**Naval Research Laboratory**
Washington, DC 20375-5320

**NRL/6910/MR—2023/3**

# Graphical User Interface for Novel Protein Generation using ProtGPT: Version 1

JEROME ANTHONY E. ALVAREZ

*STEM Student Employment Program (SSEP)*
*Center for Bio/Molecular Science and Engineering Division*

SCOTT N. DEAN

*Laboratory for Bio/Nano Science and Technology Branch*
*Center for Bio/Molecular Science and Engineering Division*

August 23, 2023

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| **1. REPORT DATE** *(DD-MM-YYYY)* | **2. REPORT TYPE** | **3. DATES COVERED** *(From - To)* |
|---|---|---|
| 23-08-2023 | NRL Memorandum Report | 04/01/2022 – 08/09/2023 |

**4. TITLE AND SUBTITLE**

Graphical User Interface for Novel Protein Generation using ProtGPT: Version 1

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**

Jerome Anthony E. Alvarez and Scott N. Dean

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
1S67

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Naval Research Laboratory
4555 Overlook Avenue, SW
Washington, DC 20375-5320

**8. PERFORMING ORGANIZATION REPORT NUMBER**

NRL/6910/MR--2023/3

**9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Defense Threat Reduction Agency
8725 John J Kingman Rd Ste 6201
Fort Belvoir, VA 22060

**10. SPONSOR / MONITOR'S ACRONYM(S)**

DTRA

**11. SPONSOR / MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION / AVAILABILITY STATEMENT**

**DISTRIBUTION STATEMENT A:** Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**

This report details the Year 4 efforts of the Naval Research Laboratory (NRL) toward the algorithm development for optimization of biologic medical countermeasures. Machine/Deep Learning (ML/DL) techniques enable directed modification of antibody proteins and the foundational implications of redesigning and synthesizing small molecules with novel properties are further developed through efficient biomolecular sequence generation. Specifically, recent advances in the production of novel single-domain antibodies (sdAbs) is mainly motivated by the emergence of antibiotic-resistant bacteria which poses a perpetual challenge in antibiotic discovery. To aid in this task, we developed a graphical user interface (GUI) using an open-source, interactive application called Shiny through the R package. This application prototype (version 1) incorporates a generative pre-trained transformer (GPT) model for sdAb generation and enables the user to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing sdAb dataset. Summary statistics of the input and output dataset, exploratory analyses, and the proteins' physicochemical characteristics are also provided. Several validation measures are also included along the sequence generation such as redundancies, antibody numbering annotations, and proper sdAb characteristics. This automated process will allow for the generation of novel antibody sequences that can subsequently be evaluated for improvements to a specifically desired pharmacokinetic profile.

**15. SUBJECT TERMS**

Antibodies
Proteins
Graphical user interface
Generative deep learning
Generative pre-trained transformer

| **16. SECURITY CLASSIFICATION OF:** | | | **17. LIMITATION OF ABSTRACT** | **18. NUMBER OF PAGES** | **19a. NAME OF RESPONSIBLE PERSON** |
|---|---|---|---|---|---|
| **a. REPORT** | **b. ABSTRACT** | **c. THIS PAGE** | | | Scott N. Dean |
| U | U | U | U | 21 | **19b. TELEPHONE NUMBER** *(include area code)* (703) 587-4188 |



**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**

This page intentionally left blank.

## Executive Summary

This report details the Year 4 efforts of the Naval Research Laboratory (NRL) toward the algorithm development for optimization of biologic medical countermeasures. Machine/Deep Learning (ML/DL) techniques enable directed modification of antibody proteins and the foundational implications of redesigning and synthesizing small molecules with novel properties are further developed through efficient biomolecular sequence generation. Specifically, recent advances in the production of novel single-domain antibodies (sdAbs) is mainly motivated by the emergence of antibiotic-resistant bacteria which poses a perpetual challenge in antibiotic discovery. To aid in this task, we developed a graphical user interface (GUI) using an open-source, interactive application called *Shiny* through the R package. This application prototype (version 1) incorporates a generative pre-trained transformer (GPT) model for sdAb generation and enables the user to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing sdAb dataset. Summary statistics of the input and output dataset, exploratory analyses, and the proteins' physicochemical characteristics are also provided. Several validation measures are also included along the sequence generation such as redundancies, antibody numbering annotations, and proper sdAb characteristics. This automated process will allow for the generation of novel antibody sequences that can subsequently be evaluated for improvements to a specifically desired pharmacokinetic profile.

This page intentionally left blank.

## 1. Introduction

In recent technological advances, commercial production of biomolecules such as antibodies and other peptides in conducted at an accelerated scale for medical applications or therapeutics. Moreover, the prominent interest in protein sequence generation for biomolecule production has its basis in their potentially vast range of applications such as antibody-based therapeutics and peptide antibiotics. In contrast, bacteria and fungi's ability to develop an adaptation to the drugs that are supposed to eliminate them is a growing concern due to a multitude of reasons such as unregulated prescription, extensive agricultural use, inappropriate prescribing, and abuse [1]. When new antibiotic agents are eventually used, the emergence of its resistant counterpart is unpreventable. However, since bacterial evolution is uncertain, the timeline for the development of antibiotic resistance is also unpredictable [2].

On the other hand, machine and/or deep learning (ML/DL) techniques for protein sequence generation are typically comprised of systematic algorithms that are typically "*trained*" on existing protein sequences rather than being reprogrammed. These nearly automated systems can process small or large data on which to be trained and generate assumptions through a latent space. Furthermore, DL relies on a series of stacked algorithmic layers which closely resemble the architecture and functionality of the brain and the stacked layers called "*artificial neural networks*" rely on many layers of nonlinear processing units for learning data representations. This architecture has been proven useful for molecular prediction algorithm training on the large datasets comprised of millions of chemical structures. For example, a DL algorithm was previously used for antibiotic discovery – it performed predictions on multiple chemical libraries and discovered a molecule from the Drug Repurposing Hub that is structurally divergent from conventional antibiotics. These predictions led into a discovery of *Halicin* which exhibits bactericidal effect against a wide phylogenetic spectrum of pathogens including *Mycobacterium* tuberculosis and carbapenem-resistant *Enterobacteriaceae* [3]. A DL algorithm was also used for antimicrobial design where antimicrobial peptide generation was entirely based on a learned latent space from peptide sequences with corresponding experimental minimum inhibitory concentrations alone [4]. While such approaches continue to develop, it is expected that DL systems will continue to contribute to antibiotic arsenals in the design and evolution of biomolecules.

In order to aid in these incredible successes and challenges, however, generated biomolecules must be appropriately engineered to exhibit pharmacokinetic properties that make them feasible to both production and application. Prospectively, in the generation of new antibody sequences, an extrapolated process should generally be required upon the production of variants that may evade previously developed antibiotic resistance and among other attributable biophysical properties. The fast and efficient generation of antibodies, specifically single-domain antibodies (sdAbs), should be readily accessible at one's disposal to ensure availability of novel proteins for medical countermeasures. Therefore, we develop a graphical user interface (GUI) for antibody-based drug design using a modified generative pre-trained transformer (GPT) model stemmed from a recently developed artificial intelligence (AI) algorithm by Hugging Face AI Community (https://huggingface.co/). This application prototype (version 1) incorporates a generative pre-trained transformer (GPT) model for sdAb generation and enables the user to follow a set of cascaded steps for the input, exploration, and generation of new sequences based on an existing sdAb dataset.

Manuscript approved August 23, 2023.

## 2. Materials and Methods

### *2.1 Datasets*

In-house, curated single-domain antibody sequences (sdAbs) from multiple studies were used as an input data for this prototype. This sdAb database consists of unique 565 antibodies with corresponding sequence IDs, organism source, experimentally measured melting temperatures, target protein, framework (FR) and complementarity-determining regions (CDR), and digital object identifier source information. All sequences were restricted to the 20 natural amino acids and sequences with non-conventional residues were excluded.

### *2.2 Protein Generative Pre-trained Transformer (ProtGPT) Model*

A modified GPT2 transformer model from a recently published paper by Ferruz et al [5] through the Hugging Face Artificial Intelligence Group (https://huggingface.co/nferruz/ProtGPT2). Because ProtGPT2 is originally a natural language processing (NLP) transformer [6] model, it has been trained using 50 million non-annotated sequences spanning the entire protein space resulting into a "*learned*" protein language. Concomitantly, it can generate sequences that are distantly related to naturally occurring proteins and whose structures resemble the known structural space and can also sample any region, such as all-β structures and membrane proteins.

### *2.3 Perplexity*

Perplexity is a measurement of exponentiated average negative log-likelihood of a given input sequence. In natural language processing, tokenization is used to split texts into smaller units or "*tokens*" as assigned inputs into a language model. By definition from the Hugging Face Artificial Intelligence Group, for a tokenized sequence $X = (x_0, x_1, \dots, x_t)$ , the perplexity of $X$ is:

$$PPL(X) = exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

where $\log p_\theta(x_i|x_{<i})$ is the log-likelihood of the $i^{th}$ token conditioned on the preceding tokens $x_{<i}$ [5]. This measurement is used as an evaluation of the model's ability to predict uniformly among the assigned units from tokenization. In relation to ProtGPT2 model, PPL values of a generated protein can vary per sequence, and therefore sorted from the lowest to highest value as a criteria – lower values are better as the sequences resemble the nearest variant of the input protein dataset.

### *2.4 Antibody Numbering Annotation*

The commonly occurring main-chain conformations of the hypervariable regions were found to occur at sites within the hypervariable regions and in the conserved beta-sheet framework through the examination of the sequences of immunoglobulins which resulted into the Chothia Antibody Numbering annotation scheme [7]. This annotation distinguishes the sdAb's complementarity-determining regions (CDRs) and framework regions (FRs). Essentially, Chothia is a structure-based scheme, created by aligning the variable region crystal structures forming CDRs instead of a sequence-based alignment. Differences between Kabat and Chothia can be found in amino acid insertion points, for example for CDR-L1 and CDR-H1, as well as the loop lengths in CDRs [8]. Overall, Chothia numbering corresponds with the three-

dimensional structures of hypervariable regions of typical length antibodies and therefore used in this study as an additional validation tool for generated sdAb sequences.

### *2.5 Shiny by R Package*

Shiny is an R Software package that is designed to build interactive applications which executes an R code as its backend (canonical form: https://CRAN.R-project.org/package=shiny). This open-source R package provides a framework for a graphical user interface (GUI) assembly which also allows the developer to host a standalone application that consists of reactive inputs and outputs. An input can also be another third-party object-oriented programming script such as C++ or Python wherein the *Shiny* app can execute provided that the compilers for third-party languages are installed in the system. The GUI described here is built on *Shiny* application executable through the R software module.

### *2.6 System Requirements*

The GUI application requires installation of Python (version 3.7 or newer; https://www.python.org/), R Software (https://www.r-project.org/), and R-Studio Integrated Development Environment (https://posit.co/products/open-source/rstudio/). Other dependencies and module versions necessary for seamless operation are discussed in subsection 3.6 of the Results section.

## 3. Results

The GUI includes cascaded steps in which the user can input, explore, and generate novel protein sequences based on the input dataset. GUI menu (blue tab panels of **Figure 1**) includes the set of steps for protein generation:

### *3.1 Step 1: Input Data*

User can upload a CSV file containing protein sequence dataset.

- **Upload CSV File** input: CSV file requirements: the headers for the protein identifier and protein sequence shall be "*name*" and "*seq*", respectively (**Figure 1**).
- **Summary** tab: protein characteristics included in the summary statistics upon dataset upload (**Figure 2**).
- **Observations** tab: a table output showing the contents of the csv file. The default number of observations to view is three (3). This can be adjusted through the numeric entry labeled as "*Number of observations to view:*" above (**Figure 3**).
- **Characteristics** tab: multiple histograms showing the distribution of the protein sequences' physicochemical characteristics (**Figure 4**).

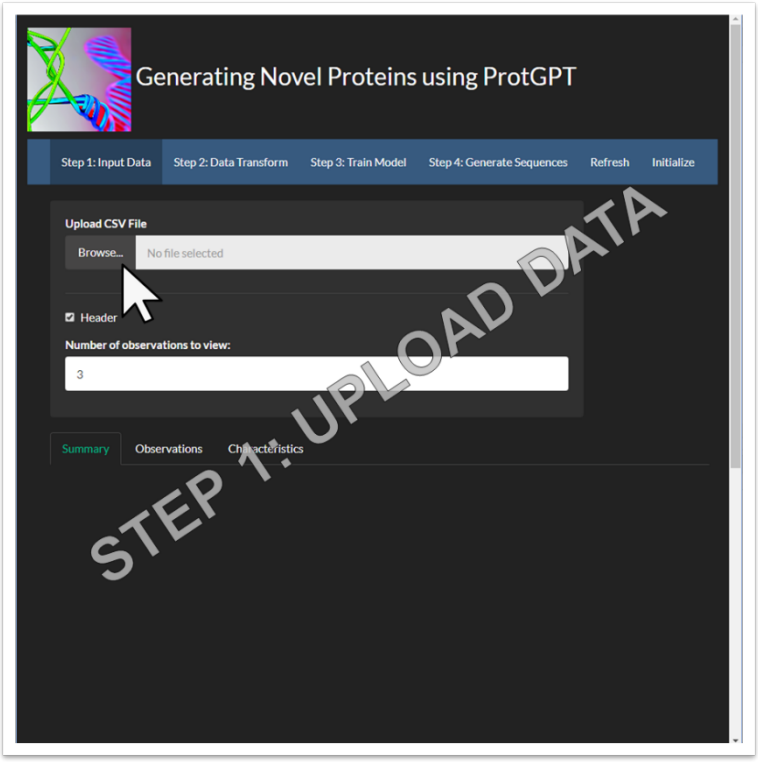


**Figure 1. GUI Start-Up Screen and Step 1: Upload Data.** A user can upload an existing comma separated values (CSV) file dataset consisting of protein sequences for model training. This step operates on the assumption that the user has installed the appropriate system requirements under section 3.6 and running the application on a continued regular use.

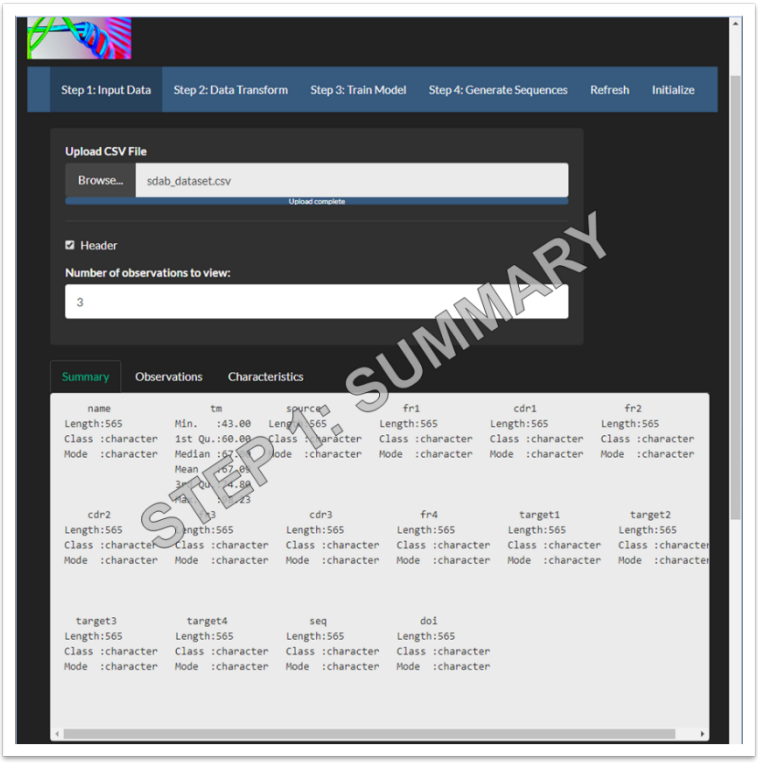


**Figure 2. Step 1: Summary Statistics.** Upon uploading a dataset, summary of descriptive statistics are displayed.

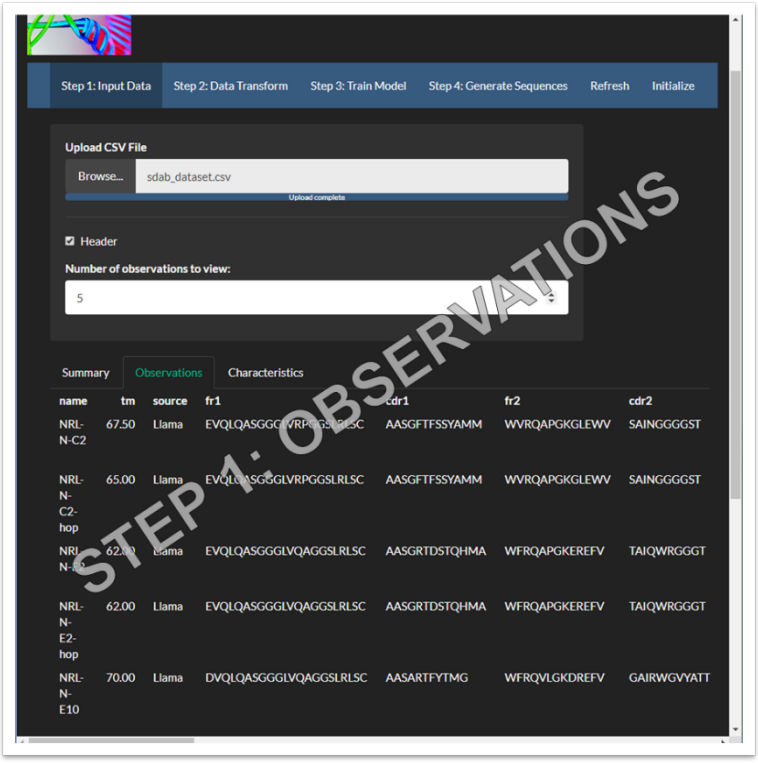


**Figure 3. Step 1: Observations.** Upon uploading a dataset, the contents of the file can be explored by indicating the number of observations to view by the provided numerical option.


**Figure 4. Step 1: Physicochemical Characteristics.** Upon uploading a dataset, several physicochemical characteristics are displayed including peptide length, hydrophobicity distribution of all the sequences, cysteine residue count, and instability index.

### *3.2 Step 2: Transform Data*

Transformation of data into training and validation test datasets for model training.

- User shall perform this step for model training in Step 3 (**Figure 5**).
- GUI will output a warning message if no dataset was uploaded in Step 1 (**Figure 6**).
- Upon transforming the dataset, GUI will output the number of sequences allocated for training and validation datasets. This data split is conducted randomly through a 90:10 ratio – 90% of sequences is allocated in training set while 10% in validation (**Figure 7**).
- User can explore the sequences in home directory as "*training.txt*" and "*test.txt*" files.

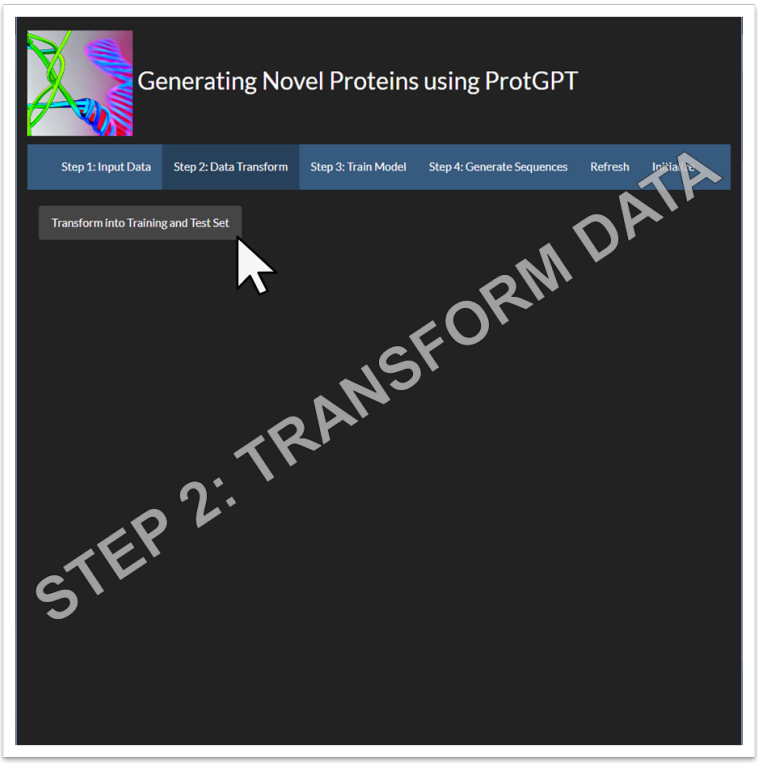


**Figure 5. Step 2: Transform Data.** Using the existing uploaded dataset, data transformation is required before model training.

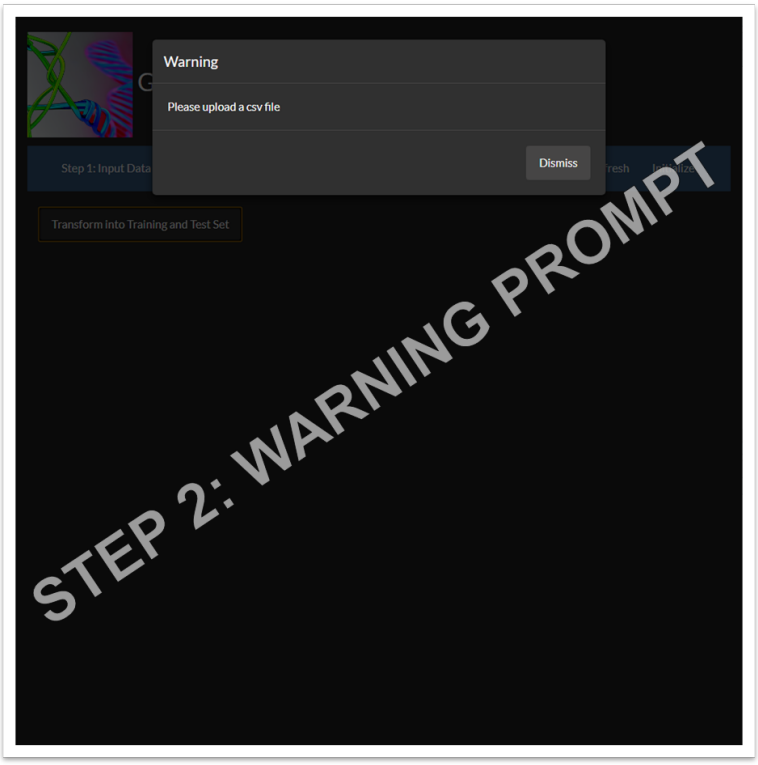


**Figure 6. Step 2: Warning Prompt.** If there is no uploaded dataset, data transformation is not possible. This step resets to Step 1.


**Figure 7. Step 2: Train-Test Splits.** Data transformation is currently set for 90:10 split ratio for model training. Once the transformation is complete, the GUI gives the number of sequences contained in both test and training sets.

### *3.3 Step 3: Train Model*

GPT model will be used for model training.

- User shall input a folder name to save the model (**Figure 8**).
- GUI will output a warning message if no folder name is present (**Figure 9**).
- Model training can take some time to complete depending on the size of the dataset. This process can be skipped if model training has already been completed and saved in the folder name specified by the user (**Figure 10**).
- **Note:** for the current version, only GPT model can be trained. The incorporation of other models (i.e., RNN and Seq2Seq-LSTM) are still in progress.

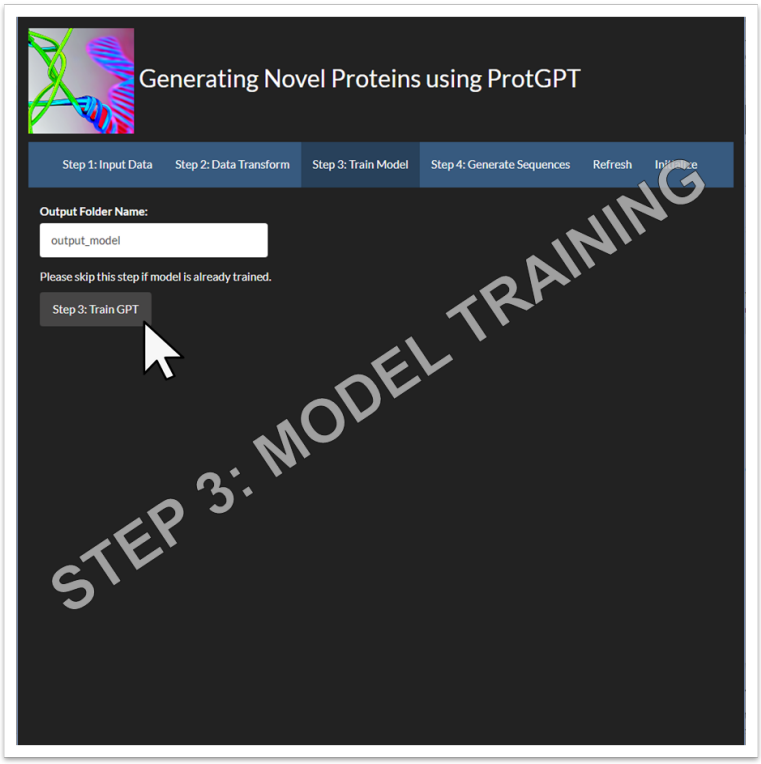


**Figure 8. Step 3: Model Training.** Using the existing uploaded dataset and after data transformation, GPT model can be trained.


**Figure 9. Step 3: Warning Prompt Training.** GPT model to be trained requires a folder name to save all information.


**Figure 10. Step 3: GPT Model Training.** A pop-up window will display the model training progress. The trained GPT model is found in the user directory with the user-specified folder name as indicated.

### *3.4 Step 4: Generate Sequences*

User can generate a desired number of proteins based on inherent characteristics of the input dataset.

- The default number is 20 proteins. An output file name can be specified by the user (**Figure 11**).
- Due to the hardware-dependent runtimes, a loading screen is provided to see progress (**Figure 12**).
- A prompt will display a message upon protein generation completion (**Figure 13**).
- Because a model can generate invalid sequences (i.e., proteins which have invalid sequence lengths, containing non-conventional amino acids, etc.), the generated proteins can be less than or equal to the desired number of new proteins specified by the user.
- **Note:** for the current version, only sdAbs can be generated. Thus, a Chothia antibody annotation scheme through Abnum (http://www.bioinf.org.uk/abs/abnum/) [2] is incorporated in the sequence generation.
- The generated sequences is saved in user directory under the folder name "*generated_sequences*" and can be accessed through a drop-down menu (**Figure 14**).
- The user can further explore characteristics of the generated proteins (**Figure 15**).

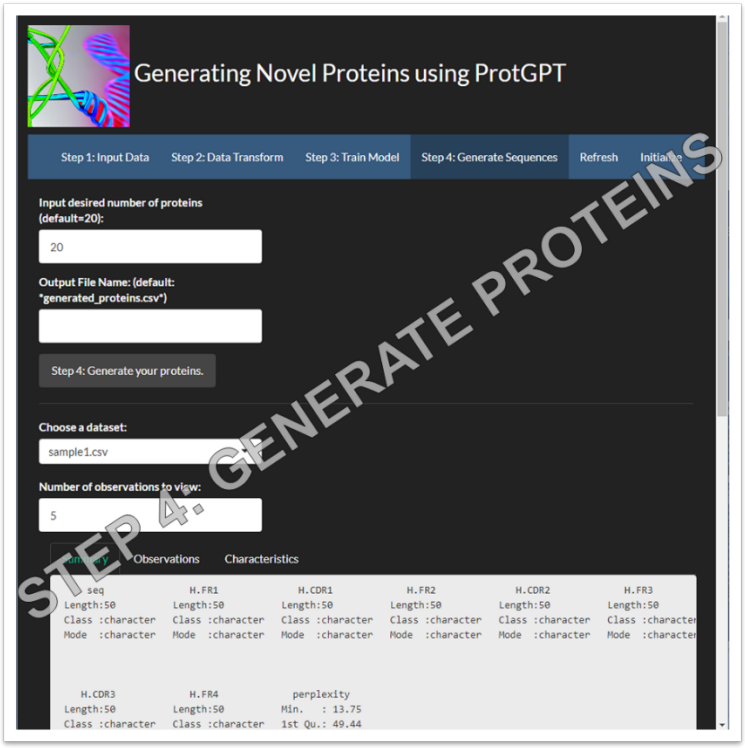


**Figure 11. Step 4: Generate Proteins using the trained GPT Model.** A user can specify the desired number of proteins to be generated. The default number of proteins to be generated is 20. Output file name can also be specified as selected.

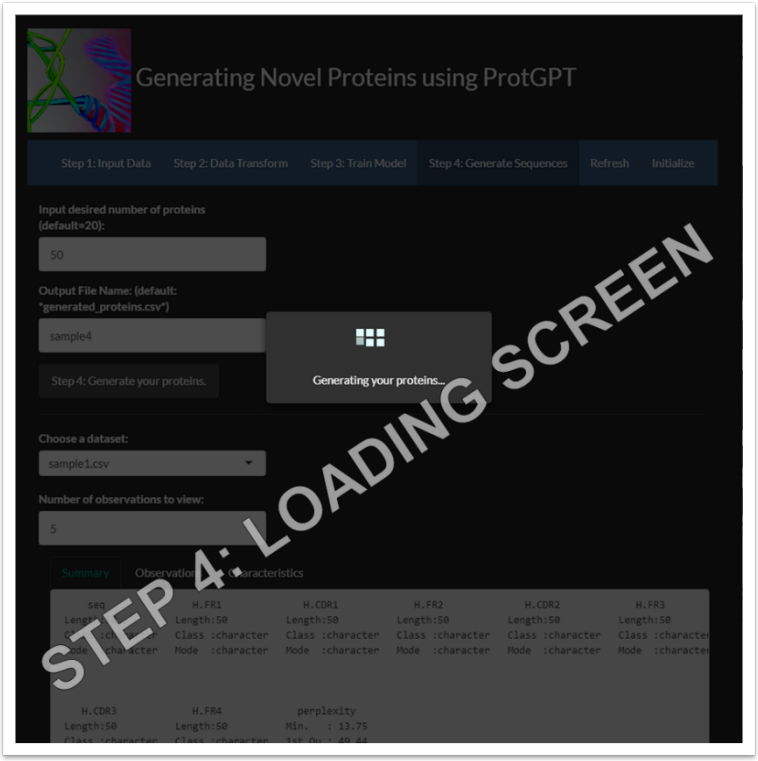


**Figure 12. Step 4: Loading Screen.** A loading screen is provided for the user to monitor protein generation progress.


**Figure 13. Step 4: Generation Complete.** A screen prompt is provided upon completion of protein generation.

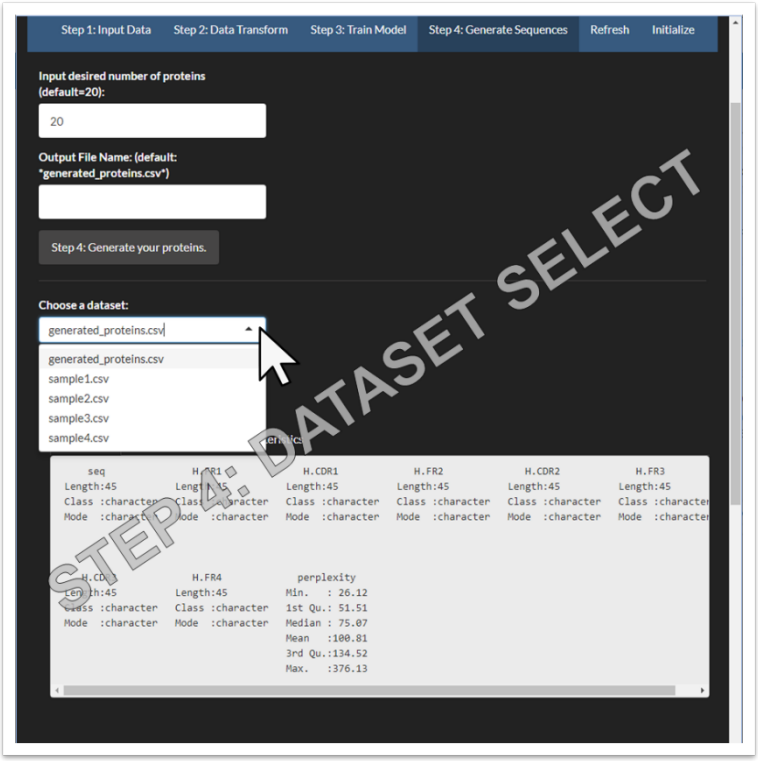


**Figure 14. Step 4: Generated Dataset Exploration.** A drop-down menu is provided to explore generated protein sequences saved in the user directory.


**Figure 15. Step 4: Generated Dataset Characteristics.** Similar to Step 1, the user can investigate each generated dataset file.

### *3.5 Refresh Page*


**Figure 16. Refresh Page.** The button provided refreshes the GUI application and resets to Step 1.

### *3.6 Initialize*

Installs GUI system requirements (**Figure 17**). The list of Python modules are summarized in Table 1.

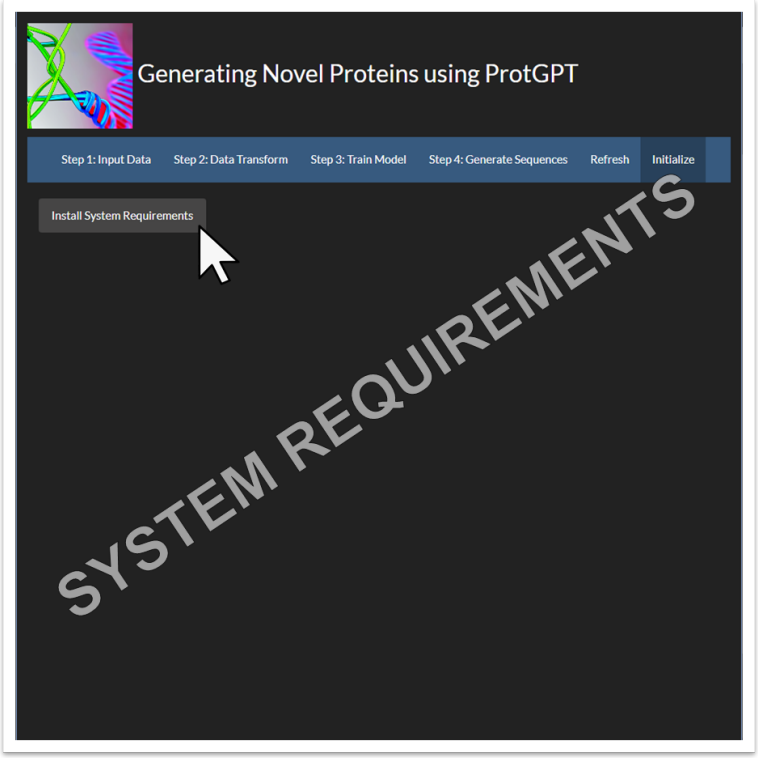


**Figure 17. Install System Requirements.** Install GUI system requirements (recommended for first time users).

**Table 1.** Python modules and versions to be installed upon installing system requirements (Figure 17).

| **Module** | **Reference** |
|---|---|
| accelerate >= 0.12.0 | https://pypi.org/project/accelerate/ |
| torch >= 1.3 | https://pypi.org/project/torch/ |
| datasets >= 1.8.0 | https://pypi.org/project/datasets/ |
| sentencepiece != 0.1.92 | https://pypi.org/project/sentencepiece/ |
| protobuf | https://pypi.org/project/protobuf/ |
| evaluate | https://pypi.org/project/evaluate/ |
| scikit-learn | https://pypi.org/project/scikit-learn/ |
| transformers | https://pypi.org/project/transformers/ |

### *3.7 Sample Generated Data*

The generated proteins, in this case sdAbs, are saved in the user directory under the "*generated_sequences*" folder. These sdAbs are further annotated indicating each sequence's framework and complementarity-determining regions sorted by its perplexity in ascending order. A sample of 5 sdAbs are shown in Table 2.

**Table 2.** Generated sdAb samples (5 annotated sequences sorted by perplexity).

| seq | H-FR1 | H-CDR1 | H-FR2 | H-CDR2 | H-FR3 | H-CDR3 | H-FR4 | perplexity |
|---|---|---|---|---|---|---|---|---|
| EVQLVESGGGLVQAGDSL | | | | | | | | |
| RLSCSASGFTLSDYGVGW | | | | | TYYADS | | | |
| FRQAPGKGLEWVSVIQSP | | | | | VKGRFT | | | |
| GGRTYYADSVKGRFTISR | | | | | ISRDNA | | | |
| DNAKNTVSLLMNSLKPED | EVQLVES | | | | KNTVSL | | WGQ | |
| TAVYFCAAVERCTTSLAN | GGGLVQ | | GVGWFR | | LMNSLK | VERCT | GTA | |
| YREWGQGTAVTLNCTSPL | AGDSLRL | | QAPGKG | | PEDTAV | TSLAN | VTL | |
| PTPYHYMGPQEVSS | SCSAS | GFTLSDY | LEWVSVI | QSPGGR | YFCAA | YRE | NC | **26.12446** |
| QVQLVESGGGLVQPGGSL | | | | | DTYADS | | | |
| RLTCAASGLIFGSYAMGW | | | | | VKGRFT | | | |
| FRQAPGKGLEFVAAISWS | | | | | ISRDFAT | | | |
| GGDTYADSVKGRFTISRD | QVQLVES | | | | NTAYLQ | | WGQ | |
| FATNTAYLQMNYLRQQD | GGGLVQP | | AMGWFR | | MNYLR | | GTQ | |
| TAVYYCKGAGRAYWGQ | GGSLRLT | | QAPGKG | | QQDTAV | | VTV | |
| GTQVTVSSA | CAAS | GLIFGSY | LEFVAAI | SWSGG | YYCKG | AGRAY | SS | **28.19061** |
| QVQLVESGGGLVQPGGSL | | | | | | | | |
| RLSCAASGGALSDYNMG | | | | | THIADS | | | |
| WFRQAPGKQRELVASITG | | | | | VKGRFT | | | |
| GGNTHIADSVKGRFTISQD | | | | | ISQDNA | DLKMQ | | |
| NAKNTVTLQMNNLTPED | QVQLVES | | | | KNTVTL | VAAY | WGQ | |
| TALYYCAADLKMQVAAY | GGGLVQP | | NMGWFR | | QMNNL | MNQRS | GTQ | |
| MNQRSVDYWEHWGQGT | GGSLRLS | | QAPGKQ | | TPEDTA | VDYW | VTV | |
| QVTVSA | CAAS | GGALSDY | RELVASI | TGGGN | LYYCAA | EH | SA | **29.94931** |
| QVKLEESGGGLVQPGGSL | | | | | TYYADS | | | |
| RLSCAASGYTYSSYSIGW | | | | | VKGRFT | | | |
| VRQAPGKEREGVAVIRSS | | | | | ISRDNS | | | |
| DGTTYYADSVKGRFTISR | QVKLEES | | | | KNTLYL | | WGQ | |
| DNSKNTLYLQMNSLEPED | GGGLVQP | | SIGWVRQ | | QMNSLE | NSQRR | GTQ | |
| AAMYFCAVNSQRRRLPSQ | GGSLRLS | | APGKERE | | PEDAAM | RLPSQ | VTV | |
| EYTYWGQGTQVTVSS | CAAS | GYTYSSY | GVAVI | RSSDGT | YFCAV | EYTY | SS | **31.38236** |
| DVQLQESGGGSVQAGGSL | | | | | TKYADS | | | |
| RLSCEASGYTYSSYSIGW | | | | | VKGRFT | | | |
| VRQAPGKGLEWVSAINSG | | | | | ISRDNS | | | |
| GLTKYADSVKGRFTISRD | DVQLQES | | | | KRTAYL | | WGQ | |
| NSKRTAYLQMNNLKPED | GGGSVQ | | SIGWVRQ | | QMNNL | | GTQ | |
| TAIYYCRAFGPADYWGQ | AGGSLRL | | APGKGLE | | KPEDTA | FGPAD | VTV | |
| GTQVTVSS | SCEAS | GYTYSSY | WVSAI | NSGGL | IYYCRA | Y | SS | **32.12616** |

## 4. Conclusions

Year 4 extended statement of work focused on building a fast and efficient system for creating novel sdAbs which is facilitated by the development of a graphical user interface. Although conventional poly- and monoclonal antibodies are indispensable reagents in basic research, diagnostics, and therapeutics, their shortcomings including batch-to-batch variability of the polyclonals, and very high cost and time-consuming production of monoclonals have pushed its direction toward the development of novel and improved sdAb sequences. Upon the initiation of the program, the Team invested on the capabilities of sdAbs due to the Center's key expertise such as: 1) NRL subject matter experts, 2) available in-house datasets including each antibody's biophysical properties, 3) lower costs associated with synthesis and manufacturing, and 4) previously reported sdAb systems that were implemented as indicated by accomplished milestones such as the use of *Recurrent Neural Networks* (RNN) and *Local Interpretable Model-Agnostic Explanations* (LIME) for the analysis and improved thermal stability of generated sdAbs.

### *4.1 Physicochemical/Pharmacokinetic Profiles of Generated Antibodies*

The generated antibodies from the ProtGPT2 model serve as variants from the input sdAb dataset that contains experimentally verified and existing sdAb sequences from several animal sources. Although a single animal source of sequences such as camelids can be trained to provide a constraint and focused sequence profile, the output sdAbs should resemble similar or better physicochemical/pharmacokinetic properties and target antigens from their parent sequences. The basic biophysical properties previously shown on Figures 4 and 15 display the amino acid lengths, hydrophobicity indices, cysteine counts, and instability indices of the parent and generated sequences, respectively. These properties are specifically chosen as preliminary inspection tools for the validity of ProtGPT2's produced sdAbs.

As an example, cysteine residues serve essential roles in protein structure and function by conferring stability through disulfide bond formation which maintains proper maturation and localization through protein-protein intermolecular interactions [9]. Accompanied by the Chothia annotation scheme applied on the generated sdAbs, the cysteines are critically important in the determination of FRs and CDRs. Moreover, their highly reactive thiol side chains form intermolecular disulfide bonds and also provides an appealing route for conjugating toxic agents to antibodies. On a specific perspective, antibody-drug conjugates (ADCs) are an important mechanism for the rapid inactivation and extensive removal of drugs and poisons from the body which emphasizes its major significance in clinical toxicology [10]. ADCs may improve the therapeutic index of cytotoxic drugs by using the interaction of the antibody with tumor-specific antigens and the four inter-chain disulfides forming cysteine residues are the main targets for conjugation due to high solvent accessibility. Additionally, high stability of sdAbs indicated by the hydrophobicity and instability indices are a well-sought biophysical property which renders their potential development as a therapeutic. Parallel comparisons between the input sdAb and ProtGPT2-generated sdAb are critical in the viability of synthetic constructs.

### *4.2 Future Directions*

Future versions of the prototype presented here will include options for training our developed and modified recurrent neural network (RNN), variational auto-encoder (VAE), and long short-term memory-sequence to sequence (LSTM-Seq2Seq) models. Multiple validation measurements and additional biophysical properties through *in-silico* methods will also be incorporated.

### *4.3 Authors' Notes*

Applicable GUI codes, including R scripts, input dataset, antibody annotations, and ProtGPT2 python scripts are available upon request.

## Acknowledgements

We acknowledge funding support through base funds of the Naval Research Laboratory (WU# 1V33) and funds from the Defense Threat Reduction Agency (HDTRA1343577).

## References


1. Ventola, C.L., *The antibiotic resistance crisis: part 1: causes and threats.* P t, 2015. **40**(4): p. 277-83.
2. Gould, I.M. and A.M. Bal, *New antibiotic agents in the pipeline and how they can help overcome microbial resistance.* Virulence, 2013. **4**(2): p. 185-191.
3. Stokes, J.M., et al., *A deep learning approach to antibiotic discovery.* Cell, 2020. **180**(4): p. 688-702. e13.
4. Dean, S.N., et al., *PepVAE: Variational Autoencoder Framework for Antimicrobial Peptide Generation and Activity Prediction.* Front Microbiol, 2021. **12**: p. 725727.
5. Ferruz, N., S. Schmidt, and B. Höcker, *ProtGPT2 is a deep unsupervised language model for protein design.* Nature Communications, 2022. **13**(1): p. 4348.
6. Vaswani, A., et al. *Attention is All you Need*. in *NIPS*. 2017.
7. Chothia, C. and A.M. Lesk, *Canonical structures for the hypervariable regions of immunoglobulins.* J Mol Biol, 1987. **196**(4): p. 901-17.
8. Abhinandan, K.R. and A.C.R. Martin, *Analysis and improvements to Kabat and structurally correct numbering of antibody variable domains.* Molecular Immunology, 2008. **45**(14): p. 3832-3839.
9. Meitzler, J.L., et al., *Conserved cysteine residues provide a protein-protein interaction surface in dual oxidase (DUOX) proteins.* J Biol Chem, 2013. **288**(10): p. 7147-57.
10. PRESCOTT, L.F., *Drug conjugation in clinical toxicology.* Biochemical Society Transactions, 1984. **12**(1): p. 96-99.