Computational Biology Helpers

Eli B. Author,[1] Ryan Z. Author,[2] Kishitij K. Author,[3] and Connor B. Author[4]

[1]Computer Science, Virginia Tech, 185 Kent St, Blacksburg VA, 24060, United States; e-mail: elibp13@vt.edu

[2]Computer Science Virginia Tech, 185 Kent St, Blacksburg VA, 24060, United States; e-mail: ryanz22@vt.edu

[3]Computer Science Virginia Tech, 185 Kent St, Blacksburg VA, 24060, United States; e-mail: kishitijk@vt.edu

[4]Department of Computer Science, Virginia Tech, 1713 Patrick Henry Dr NW, Blacksburg VA, 24060, United States; e-mail: cbrodish03@vt.edu

ABSTRACT

Many software engineers interested in computational biology struggle with the complexity of algorithms and associated projects. Traditional biology tools you might find with a quick internet search are often outdated, unintuitive, and sport poor performance. The solution to this problem is a highly interactive web application, combining various computational biology features like BLOSUM (Block Substitution Matrices) and sequence alignments in two- and three-dimensional visualizations for both beginners and experts. Users can also edit, save, and share algorithm input and output for convenience. Our development process was inspired by software design principles instilled during the course and guided by usability heuristics to make sure we conformed to typical standards. In addition, we conducted weekly scrum meetings to address potential problems and solutions during development. By making these algorithms accessible and intuitive, we aim to bridge the gap between software engineers and computational biology.

BACKGROUND & KEY WORDS

Computational biology, software engineering, algorithm visualization, bioinformatics, sequence alignment, substitution matrices, scrum meeting

INTRODUCTION

Computational biology plays a critical role in bridging biology and computer science, offering tools and algorithms to analyze biological data at scale. However, many software engineers struggle to dive into this interdisciplinary field due to the complexity of the algorithms and the lack of intuitive tools. As students embarking on a journey into computational biology, we identified a significant gap: existing tools are often difficult to navigate, outdated, and poorly optimized for modern web environments.

Motivated by these challenges, we developed Computational Biology Helpers, an interactive web application designed to simplify computational biology for beginners and experts alike. By combining intuitive design principles with robust algorithm implementations, such as BLOSUM and sequence alignments in both 2D and 3D visualizations, our platform aims to enhance accessibility and usability. Throughout development, we adhered to best practices in software engineering, such as type-safe programming with TypeScript, and leveraged modern frameworks like React and Next.js for performance optimization. With a strong emphasis on usability heuristics and continuous integration, we provide a scalable, efficient, and engaging platform for learning and applying computational biology concepts.

RELATED WORK

In the field of computational biology, there have been numerous attempts to create interactive tools for algorithm visualization and application. For instance, Sinnefa's GitHub repository [1] extends the Needleman-Wunsch algorithm to handle three sequences simultaneously. While innovative, this tool lacks interactivity and usability, especially for non-expert users. Similarly, the BLOSUM matrix computation guide on Wikipedia offers an excellent theoretical resource but does not provide practical implementation support [2].

Another example is an interactive website hosted on GitHub Pages [3], which facilitates global alignment computations. Despite its functionality, users frequently report a cluttered and confusing interface, which detracts from its utility. These tools often overlook fundamental usability principles, such as clarity, consistency, and accessibility.

Our project addresses these shortcomings by combining cutting-edge deployment technologies with intuitive design. For example, we implemented 2D and 3D sequence alignment visualizations that balance simplicity and power, ensuring both beginners and experts can leverage the platform effectively. Additionally, features such as a clean interface, customizable views, and seamless input/output sharing were inspired by established usability heuristics as discussed in class.

HIGH-LEVEL DESIGN

The high level design behind Computation Biology Helpers is a React Next.js Webapp built using Typescript. The use of typescript ensures that all code is type safe and therefore less prone to bugs, a design choice we chose while making the web application. The use of React & Next.js are for optimizations to speed necessary for the complex algorithms that are in Computational Biology. Our app uses libraries such as Three.js to help visualize the algorithms. The use of other libraries represents a design choice to pull from existing code and libraries to assist in the calculation of these algorithms.

Our webapp uses GitHub to ensure consistent code and workflows across the team. This is based on the software engineering idea that the codebase should be maintained on a repository accessible to all developers involved so that we can work in sync.

DEPLOYMENT & MAINTENANCE

The deployment of our project was streamlined using modern DevOps practices to ensure efficiency and scalability. Our application is hosted on Vercel, leveraging its seamless integration with GitHub to enable Continuous Integration and Continuous Deployment (CI/CD). Any updates pushed to the GitHub repository are automatically reflected on the live application within less than 3 minutes, eliminating the need for manual deployment processes. We also configured the system to use a custom domain purchased through Cloudflare, providing users with a professional and reliable platform. To ensure long-term maintenance, the repository is mirrored to a private GitHub repository using Git Workflows, safeguarding our work and enabling secure collaboration. This setup not only enhances the development cycle but also ensures that the platform remains up-to-date and responsive to user needs.

CONCLUSION

The Computational Biology Helpers project showcases the potential of combining computational tools with modern software engineering principles to address a critical gap in the accessibility of computational biology. By leveraging technologies like React, Next.js, and Three.js, and deploying our app using CI/CD workflows, we created a solution that balances power, usability, and scalability. Our platform allows users to visualize and interact with complex algorithms, enhancing both learning and practical application.

Looking forward, this project sets the foundation for future enhancements, including expanding the library of supported algorithms, improving UI/UX elements, and incorporating multithreading to boost computational performance. We hope this initiative serves as a bridge between software engineering and computational biology, inspiring more accessible and innovative tools in the field.

DISCUSSION

The development of *Computational Biology Helpers* presented unique challenges, particularly in implementing 3D visualizations using Three.js, which had not been applied to sequence alignments before. Designing clear and interactive 3D models while maintaining performance required innovative problem-solving and extensive optimization. Despite these advancements, some features, such as hovering over matrix cells in 2D alignments to reveal additional data, are absent and represent areas for improvement.

This project demonstrates the potential of combining modern web technologies with computational biology to bridge the gap between software engineers and biologists. It provides a valuable tool for students, researchers, and professionals, showing how technology can simplify complex interdisciplinary problems.

Key lessons learned include the importance of scrum meetings, which helped the team address challenges early, maintain steady progress, and foster collaboration. Moving forward, addressing current limitations and expanding visualization capabilities will further enhance the platform's utility and impact.

ACKNOWLEDGEMENTS

At the closing end of this project, we'd like to take this opportunity to thank our professors and fellow classmates for providing invaluable feedback during critical stages of development. Without it, we wouldn't have been able to implement this project to our fullest extent. A special thanks to Virginia Tech for providing us with the opportunity and resources to complete this task.

REFERENCES

Sinnefa. 2019. Three-sequence alignment using dynamic programming. GitHub repository. Retrieved December 17, 2024, from https://github.com/Sinnefa/three-sequence-alignment-using-dynamic-programming.

Wikipedia contributors. 2023. BLOSUM. In Wikipedia, The Free Encyclopedia. Retrieved December 17, 2024, from https://en.wikipedia.org/wiki/BLOSUM.

Grant, B. 2020. Interactive demo for Needleman–Wunsch algorithm. Retrieved December 17, 2024, from https://bioboot.github.io/bimm143_W20/class-material/nw/.