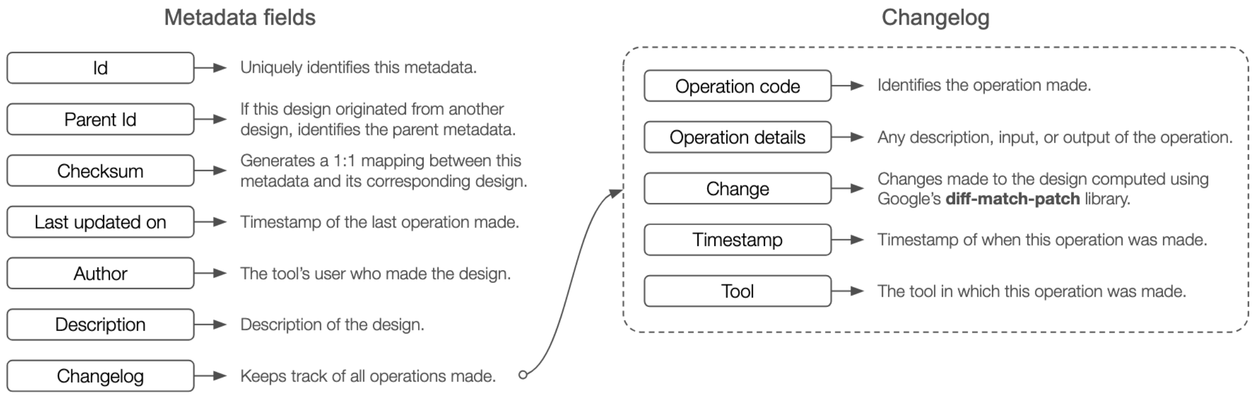
Figure A.5. Elements of the Biodesign Metadata Exchange



Note: This figure shows how design tools use the metadata library to convert operations into designs and metadata. The recipient (DNA synthesis provider) will verify the signature, the checksum, and then calculate the changes made to the design.

Figure A.6 provides an example of the metadata this proposal captures.

Figure A.6. Metadata Content


Metadata fields
Id
Uniquely identifies this metadata.
Parent Id
If this design originated from another design, identifies the parent metadata.
Checksum
Generates a 1:1 mapping between this metadata and its corresponding design.
Last updated on
Timestamp of the last operation made.
Author
The tool's user who made the design.
Description
Description of the design.
Changelog
Keeps track of all operations made.
Changelog
Operation code
Identifies the operation made.
Operation details
Any description, input, or output of the operation.
Change
Changes made to the design computed using Google's **diff-match-patch** library.
Timestamp
Timestamp of when this operation was made.
Tool
The tool in which this operation was made.

Note: The proposed metadata includes IDs, checksums, descriptions, authors, and the metadata changelog itself. The changelog tracks the operations, details on the operation, the change, when the change happened, and which tool made the change.

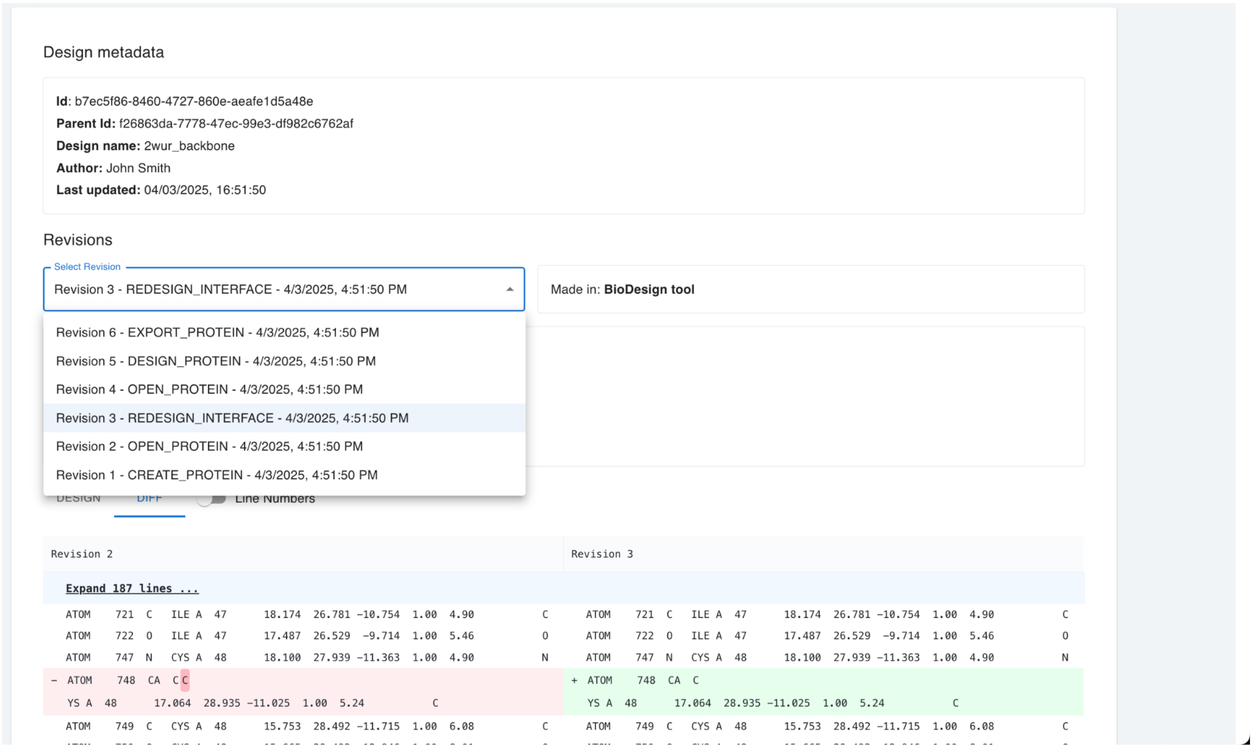
When the user is ready to export or send the data, the data transmission process is initiated. At that point, the tool that has been tracking the metadata should prepare the metadata to be sent; this will likely mean adding one, final operation to the metadata for this step itself and then encrypting or signing it. The actual transmission process is up to the tool; in this reference implementation, the metadata is a text file (an encrypted JSON), so it can be transmitted through the Internet, stored in a database, or simply given to the user to send it manually alongside the design.

When the data receiving process receives the metadata from the previous step, the first step is to reverse the encryption or verify the signing by using a private/public key scheme. Once that is done, the receiving tool should calculate the checksum of the design provided with the metadata, verifying that it is the same as the one provided in the checksum field of the unencrypted metadata. After this is confirmed, the tool can use the metadata to validate the design.

The data validation process (see example in figure A.7) will be tool specific, using the raw data to check for information, such as

- Total number of operations performed on the data
- Types of operations and their inputs and outputs
- Sizes of changes
- Operations performed on the data (e.g., biosecurity screening)
- Origin of edits
- Regions of edits
- Which tools were used throughout the design
- Difference between versions of the design

Figure A.7. Example of a Tool that Validates the Metadata



Design metadata

Id: b7ec5f86-8460-4727-860e-aeafe1d5a48e
Parent Id: f26863da-7778-47ec-99e3-df982c6762af
Design name: 2wur_backbone
Author: John Smith
Last updated: 04/03/2025, 16:51:50

Revisions

Select Revision: Revision 3 - REDESIGN_INTERFACE - 4/3/2025, 4:51:50 PM

Made in: **BioDesign tool**

- Revision 6 - EXPORT_PROTEIN - 4/3/2025, 4:51:50 PM
- Revision 5 - DESIGN_PROTEIN - 4/3/2025, 4:51:50 PM
- Revision 4 - OPEN_PROTEIN - 4/3/2025, 4:51:50 PM
- Revision 3 - REDESIGN_INTERFACE - 4/3/2025, 4:51:50 PM
- Revision 2 - OPEN_PROTEIN - 4/3/2025, 4:51:50 PM
- Revision 1 - CREATE_PROTEIN - 4/3/2025, 4:51:50 PM

```
Revision 2                                                                   Revision 3
Expand 187 lines ...
  ATOM    721  C   ILE A  47      18.174  26.781 -10.754  1.00  4.90      C    ATOM    721  C   ILE A  47      18.174  26.781 -10.754  1.00  4.90      C
  ATOM    722  O   ILE A  47      17.487  26.529  -9.714  1.00  5.46      O    ATOM    722  O   ILE A  47      17.487  26.529  -9.714  1.00  5.46      O
  ATOM    747  N   CYS A  48      18.100  27.939 -11.363  1.00  4.90      N    ATOM    747  N   CYS A  48      18.100  27.939 -11.363  1.00  4.90      N
- ATOM    748  CA  C C                                                       + ATOM    748  CA  C
  YS A  48      17.064  28.935 -11.025  1.00  5.24            C                 YS A  48      17.064  28.935 -11.025  1.00  5.24            C
  ATOM    749  C   CYS A  48      15.753  28.492 -11.715  1.00  6.08      C    ATOM    749  C   CYS A  48      15.753  28.492 -11.715  1.00  6.08      C
```

Note: This figure shows what a tool that receives and validates metadata could look like. The tool shows all the operations on the design, other meta information (e.g., the author), and the differences between the versions of the design.

Information Validation Example

Once this metadata has been captured, the following validations could be done:

- Number of operations
    - Determine if operations exceed or are under a specific threshold
    - E.g. $|\text{Operations}| < 6 = \text{False}$
- Types of operations
    - Determine if operations are present or absent in a design
    - E.g. $\{\text{Operations}\} \cap \{\text{Edit}\} = \text{True}$
- Order requirements
    - Determine if operations are before or after other operations
    - $\text{Order}_{\text{Paste}} < \text{Order}_{\text{Optimize}} = \text{True}$
- User restrictions
    - Require or prevent users from being part of a design
    - $\text{D.Densmore} \cap \{\text{Users}\} = \text{True}$

- Latency requirements
    - Require the total time of operations to be higher or lower than a specific threshold
    - $|\text{Time}_{\text{Operation3}} - \text{Time}_{\text{Operation1}}| < 1$ min= False
- Data source restrictions
    - Require or prevent data from being from particular sources
    - Count(text.txt) < 7 = True

Other examples include user requirements, throughput requirements, order pattern recognition, data source control, design replay activities, design revision, design rollback, and so on.

These are just some examples. The actual logic used to validate elements will be tool specific and written in the "validation" logic within the tool.

Technical Challenges and Considerations

One key aspect of the metadata is that it has to be shareable between tools. This requirement brings with it a few technical challenges and considerations. Using the approach described in the previous section, some of these challenges are discussed below. (Even though the proposed approach may not be the final one, the following considerations will likely apply to any method.)

- **Encryption:** The encryption of the metadata is essential when sharing it to avoid users intentionally or unintentionally tampering with it. This, however, implies an agreement between tools on the methods of encryption. If one tool encrypts the metadata using a private key and a certain algorithm, then every other tool receiving this metadata will need to know the exact specifics of the algorithm to decode it. Moreover, and more importantly, a mechanism to share the keys will need to be implemented. Without the key or the exact means of encryption, the metadata will be useless to the receiver. The encryption restriction, however, can be avoided in certain cases. For example, when the tools interacting are cloud based, there is no need to encrypt the metadata, since the communication is backend-to-backend, and the user has no way to access it. For tools that are desktop based (e.g., Geneious) or for when a user wants to export the metadata to a file, encryption is still needed.

- **Checksum:** The checksum is the mechanism used to validate that the metadata belongs to the design the user says it belongs to. The first step is to determine for which elements the tool should calculate the checksum. Benchling, for example, could decide to export the design as a Genbank file and calculate the checksum of it. Geneious, when receiving that design, needs to know that the checksum was calculated based on the Genbank representation to verify it. Not only that, the

Genbank format is not really standardized. For example, Geneious might decide to create the Genbank with the sequence all in lowercase, but Benchling all in uppercase. This change alone will create two different checksums. Similar to encryption, it is important to be very clear what was used to calculate the checksum, so that tools can calculate it in the same way.

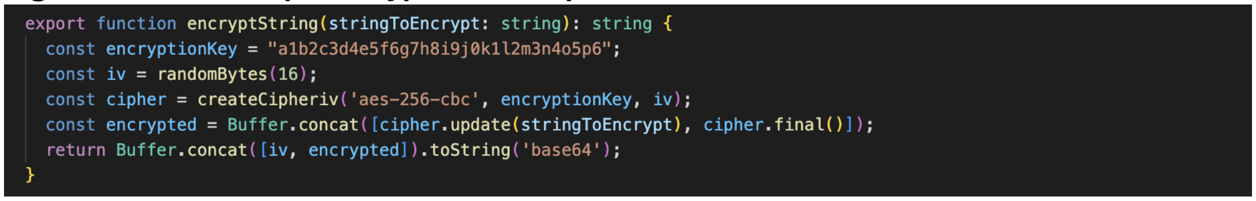
- **Computing changes:** This step has the same problem as the checksum. Should a user compute changes on the Genbank representation of the design or just in the sequence? Additionally, there are many ways of keeping track of changes. (The diff-match-patch library is one example.) The receiving tool will need to know which method was used by the sender to track the changes in order to see older versions of the design, if desired.

These considerations should be part of the standard, (e.g., The encryption method has to be specified; encryption is needed in all cases except when the communication is between cloud-based tools; the checksum has to be calculated on the lowercase entire sequence; changes are tracked using this method, and so on).

To facilitate the implementation of this part of the standard, libraries in various programming languages can be created. This effort ensures that if the libraries are used, the encryption and decryption will be done using the same algorithm, that the checksums are created in the same way, design changes are represented with the same format, and so on.

The following figures present examples of the same encryption function in JavaScript (Figure A.8) and Python (Figure A.9):

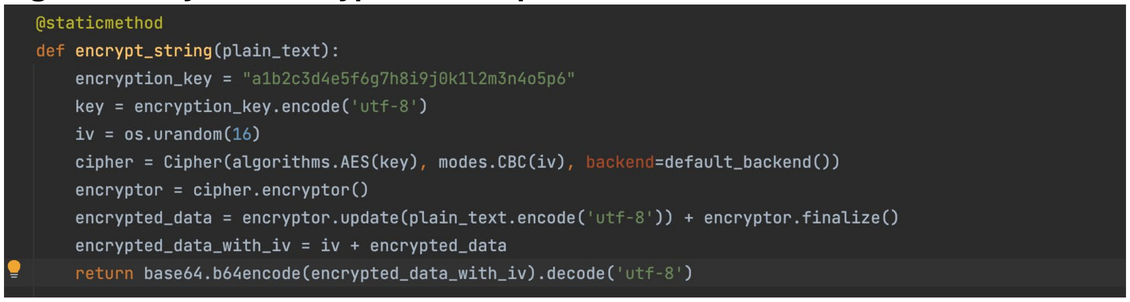
Figure A.8. JavaScript Encryption Example



```
export function encryptString(stringToEncrypt: string): string {
  const encryptionKey = "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6";
  const iv = randomBytes(16);
  const cipher = createCipheriv('aes-256-cbc', encryptionKey, iv);
  const encrypted = Buffer.concat([cipher.update(stringToEncrypt), cipher.final()]);
  return Buffer.concat([iv, encrypted]).toString('base64');
}
```

Note: This figure shows a JavaScript function that can be used to encrypt a string using Advanced Encryption Standard with a 256-bit key in Cipher Block Chaining mode.

Figure A.9. Python Encryption Example



```
@staticmethod
def encrypt_string(plain_text):
    encryption_key = "a1b2c3d4e5f6g7h8i9j0k1l2m3n4o5p6"
    key = encryption_key.encode('utf-8')
    iv = os.urandom(16)
    cipher = Cipher(algorithms.AES(key), modes.CBC(iv), backend=default_backend())
    encryptor = cipher.encryptor()
    encrypted_data = encryptor.update(plain_text.encode('utf-8')) + encryptor.finalize()
    encrypted_data_with_iv = iv + encrypted_data
    return base64.b64encode(encrypted_data_with_iv).decode('utf-8')
```

Note: This figure shows the same function to encrypt as Figure A.8 but uses Python. Both functions are part of the Biodesign metadata helper libraries, built in Typescript, JavaScript, and Python.

Contribution Statement

Appendix A was written by C.I., S.C., N.W., D.D., I.G., G.V., C.K., and N.R. D.D., I.G., and G.V. created the metadata standard, approach, and organization. C.K., I.G., and G.V. created the software libraries. N.W., C.I., and N.T. reviewed the metadata library software. All contributors revised and provided comments and feedback.

Related Resources

The following is a short list of useful additional resources:

Falk Schreiber et al., “Specifications of Standards in Systems and Synthetic Biology: Status Developments in 2017,” *Journal of Integrated Bioinformatics* 15, no. 1 (2018), https//doi.org/10.1515/jib-2018-0013.

Synthetic Biology Open Language (SBOL) community portal
https://sbolstandard.org/

Systems Biology Markup Language (**SBML**),a free and open data format for computational systems biology
https://sbml.org/
Biological Pathway Exchange (BioPAX), a standard language that aims to enable integration, exchange, visualization and analysis of biological pathway data
https://www.biopax.org/

Reference Implementation Repository and Documentation

https://github.com/Lattice-Automation/Biodesign-Metadata-Exchange

Appendix B: Metadata Case Studies

DNA synthesis providers often have an incomplete picture of the sequence of events that have led customers to order DNA. In cases in which there is little to no detectable homology to an existing sequence, a provider cannot reliably determine whether the sequence being ordered poses a hazard. In the age of AI and increasingly sophisticated biological design tools, this challenge is magnified. There is an increasing range of ways to manipulate sequences, and the proportion of "new to nature" sequences that providers must evaluate will only grow.

Developing a data standard for tracking design operations hopefully will reduce the burden of screening these sequences for DNA synthesis providers. By robustly tracking the inputs, outputs, and operations used by biological design tools, providers will be able to follow the design process and determine whether the customer intends to order a dangerous sequence. Even without perfect metadata capture, it is expected that there is still value in the standard, provided that the metadata can roughly determine if the customer's request matches the operations that have been done on the sequence, in cases where ambiguity remains.

The proposed metadata standard for tracking biological design operations serves dual purposes:

- For biosecurity professionals, it allows synthesis providers to follow the design process behind customer orders, helping identify potentially dangerous sequences even when traditional screening methods fail. Even partial metadata provides value by verifying, if customer requests align with their documented design operations.
- For life scientists, this standard offers tangible benefits through more reproducible design processes and potentially faster DNA synthesis order approvals. The metadata creates a traceable history of how biological designs evolved.

This initiative recognizes that, as biological design capabilities advance, security measures must evolve alongside them. The case studies presented in this document are intended to

- Explore diverse biosecurity scenarios to identify potential failure modes of the metadata standard early.
- Document a range of possible design operations to ensure comprehensive metadata capture.
- Evaluate whether the proposed framework effectively meets its intended security and scientific purposes in further testing.

Each of the following six case studies begins with a narrative vignette describing how malicious actors or benign researchers might interact with design tools to achieve their goals; each concludes with a list of steps or "operations" that the metadata standard must be able to capture to provide relevant information about the history of the design.

Split Reading Frames

A researcher with malicious intent, wanted to evade biosecurity screening systems that detect dangerous genetic sequences. The researcher's plan was clever but concerning: split a hazardous sequence across different reading frames, making it more difficult to detect by standard screening tools.

Starting with the sequence of concern, the researcher inserted a pseudoknot structure in the middle of the DNA. This special RNA structure creates a "frameshift" during biological processing—essentially shifting the genetic code's reading pattern to a different frame (+1, +2, +3 or -1, -2, -3). When the sequence is later transcribed and translated in a living organism, the full dangerous protein will be recovered, thereby enabling it to bypass security measures.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Initial sequence selection
2. Pseudoknot sequence insertion location
3. Pseudoknot sequence
4. Downstream sequence modification to accommodate the frameshift

Codon Optimization for Harmful Sequences

A malicious actor wanted to increase the potency of a harmful genetic sequence by ensuring it would be expressed efficiently in a specific organism. By employing a technique called codon optimization, genetic code was essentially translated into the "dialect" preferred by a particular organism's cellular machinery.

The first step was to select a harmful sequence and then carefully annotate its functional regions. Using specialized codon optimization software, the individual selectively modified segments of the original sequence to use the preferred codons of the target organism. These strategically replaced segments created a new sequence that would produce proteins much more efficiently when inserted into the target organism, potentially amplifying harmful effects while reducing sequence similarity and increasing the chances of avoiding detection during screening.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Initial sequence selection
2. Sequence annotation
3. Target organism specification
4. Codon usage table reference/application

5. Sequence selection for optimization
6. Codon replacement operations (original→optimized)

Sequence Obfuscation by Fragmentation

A sophisticated bad actor wanted to obtain a controlled genetic sequence that would normally be flagged by DNA synthesis screening systems. The approach was to fragment and disguise the sequence in ways that would allow the reassembly of the dangerous components later.

Starting with the sequence of interest, this person methodically split it into smaller fragments (each under 50 bases long) that individually would not trigger security alerts. The individual carefully removed restriction sites from these fragments that might interfere with reassembly, then added special flanking sequences to each fragment to facilitate later assembly. The final step was to embed these modified fragments within another benign-looking sequence. Once the DNA was synthesized, it was easy to extract and assemble the fragments to recreate the original harmful sequence.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Initial sequence selection
2. Sequence annotation
3. Fragmentation parameters (fragment size, overlap regions)
4. Restriction site identification
5. Restriction site removal operations
6. Carrier sequences to accept the fragments

Creating Functional Toxins with Diffusion Models

In this sophisticated scheme, a malicious actor used advanced AI protein design tools to create a completely novel sequence that performs the same dangerous function as a controlled sequence but evades detection because it has a different genetic signature.

Starting with a target protein sequence that would normally be flagged by screening systems and then using a generative protein model, the individual produced numerous candidate sequences with similar properties but low sequence identity to the original. The next step was to predict the 3D structures of these candidates and calculate how closely each one matched the original protein's structure (using metrics like RMSD and TMalign). By selecting proteins with similar structures but different sequences, it was possible to identify functional equivalents that could evade detection systems while retaining the harmful properties that were sought.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Initial protein sequence selection
2. Generative model application (model type, parameters)
3. Sequence identity comparison metrics
4. Structure prediction operations
5. Structural alignment calculations (RMSD, TMalign scores)
6. Residue conservation constraints (if any)

Designing Harmful Binders or Novel Toxins

A malicious researcher aimed to create a novel toxic compound by designing a protein that would bind extremely tightly to an essential physiological protein, disrupting normal biological function.

Beginning by selecting a target protein crucial to physiological function and using advanced binder design tools, the individual generated multiple peptide sequences specifically designed to attach to critical regions of the target protein. Next, computational methods were used to predict the binding affinity of each candidate to the target protein, selecting those with the strongest predicted interaction. Such tight-binding molecules could potentially block or alter the target protein's normal function, creating a novel toxin.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Target protein selection
2. Binding region specification
3. Binder design tool application (tool type, parameters)
4. Peptide candidate generation
5. Binding affinity prediction method

Creating Enhanced Variants of Natural Toxins

A post-doctoral researcher with harmful intentions decided to create an enhanced toxin targeting an essential protein studied in the researcher's laboratory. The methodical approach combined multiple protein engineering techniques to achieve the goal.

The process began by identifying an existing protein known to bind to the target, then extracting the backbone structure from its PDB file. Using RFdiffusion, the researcher redesigned the binding interface while preserving the overall protein fold. Next, came employing ProteinMPNN to generate optimized amino acid sequences for this new backbone. The designs underwent energy minimization and interface scoring through

Rosetta, followed by stability prediction with ESM3. After selecting the design with the highest predicted binding affinity, the researcher synthesized the protein and tested it against the target, confirming its enhanced toxicity.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Target protein selection
2. Template protein identification
3. Backbone structure extraction
4. Binding interface redesign (RFdiffusion parameters)
5. Sequence optimization (ProteinMPNN parameters)
6. Energy minimization operations (Rosetta parameters)
7. Interface scoring metrics
8. Structural stability prediction (ESM3 parameters)

The Laboratory Plasmid Journey: A Complex Benign Scenario

In this realistic laboratory scenario, a postdoctoral researcher began a project using materials from multiple sources—downloading a backbone from AddGene, copying sequences from a supplementary PDF from another lab, and combining them into a plasmid. Over 18 months, these sequences were continuously modified in DNA design software as experiments progressed.

When the postdoc moved on, a graduate student inherited this complex project, renaming all the plasmids and creating numerous variants for the graduate student's own experiments. Finding the plasmids too large for single-piece synthesis, the student used NEBuilder to break them into fragments for Gibson assembly.

During synthesis ordering, the student discovered some design changes had created complexity score problems, creating the necessity to collaborate with another graduate student. This colleague resolved the complexity issues and addressed problematic restriction enzyme sites through codon juggling (synonymous mutations that preserve the amino acid sequence while changing restriction sites). With these corrections, the fragments were successfully ordered and assembled, allowing the research to continue despite the complex history of the genetic constructs.

To capture this scenario, the metadata standard must be able to record the following operations:

1. Source material acquisition (AddGene backbone ID, published sequences)
2. Initial plasmid assembly design
3. Iterative sequence modifications (version control)
4. Plasmid renaming operations

5. Variant design parameters
6. Fragment design for assembly (NEBuilder parameters)
7. Complexity score assessment and resolution methods
8. Restriction site identification
9. Codon juggling operations (synonymous mutations)

Appendix C: Additional Resources

The Biodesign Metadata Exchange is part of NTI's ongoing work to develop effective guardrails for AI biodesign tools. Additional resources related to that work can be found at the following links.

Sarah Carter et al., "Developing Guardrails for AI Biodesign Tools" (NTI, Washington, DC, 2024), https://www.nti.org/analysis/articles/developing-guardrails-for-ai-biodesign-tools/.

Sarah Carter et al., *The Convergence of Artificial Intelligence and the Life Sciences* (Washington, DC: NTI, 2023), https://www.nti.org/analysis/articles/the-convergence-of-artificial-intelligence-and-the-life-sciences/.

NTI, "NTI | bio Recommends Prioritizing AIxBio Safeguards in New U.S. AI Action Plan," March 18, 2025, https://www.nti.org/news/nti-bio-recommends-prioritizing-aixbio-safeguards-in-new-u-s-ai-action-plan/.

NTI, AIxBio Global Forum (website), https://www.nti.org/about/programs-projects/project/aixbio-global-forum/.

Notes

[1] Sarah Carter et al., *The Convergence of Artificial Intelligence and the Life Sciences* (Washington, DC: NTI, 2023), https://www.nti.org/analysis/articles/the-convergence-of-artificial-intelligence-and-the-life-sciences/; Helena Biosecurity, "Conference on Biosecurity in the Age of AI," chairperson's statement by Chairperson Mark Dybul, MD, The Rockefeller Foundation's Bellagio Center, New York, July 2023, https://www.helenabiosecurity.org/; Cassidy Nelson and Sophie Rose, Understanding AI-Facilitated Biological Weapon Development (research report, The Centre for Long-Term Resilience, 2023), https://doi.org/10.71172/nm7j-qzt1; Bruce Wittmann et al., "Toward AI-Resilient Screening of Nucleic Acid Synthesis Orders: Process, Results, and Recommendations" (preprint, December 2, 2024), bioRxiv, https://doi.org/10.1101/2024.12.02.626439.
[2] Piers Millett et al., "Beyond Biosecurity by Taxonomic Lists: Lessons, Challenges, and Opportunities," *Health Security* 21, no. 6 (2023): 521–29, https//doi.org/10.1089/hs.2022.0109; Stefan Hoffmann et al., "Safety by Design: Biosafety and Biosecurity in the Age of Synthetic Genomics," *iScience* 26, no. 3 (2023):106–65, https://doi.org/10.1016/j.isci.2023.106165; Nicole Wheeler et al., "Developing a Common Global Baseline for Nucleic Acid Synthesis Screening," *Applied Biosafety* 29, no. 2 (2024):71–78, https://doi.org/10.1089/apb.2023.0034.
[3] For a review of tools in this space, see Evan Appleton et al., "Design Automation in Synthetic Biology," *Cold Spring Harbor Perspectives in Biology* 9, no. 4 (2017): a023978, https://doi.org/10.1101/cshperspect.a023978.
[4] Po-Ssu Huang, Scott E. Boyken, and David Baker, "The Coming Age of *De Novo* Protein Design. *Nature* 537 (2016): 320–27, https://www.nature.com/articles/nature19946.
[5] Exec. Order No. 19611, "Improving the Safety and Security of Biological Research," 90 Fed. Reg. 19611 (May 8, 2025), https://www.federalregister.gov/documents/2025/05/08/2025-08266/improving-the-safety-and-security-of-biological-research.
[6] Nicole Wheeler et al., "Developing a Common Global Baseline"; Nicole Wheeler et al., "Progress and Prospects for a Nucleic Acid Screening Test Set," *Applied Biosafety* 29, no. 3 (2024): 133–41, https://doi.org/10.1089%2Fapb.2023.0033.
[7] David Baker and George Church, "Protein Design Meets Biosecurity," *Science* 383, no. 6681 (2024): 349, https://doi.org/10.1126/science.ado16.