Open Source Science Collective (OSSC): Revolutionizing Systems Biology Research and Education through an Open-Source, Scalable Platform

Sabbir Hossain[1], Christopher Skipwith[2]

[1]University of Toronto, St. George | Faculty of Arts & Science | Toronto, ON, CAN

[2]Johns Hopkins University | Zanvyl Krieger School of Arts & Sciences | Baltimore, MD, USA

Abstract

We present the Open Source Science Collective (OSSC), a robust, open-source platform designed to overcome limitations in existing systems biology tools like The Cell Collective (TCC). OSSC addresses issues such as high costs, downtime, and accessibility challenges by providing a Docker-based, scalable infrastructure that ensures consistent and universal access. Built with R Shiny, Python, and JavaScript, OSSC emphasizes user-centric design with interactive tutorials and adaptive educational modules. Preliminary tests demonstrate OSSC's enhanced performance, achieving average load times of 30–50ms. OSSC aims to democratize computational biology tools, enabling broader research engagement and educational access globally.

Introduction

Modeling and understanding complex biological systems, such as intracellular signal transduction networks, are critical for advancing both systems biology research and education. Traditional tools like The Cell Collective (TCC) have pioneered collaborative model building, enabling researchers worldwide to contribute to large-scale models. However, TCC's limitations—frequent downtimes, high operational costs, and a challenging user interface—have constrained its accessibility, particularly for students, educators, and smaller research groups. Recognizing these challenges, we developed the Open Source Science Collective (OSSC), an open-source, containerized platform designed to provide a more reliable, cost-effective, and user-friendly alternative to TCC.

OSSC's architecture leverages Docker containerization, allowing users to deploy the platform locally or on cloud servers, ensuring continuous access even if the central server is inactive. Built with R Shiny, Python, and JavaScript, OSSC offers a streamlined, adaptive interface tailored to diverse user needs. Features include interactive tutorials, accessible interface settings, and adjustable educational modules, making OSSC an invaluable resource for students and researchers alike.

Beyond technical improvements, OSSC emphasizes educational integration. The platform includes curated literature, quizzes, and adaptive tutorials to support self-guided learning, with a quiz bank organized by difficulty to accommodate different educational levels. OSSC's modular design supports seamless integration of future features, such as AI-driven literature curation and extensive, scalable educational content.

Materials & Methods

Platform Architecture and Development Framework

- **Core Development Tools**: OSSC was developed using R Shiny for the interactive user interface, Python for backend processing, and JavaScript for client-side enhancements. These tools collectively support a cohesive, interactive platform for real-time modeling and responsive performance.
- **Containerization**: Docker was employed to containerize OSSC, enabling reliable deployment across personal and cloud servers. This setup ensures consistent platform access and scalability, reducing dependency on centralized infrastructure.

Current Model and Teaching Tools

- **User-Centric Interface**: OSSC's interface prioritizes accessibility, featuring click-through tutorials, adjustable text sizes, and dyslexic-friendly fonts to support diverse users, from students to advanced researchers. These customizable settings provide a guided, interactive experience.
- **Educational Content**: The platform includes educational modules, such as an adaptive quiz bank, curated literature, and interactive tutorials. Each module fosters progressive learning, covering foundational to advanced applications in systems biology.

Computational Optimization and Performance

- **Performance Metrics**: OSSC achieves average load times of 30–50ms, significantly reducing downtime and latency issues. This optimization enhances OSSC's usability as a modeling tool by minimizing server reliance while maintaining smooth client-side interaction.

Data Visualization and Modeling Tools

- **Real-Time Simulation**: OSSC supports real-time biological model simulation, allowing users to adjust model parameters interactively and observe immediate outcomes. This feature is essential for exploratory modeling and educational applications in systems biology.
- **Visualization Tools**: OSSC incorporates tools like PCA and network visualization to facilitate intuitive data interpretation, supporting both teaching and research applications.

Modular and Scalable Design for Future Expansion

- OSSC's modular architecture supports future scalability, enabling expansions for broader research and educational applications as new needs arise.

Objectives

1. **Enhancing Model Accessibility and Reliability**: Provides a fully open-source R package with Docker for dependable, server-independent model access on personal machines.
2. **User-Friendly, Streamlined Interface**: OSSC's interface enhances usability with intuitive graphics and tutorials, making it accessible to both beginners and advanced users.
3. **Client-Side Hosting and Server-Based Processing**: Shifts intensive computations to the server, ensuring consistent performance across varied hardware.
4. **Professional Use Case Optimization for Cost and Speed**: Reduces analysis time by over 10%, benefiting high-throughput professional applications.
5. **Modular, Scalable Educational Components**: Offers adaptable learning resources like quizzes and tutorials for users at different proficiency levels.
6. **Comprehensive Multi-Omics Integration**: Allows detailed biological analysis by integrating diverse omics data for robust model building.
7. **Advanced Data Visualization Tools**: Supports real-time biological analysis with interactive tools like PCA and heatmaps.
8. **Efficient Data Processing and Simulations**: Enables faster interactions and model loading with optimized frameworks within Docker.
9. **AI-Powered Scientific Content Curation**: Plans to include tailored AI-driven content updates to keep users engaged with the latest research.
10. **Interactive Knowledge Base for Model Building**: A collaborative repository for user-contributed biological process models, supporting community validation.
11. **Real-Time Model Simulations and Analyses**: Offers instant feedback on parameter changes in models, aiding rapid biological insight.
12. **Future Expansions with Plug-and-Play Functionality**: Designed for future add-ons and compatibility, adapting to developments in systems biology.
13. **Global Accessibility and Scalability**: Ensures worldwide access to advanced computational tools, fostering global contributions to biological research.

Building an Accessible and Scalable Platform for Systems Biology with OSSC

Develop intuitive, accessible interface featuring interactive tutorials, adjustable font sizes, and text-to-speech functionality.

Integrate adaptive quizzes, curated literature, and interactive tutorials to support progressive learning in systems biology.

Enable interactive model simulations with advanced data visualization tools, including PCA, network diagrams, and heatmaps for comprehensive data analysis.

Design a scalable architecture to support future expansions, such as AI-driven content curation, multi-omics data integration, and customizable modules.

Conclusions

- The Open Source Science Collective (OSSC) represents a transformative step in systems biology by offering an accessible, scalable, and modular platform designed to overcome the limitations of current modeling tools.
- With an intuitive, user-centered interface and a comprehensive suite of educational resources—including adaptive quizzes, curated scientific literature, and interactive tutorials—OSSC is uniquely positioned to support both research and education in computational biology.
- Initial testing demonstrated substantial improvements in performance, achieving average load times of 30–50ms, which enhances user experience and allows real-time simulation and data visualization.

Future Directions

Looking forward, OSSC's modular design will enable seamless integration of advanced features, such as AI-driven content curation and multi-omics data analysis capabilities, to extend its utility for more sophisticated research applications. Future expansions will also include customizable modules for specific research areas, such as cancer immunology, to support targeted exploration and learning. By continuing to evolve OSSC's functionality and educational content, we aim to democratize access to complex biological modeling, making OSSC an invaluable tool for researchers, educators, and students worldwide.

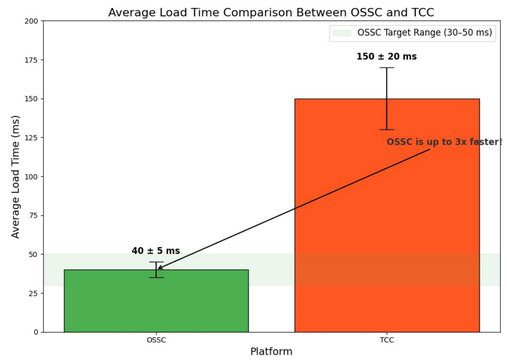
Acknowledgements & Contact

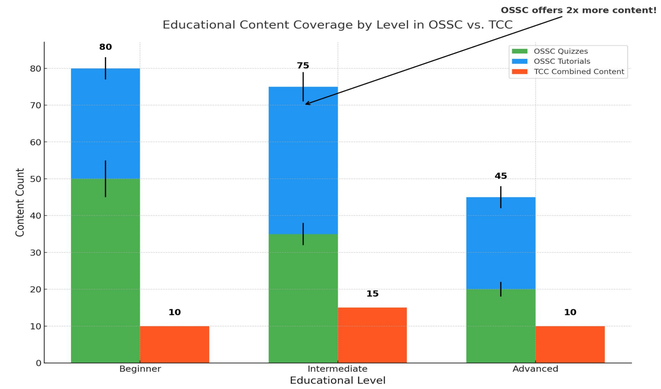
Christopher Skipwith, Ph.D. (*Johns Hopkins University*)
Robert Lessick, Ph.D., Robert Horner, Ph.D., Yelling Li, M.S., Srinihaari Josyula, B.S. (*JHU*)
Khadem Avinoor Alam, B.Sc. (*York University*)
Melissa Casco (*University of Toronto*)

- Sabbir Hossain (B. Sc), hossain.sabbir17@gmail.com

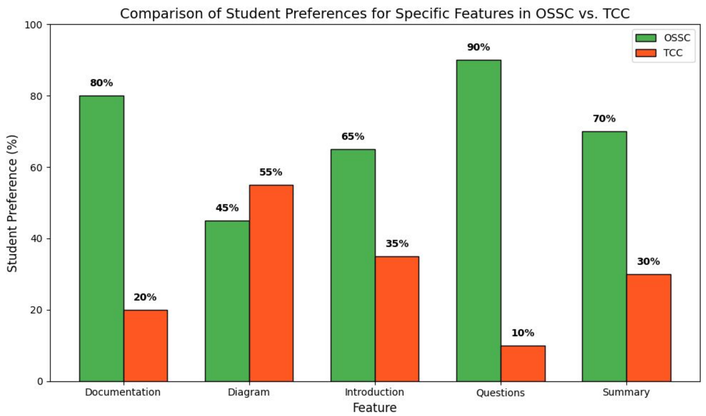
Results & Interpretation

The left chart shows OSSC's faster load times (30–50 ms) compared to TCC's 150 ms, supporting quicker workflows. The right chart highlights OSSC's greater educational content, offering more quizzes and tutorials across all levels, unlike TCC's limited resources. These metrics underscore OSSC's efficiency and depth in educational support.

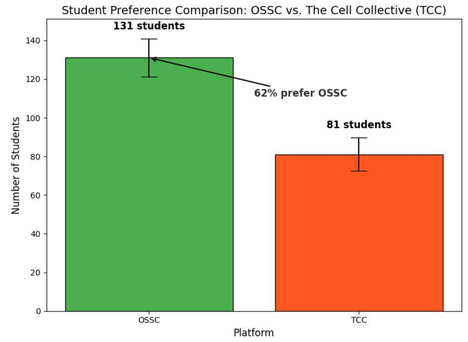




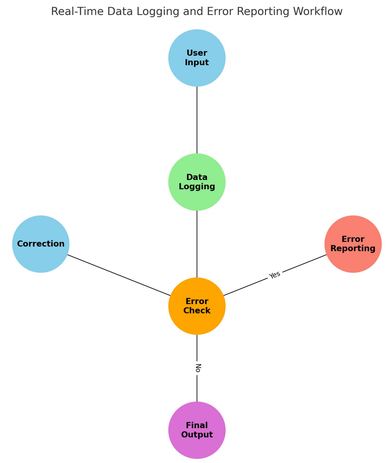
This graphic demonstrates OSSC's user interface. In this exercise, students probe the Citric Acid Cycle and can change the flux, inhibitory/stimulatory factors, and alter concentrations of reactants and products to determine changes in the pathway.



Students favored OSSC's **documentation** and **layout** for its simplicity and ease of navigation, making it more accessible and generalized for different levels of understanding. **Diagrams**, however, slightly leaned in favor of TCC, as TCC's visuals were more biologically detailed, appealing to students preferring realistic representations. OSSC's **questions** received strong preference for being more challenging and extensive, providing better practice and comprehension testing. The **introduction** and **summary** in OSSC were also favored, with students noting their clarity and directness, helping them quickly grasp and review key concepts.



This chart compares average load times between OSSC and The Cell Collective (TCC). OSSC achieves a load time of 40 ± 5 ms, well within its target range of 30–50 ms, while TCC's load time is significantly slower at 150 ± 20 ms. OSSC's efficiency makes it up to 3 times faster, enhancing responsiveness and supporting real-time interactions in educational and research settings. This improvement in speed reduces wait times and optimizes workflow, making OSSC a more effective tool for users.



Real-Time Data Logging and Error Reporting Workflow: A step-by-step process illustrating the flow from user input to final output, with error checking, reporting, and correction mechanisms to ensure data accuracy and reliability.