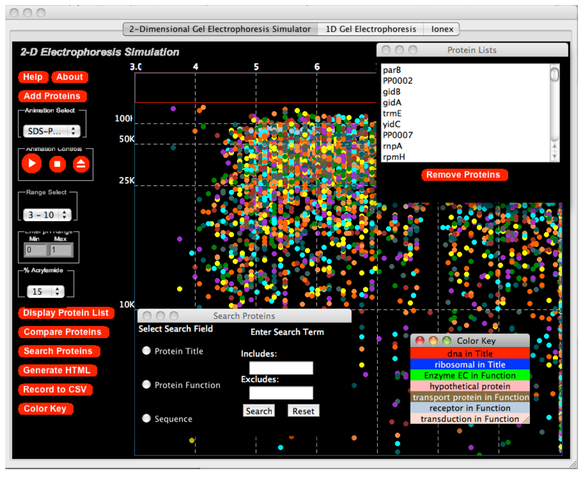
Developing and Sustaining Scientific Software Applications for Biochemistry Education

Amanda Fisher[1], Kyle Dewey[2], Paul Craig[1]. [1]Chemistry, [2]Biological Sciences, Rochester Institute of Technology, Rochester, NY

Abstract # 903

The separations simulations team at RIT has been developing interactive simulations of biological separation processes for over 10 years. Simulations of 1D electrophoresis, 2D electrophoresis and ion exchange chromatography have been presented at ASBMB conferences during that time. The simulations have recently undergone a number of significant changes. Our presentation will detail the challenges overcome in dealing with expanding data resources, advances in Internet technology and the options in computing platforms and operating systems. Early simulations dealt with very limited datasets based only on the Protein Data Bank format; recent developments include Genbank and FASTA formats. First attempts were designed to work only in Windows; this has been expanded to include Mac OSX, Linux and the popular web browsers in all of those platforms. Early programming was done in C++ and Java AWT, which were both platform dependent; our current approach is to develop applications using Java Swing, which in most cases leads to applications that operate on the most common operating systems and browsers. Support for this work has been provided by the RIT Colleges of Science and Computing and Information Science and by the Merck/AAAS Undergraduate Science Research Program.

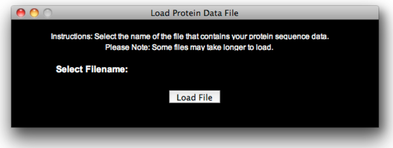


The original JBioFramework 2D Electrophoresis simulation on a Macintosh computer.

To prepare JBioFramework for use as a teaching tool:

Goal	Task
Display a standard GUI across all platforms	Convert Java AWT components to Java Swing, use relative vs. absolute positioning.
Allow for easy installation	Simplify file structure
Increase extensibility	Change to object-oriented program design
Be able to search for proteins online	Use GenBank (GBK) and FASTA file format

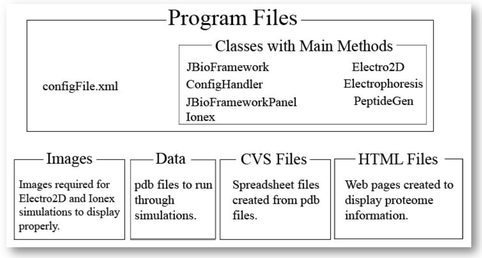
Original Problems with AWT:



The load protein file window on a Macintosh system

Using Java AWT components and absolute positioning in a graphical user interface (GUI) can lead to inconsistencies in display across platforms- a Windows user will see a different display than a Macintosh user. Thus, a school using an unsupported platform may have seen an unpolished or confusing GUI.

Original JBioFramework File Structure:



Files structure needed to install original JBioFramework

The original file structure required many folders and program files that could be activated independently of the main program but were not fully functional by themselves.

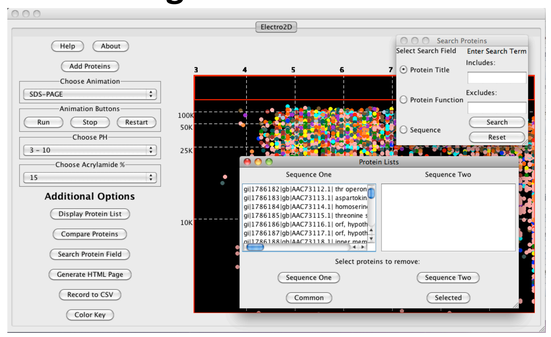
Steps to Add a New Simulation to original JBioFramework:

1. Create a simulation that displays inside of an AWT container component.
2. Add new simulation container to JBioFramework .
3. Add specifics of new simulation to configuration file in HTML code.

Original File Type Supported by JBioFramework:

The PDB file format contains information on the three dimensional locations of atoms in a protein. While the simulations could run using PDB format, implementing protein sequence searches using it would be cumbersome and unreliable.

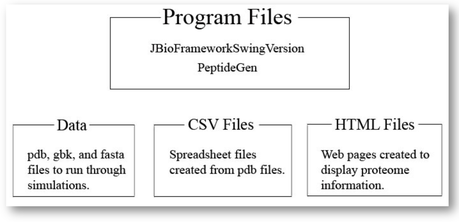
Current JBioFramework with Java Swing:



Current JBioFramework on a Macintosh computer

With a Java Swing GUI, JBioFramework will display the same across almost all platforms, ensuring every student sees an informative GUI no matter what platform they use.

Current JBioFramework File Structure:



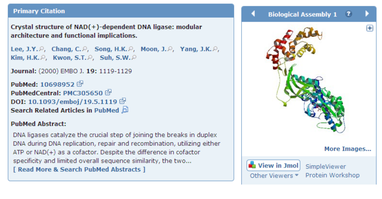
Files structure needed to install current JBioFramework

The current file structure is smaller and easier to set up and maintain.

Steps to Add a New Simulation to current JBioFramework:

1. Create a simulation that displays on a JPanel in the JBioFramework main class tab frame.

New File Types Supported by JBioFramework:

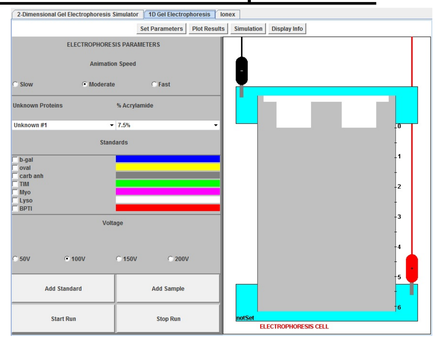


PDB search for 1dgs (DNA Ligase)

The FASTA and GBK file formats can contain either protein or genome sequence information. The GBK format also contains pdb entry numbers, allowing reliable implementation of protein searches.

Future Goals

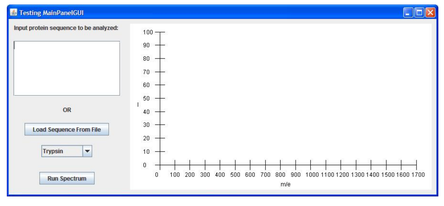
- Update 1D Electrophoresis



Original 1D Electrophoresis simulation

1D Electrophoresis needs to be converted from Java AWT and absolute positioning to Java Swing and relative positioning in much the same way the 2D Electrophoresis simulation was, so that students everywhere will see the same GUI.

- Add a tandem mass spectrometer simulation



Current simulation of tandem mass spectrometer.

The tandem mass spectrometer simulation is currently in development. It is being coded in Java Swing and utilizes JBioFramework's new easy extendibility.

- Use JBioFramework as a teaching tool

Dr. Paul Craig is currently using the 2D Electrophoresis in his proteomics and separations classes at RIT, and will be implementing the 1D Electrophoresis, Ionex, and Tandem Mass Spectrometer simulations in the future.

References and Acknowledgements

- RIT College of Science, GCCIS at RIT, Office of the Vice President for Research at RIT
- Merc/AAAS Undergraduate Science Research Program
- Protein Data Bank, JavaWorld magazine
- The RIT JBF team: Jill Zapoticznyj, Adam Bazinet, Janine Garnham, Richard Rodrigues, John Manning