# Mason Bair - Choose Ohio First - PyImageJ

KENT STATE UNIVERSITY

## Abstract

The purpose of this research experience was to learn more about various pieces of technology and implement them in a real-world situation. The research was focused on working with technologies such as Flask, Ray, Angular, Docker, and Linux. These technologies were then used in conjunction with PyimageJ and GROMACS, cell and protien visualization softwares, to help refine and manipulate these images while the software ran on a server.

## Technical Considerations

To begin the research, the PyimageJ software runs off a few different programs, namely Flask, ImageJ, Ray, XPRA, and Angular. Flask helps the software run on a server, ImageJ does the image analysis, Ray manages the power consumption of running the software, XPRA helps with browser data visualization and Angular runs the frontend website.

GROMACS is built off of a complex set of Python scripts, that when put together allow for completely automating the protein manipulation process for researchers. These scripts utilize Flask, Ray and XPRA for api and backend communication, job submitals and protein visualizations

## Research Steps

This project involved me implementing a pre-built PyimageJ Image Analysis application onto a server for students and professors to use.

This application is designed to put together pieces of an image taken from a microscope into one big image. This can either be done into 2D or 3D.

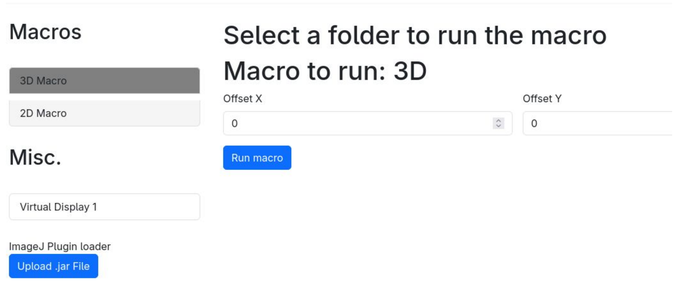
This task involved 3 different steps:

- Setting up an environment on a server to run the application off of
- Learn more about linux and a web service application called Apache.
- Host the applications frontend off of Apache for the world to see.

Then once professors could submit jobs for PyimageJ analysis, I switched over to setting up and automating GROMACS Protien analysis.

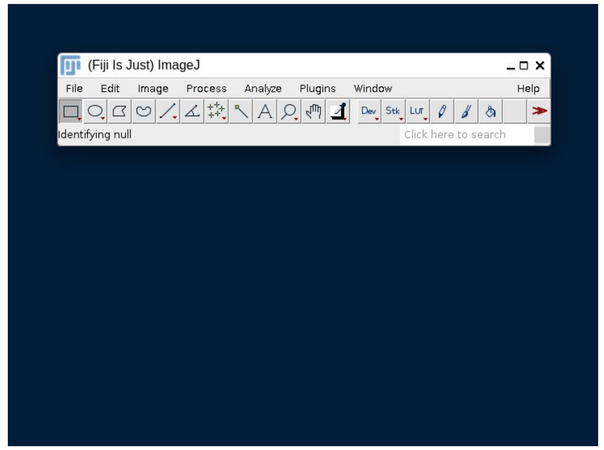
This helps save time for researchers working with Molecular Dynamic Simulations.

Some steps of this automation requires a visualization and manual inspection of the protein. So Visual Molecular Dynamics (VMD) was implemented on the XPRA display to help visualize and provide researchers with a visual of the work so far.


## Background

When the research project started, the server was bare, so the main goal of this project was to deploy each piece of software and upgrade it for future flexibility. The reason a server was necessary is because these programs are very resource-intensive that processes large files, which can be tens of gigabytes in size. In addition, this software will also be used by researchers, and many researchers do not have easy access to a machine powerful enough. So, by using programs like Docker, and Flask, they make deploying and running these programs on a server easy and scalable.




## Future Work

For the future of this project there is a lot of work that could potentially be done.

The Biology researchers currently have a txt file with all the commands they need to run for the GMX protein analysis. So turning more of these text files into Python scripts would be some more next steps.

In addition, for the ImageJ web platform, there could be more plugins created and added for the brain cell analysis, as well as coming up with a way for multiple researchers to utilize the platform. This could include running jobs, authorization handling, and compute power management.

## Conclusion

In conclusion, I built up python scripts for automating GROMACS for protein analysis. Plus, I built up a containerized web application for running imageJ plugins for brain cell analysis jobs. All while managing and being an administrator on a research server.