VCU College of Engineering

CS 25-343 Interactive Online Interface for Visualization and Analysis of Molecular Data

Project Proposal

Prepared for

Lukasz Kurgan

VCU College of Engineering

By

Benjamin Baldwin

Megan Lee

Mahmuda Sarker

James West

Under the supervision of

Ajay Arya, Sushmita Basu Das, Lukasz Kurgan

10/14/2024

Executive Summary

The project aims to redesign the graphical interface for DescribePROT, a tool that provides amino acid-level descriptors of protein structures and functions. As the volume of protein sequence data continues to expand rapidly, researchers require fast, reliable, and visually appealing methods for analyzing this information. Currently, DescribePROT's graphical output for protein-level characterization falls short in performance, often exceeding acceptable generation times, and the interface lacks the modern aesthetics and interactive elements expected by users.

This project will address these shortcomings by reducing the interface generation time to less than one second and overhauling the design to improve both visual appeal and interactivity. The redesign will maintain current functionality while enhancing reliability and user experience. New features, such as improved zoom and panning capabilities, will allow for more detailed, clear, and efficient data analysis. The interface will seamlessly integrate with DescribePROT’s existing database and infrastructure, ensuring compatibility and ease of use.

Key deliverables include the development of fully optimized code, comprehensive documentation, a user manual, and deployment instructions, along with an analysis of the best system environments for optimal performance. The project will adhere to relevant software standards, including ISO 9241-110 for user interaction, WCAG 2.1 for accessibility, and PEP 8 for Python best practices.

Supported by Dr. Kurgan and sponsored by the VCU School of Engineering, this project will be carried out in two phases, with significant milestones set for both the Fall 2024 and Spring 2025 semesters. By the end of the project, a fully functional and enhanced interface will be delivered, providing a more efficient and user-friendly tool for the DescribePROT team and its users.

Table of Contents

Section A. Problem Statement

With protein sequence data expanding at an unprecedented rate, researchers are increasingly pressed to find effective ways to analyze and characterize this information at various levels of complexity. As of the 2020_04 release of UniProt, a comprehensive resource of protein sequence and functional information, there were over 189 million protein-coding regions [1]. Information about these proteins could be found in many different sources but they all notably lacked large amounts of Amino Acid level annotations.

DescribePROT (Database of structure and function residue-based predictions of PROTeins), launched in 2020, emerged as a resource providing these much needed AA-level descriptors of protein structures and functions. As of their second paper, published in 2023, the database currently covers 19 structural/functional descriptors for proteins in 273 reference proteomes generated by 11 accurate and complementary predictive tools [2].

Despite its comprehensive nature, the current graphical output of DescribePROT presents significant challenges. The existing graphical interface is slow, often exceeding user expectations regarding generation times. Furthermore, the visual appeal and interactive elements of the interface require improvement to align with modern usability and aesthetic standards.

This project aims to address these issues by redesigning the graphical output for protein-level characterization within DescribePROT. The primary goal is to reduce graphical interface generation time to under one second. In addition to improving speed, the project will focus on overhauling the aesthetics of the interface, making the graphical representations clearer and more visually appealing, with enhanced interactive elements for a better user experience. Ensuring reliability is also a key objective, as the system must function without any loss of functionality. The redesigned interface must seamlessly integrate with DescribePROT's existing database to maintain continuity and ensure smooth operation for users.

While DescribePROT has its own unique set of functions, other protein databases offer graphical outputs that can serve as valuable sources of inspiration for this project. For example, the Protein Data Bank (PDB) features a modern layout, illustrated in figure 1, that effectively aligns with its sequence data, providing a clean and intuitive interface. Similarly, MobiDB, illustrated in figure 2, offers graphical outputs that, while not identical in function, share comparable visual and structural elements that could inform our redesign efforts.

This project is supported by Dr. Kurgan and sponsored by the VCU School of Engineering, with the DescribePROT team and their users as our stakeholders. Upon conclusion of this project in the Spring of 2025, our team will have delivered a graphical presentation that is faster than its current implementation. This new interface will also be given updated visuals to be more closely aligned with other similar databases. All of this will be done with a focus on reliability to ensure there is no loss of functionality in the transition.

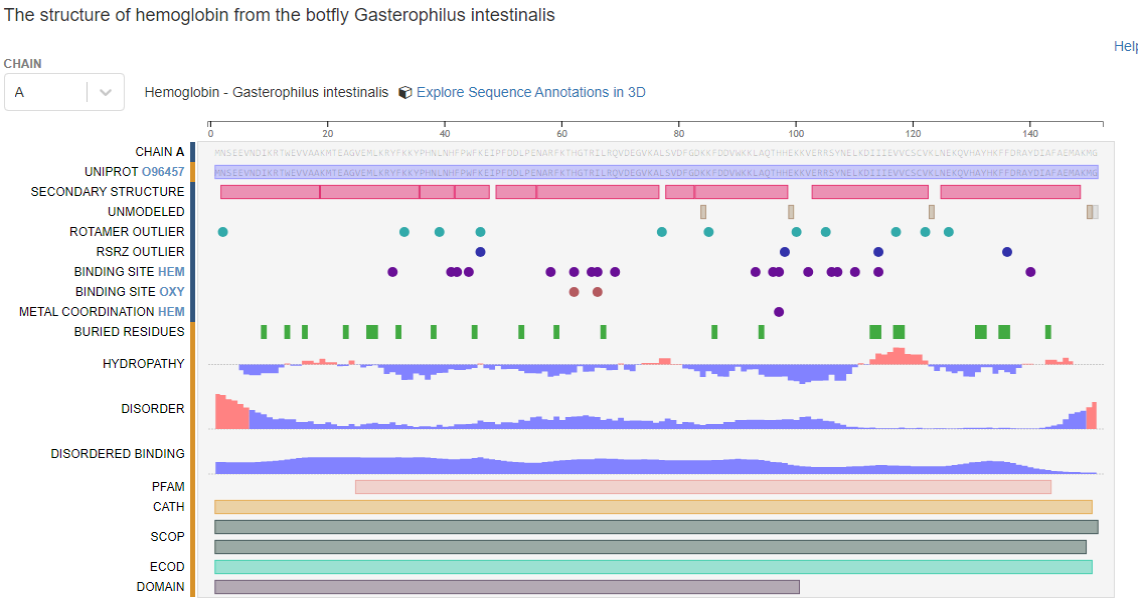

Figure 1. TheProtein Data Bank (PDB) graphical interface [3].

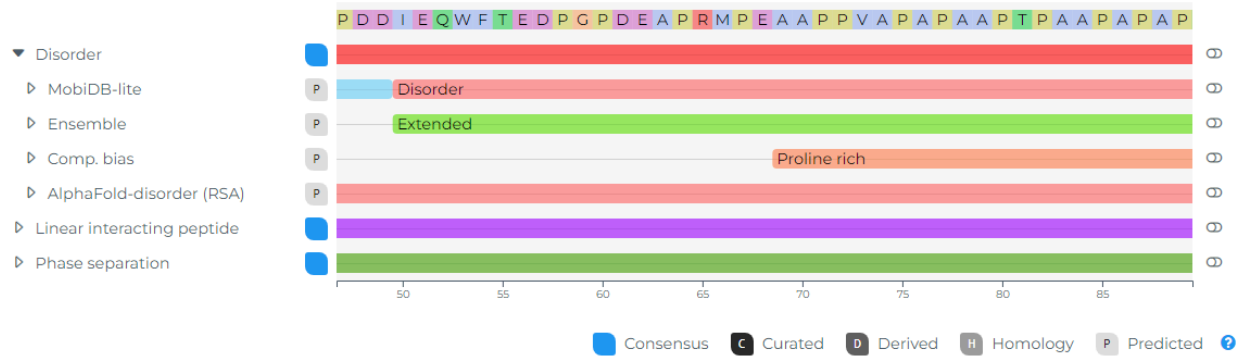

Figure 2. The MobiDB graphical interface [4].

Section B. Engineering Design Requirements

B.1 Project Goals (i.e. Client Needs)

Our overall goals for this project are on enhancing the Protein Level Characterization graphical interface to better serve the user's needs. Our aim is not only to streamline the generation time of the graphical interface but also to elevate the overall aesthetic appeal of the visualization functions. The specific goals are listed here:

- To study and identify potential technology alternatives for the creation of the protein graphical interface
- To decrease the generation time of the graphical interface
- To improve the appearance of the visualization functions
- To maintain the current level of reliability

B.2 Design Objectives

- The new design will be faster, have different graphics, and have slightly different functionality.
- The main objective is to reduce the loading time when a protein's data is searched in the database and its information is displayed.
- Another objective is to improve the zoom feature by adding a wider range of annotation markers that appear based on the zoom level to increase readability of the data at different levels.
- Another objective for the zoom feature is to improve the panning and how the zoom is activated.
- The goal of the project is to improve the current functionality of the displayed data while still retaining the same structure and layout.

B.3 Design Specifications and Constraints

- The design implementation has to be compatible with current infrastructure with some flexibility on libraries and language used.
- The new design must also work within the same larger framework.
- The new design must also be compatible across browsers.
- The design must preserve the current features and only improve them.

B.4 Codes and Standards

This project, "Interactive Online Interface for Visualization and Analysis of Molecular Data," will adhere to the following relevant standards to ensure usability, performance, and technical consistency across various components of the system.

1. Software and Interface

ISO 9241-110 (Ergonomics of Human-System Interaction): This standard ensures that the interface design is user-friendly, intuitive, and supports clear navigation. By applying these guidelines, we aim to provide an effective and enjoyable user experience for researchers, focusing on easy-to-understand annotations and smooth interaction with molecular data.

WCAG 2.1 (Web Content Accessibility Guidelines): In order to make the interface accessible to a wide range of users, this standard will be followed. This includes ensuring clear labeling, avoiding reliance on color-coding alone, and incorporating features that support all users in engaging with the platform effectively.

Python Best Practices (PEP 8): Since Python will be the primary development language, following PEP 8 ensures code readability, maintainability, and efficiency. This approach helps avoid runtime delays and improves the overall performance of the interface.

2. Data Handling and Visualization

MySQL and PHP Best Practices: The project will use MySQL for database management and PHP for server-side scripting. Best practices for MySQL will ensure efficient database design, secure querying, and optimized data retrieval. PHP will follow standards for secure and efficient server-side processing.

Database Management System (DBMS) Standards: We will adhere to DBMS principles for the organization, retrieval, and management of data, ensuring consistency and performance when handling molecular datasets.

CSV Format (RFC 4180): Users will have the option to download data in CSV format. This standard ensures the exported data is structured, easy to share, and widely accepted for data exchange.

MIAME (Minimum Information About a Microarray Experiment): These guidelines will be followed for the consistent and structured representation of molecular data, improving data clarity and usability in research.

3. Graphics and Interaction

SVG 2 (Scalable Vector Graphics): To maintain the clarity and sharpness of visualized molecular data at all zoom levels, we will use SVG standards. This ensures that visualizations remain readable and precise, regardless of how the user manipulates the view.

4. Compatibility

ISO/IEC 25010 (Systems and Software Quality Requirements and Evaluation): This standard will ensure that the system is tested for reliability and performance on various operating systems. Special attention will be given to compatibility with widely-used platforms, including Linux distributions such as Ubuntu, to ensure flexibility across different research environments.

5. Data Integrity and Error Handling

JSON Format (RFC 8259): We will use JSON as a standard format for data exchange in the interface. This ensures consistency in data transmission and reduces the likelihood of errors when handling molecular data or interaction logs.

Python Error Handling Best Practices: To ensure robust performance and prevent system crashes, the project will follow Python's best practices for error handling, using try-except blocks to catch and manage potential errors gracefully. This will help maintain system stability during both normal operation and unexpected user interactions.

Section C. Scope of Work

The project scope defines the boundaries of the project encompassing the key objectives, timeline, milestones and deliverables. It clearly defines the responsibility of the team and the process by which the proposed work will be verified and approved. A clear scope helps to facilitate understanding of the project, reduce ambiguities and risk, and manage expectations. In addition to stating the responsibilities of the team, it should also explicitly state those tasks which fall *outside* of the team's responsibilities. *Explicit bounds* on the project timeline, available funds, and promised deliverables should be clearly stated. These boundaries help to avoid *scope creep*, or changes to the scope of the project without any control. This section also defines the project approach, the development methodology used in developing the solution, such as waterfall or agile (shall be chosen in concert with the faculty advisor and/or project sponsor). Good communication with the project sponsor and faculty advisor is the most effective way to stay within scope and make sure all objectives and deliverables are met on time and on budget.

C.1 Deliverables

The project, "Interactive Online Interface for Visualization and Analysis of Molecular Data," will result in the following key deliverables, ensuring the project meets both functional requirements and high standards of usability and performance:

Key Deliverables

Working Code: Fully functional and optimized code for the interface, including MySQL database queries, PHP scripts, and all front-end components. The code will be tested to meet performance and usability standards.

Deployment Instructions: Clear, concise, and easy-to-follow deployment instructions that include server setup, MySQL database installation, and PHP configuration. These will ensure the system can be reliably deployed in various environments.

Documentation

Code Documentation: Detailed documentation of the codebase, following industry standards (e.g., Python PEP 8) to ensure clarity and maintainability.

User Manual: A guide designed for end users, detailing how to use the interface, visualize data, and export data via the CSV download feature.

Installation Guide: Comprehensive installation instructions for system setup on different platforms, with a focus on compatibility with Linux systems (e.g., Ubuntu).

Study on Optimal Environment

This analysis and report define the best software platform (libraries) for developing the new visualization using current hardware and OS.

Final Report and Presentation

A complete report summarizing the project development process, system design, and results, alongside a polished presentation highlighting key project achievements, to be delivered at the Capstone EXPO and other academic presentations.

Presentation Slides

Professional slide decks for all project presentations, including those for interim reviews and the final Capstone EXPO presentation.

C.2 Milestones

This section outlines the key milestones for the project, divided into the 2024 Fall semester and 2025 Spring semester. These milestones will guide the team through the development process, ensuring timely completion of the project deliverables.

Fall 2024 Milestones

1. **Understanding the current solution**

   Timeline: Early September 2024 - Mid October 2024

   Description:

   This milestone focuses on thoroughly understanding the current solution in use. This involves analyzing the existing system, identifying its key functionalities, and recognizing any limitations or areas for improvement.

   Deliverable:

   This project proposal report will be the first deliverable. The proposal will include sections up to C.3. Resources, with additional sections to be added in a forthcoming report.

2. **Evaluate Alternatives for the Design**

   Timeline: Mid October 2024 - Mid-November 2024

   Description:

   After understanding the current solution, the team will evaluate various alternatives that could be used to solve the problem that the current solution is facing. This phase involves researching potential approaches to enhance or replace the current solution. The alternatives will be assessed based on alignment with project goals.

   Deliverable:

   A list of evaluated alternatives with a brief description of each, along with the pros and cons for each design.

3. **Compare the Alternatives**

   Timeline: Mid November 2024 - Late November 2024

   Description:

   The identified alternatives will be compared using a structured evaluation framework in this phase. Factors such as cost, performance, complexity, and implementation risks will be considered to determine the most suitable design option.

   Deliverable:

   A comparative analysis document or presentation outlining the evaluation criteria and ranking the alternatives, leading to a recommendation for the best option.

Spring 2025 Milestones

1. **Select the Best Option and Develop a Prototype**

   Timeline: Mid January 2025 - Early February 2025

   Description:

   Based on the comparative analysis, the best graphing tool alternative will be selected. A prototype of the chosen design will be developed to validate the concept and assess its viability in practice.

   Deliverable:

A working prototype, along with a design and development report documenting the design process and rationale.

2. **Re-evaluate the Selected Choice**

   Timeline: Early February 2025 - Late February 2025

   Description:

   After the prototype is completed, the team will re-evaluate the code. This will involve testing the prototype against the project's performance requirements and making necessary refinements to ensure it meets the desired objectives.

   Deliverable:

   An evaluation report summarizing the prototype testing results, including any adjustments or improvements made to the design.

3. **Implement Additional Panels**

   Timeline: Late February 2025 - Late March

   Description:

   With the refined code, the team will proceed to implement the remaining components or panels needed to complete the system. This final phase ensures that the solution is fully operational and ready for final testing and presentation.

   Deliverable:

   The final implementation of the system, along with a final report and EXPO presentation.

C.3 Resources

The following section introduces the required resources and software libraries.

1. **Input Data**

   Access to accurate and relevant input data is crucial for visualizing molecular data through the graph. The project relies on this data to visualize the graph and determine whether the result is suitable for our project objectives.

2. **Access to the Database**

   Continuous access to the project's operational database is necessary for retrieving input data. At the end of the project, the database will be used in real-time.

3. **Current Code Base**

   The existing code base will serve as the foundation for further development. The code utilizes the following key libraries:

- **Operating System Interface:** For interacting with the file system and managing directories.
- **NumPy:** For numerical computations and array operations.
- **Pandas:** For data manipulation and analysis.
- **CSV:** For reading from and writing to CSV files.
- **Plotly:** For creating interactive plots and visualizations.
- **MinMaxScaler (from sklearn.preprocessing):** For data normalization and scaling. These libraries are essential for various computational tasks such as data preprocessing, visualization, and analysis.

4. **Live Database**

   Access to the live database is required for handling real-time data and performing operations in a production-like environment. This will enable accurate testing and performance evaluations in real-world conditions.

5. **Software and Platforms**

   The project may also require additional software resources, including:

   - Integrated Development Environments (IDEs) for coding and debugging.
   - Version control systems (e.g., Git) for managing and tracking code changes across the team.

References

[1]Uniprot: A worldwide hub of Protein knowledge. (2018). Nucleic Acids Research, 47(D1). https://doi.org/10.1093/nar/gky1049

[2]Basu, S., Zhao, B., Biró, B., Faraggi, E., Gsponer, J., Hu, G., Kloczkowski, A., Malhis, N., Mirdita, M., Söding, J., Steinegger, M., Wang, D., Wang, K., Xu, D., Zhang, J., & Kurgan, L. (2023). DescribePROT in 2023: More, higher-quality and experimental annotations and improved data download options. Nucleic Acids Research, 52(D1). https://doi.org/10.1093/nar/gkad985

[3]Bank, R. P. D. (n.d.). 1D PFV: 2C0K. RCSB PDB. https://www.rcsb.org/sequence/2C0K

[4]P04637 - Cellular tumor antigen p53. MobiDB. (n.d.). https://mobidb.org/P04637