AI For Discovery and Self-Driving Labs

Nicholas Poon[1]
[1]Department of Computer Science, University of Toronto, Sandford Fleming Building, 40 St. George Street, Toronto, Ontario, M5S 2E4

Background

ChemOS 2.0 is a software package that was developed to assist with the **operation of automated equipment** inside chemistry labs. Being able to integrate with semi or fully automatic labs, ChemOS 2.0 **facilitates chemical and materials experiments** by providing chemists and researchers with an incredible amount of flexibility. From running fully unsupervised experiments to accepting feedback from researchers, ChemOS 2.0 is adaptable in supporting the needs of different researchers, by allowing varying amounts of autonomy. The development of ChemOS 2.0 is a crucial step in **achieving the goal of a fully autonomous chemistry lab**.

A **self-driving laboratory (SDL)** is one such implementation of the fully autonomous chemistry lab. These labs use **artificial intelligence (AI) algorithms** to analyze experimental results to construct models to understand and hypothesize about the current experiment. For example, this model could provide recommendations on the next experiment to run, based on the researcher's goal. These recommendations can be provided to maximize the yield of a reaction, reduce the time of an experiment, minimize reactant usage, and more.

The flexibility in ChemOS 2.0 creates an incredible amount of potential by increasing the possible applications of the software. For example, the integration of the SDL with ChemOS 2.0 can **accelerate and simplify the process of discovering new compounds**. In particular, the software can assist with **drug discovery**, by **shortening the target validation and identification step** in the drug development pipeline.

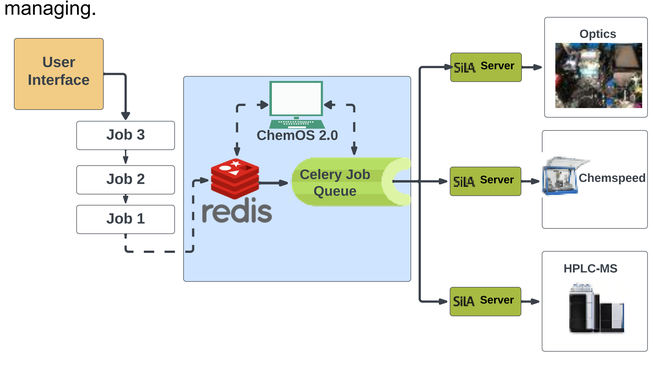
However, ChemOS 2.0 lacks a scalable and extendable **graphical user interface (GUI)** that researchers can use to interact with it. Thus, throughout the summer, under the supervision of Professor Alán Aspuru-Guzik and mentorship of Mohammad Ghazi Vakili, I worked on creating a **web application**, with Python and SQL, along with the Flask, SQL Alchemy, and psycopg2 libraries.

The web application gives a lot of control over the self driving laboratory, through ChemOS 2.0. In particular, it allows users to **submit jobs** to, **query results** from, and **query job status** from the machines in the University of Toronto's Department of Chemistry's SDL. This GUI also allows for researchers to **view the analysis and visualization of the data generated from experiments**.

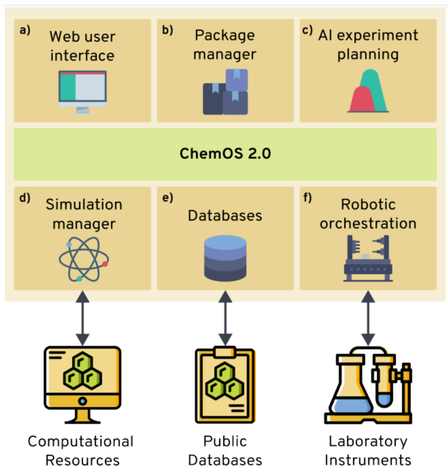
Job Queue

To submit a job to a certain machine, users must simply fill out the **Flask form** for the desired machine and then press the submit button. The job queue management system will manage the rest.

To implement the job queue, the team decided to use an open source task queue system, **Celery**, along with a caching server, **Redis**. We used Celery to give each machine in the SDL its own queue of jobs, stored in Redis, which ChemOS 2.0 and Celery are in charge of managing.

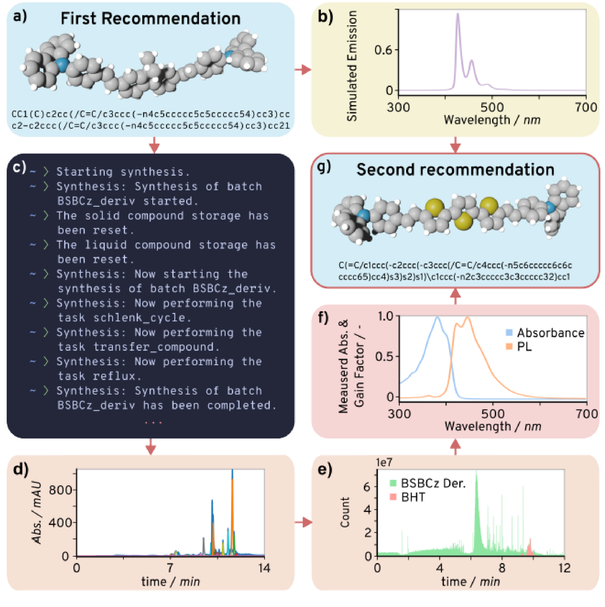


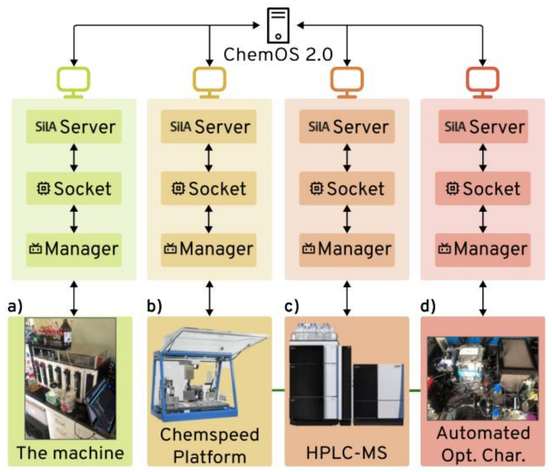
Method



ChemOS 2.0 has many different components that are built in to allow for the **seamless integration** into chemistry labs[1]:

a) A **web application** that provides a GUI for ease of use
b) **NIX Package Manager** is used to track all software versions and dependencies. This enables this software configuration to be replicated on any device.
c) Uses **Bayesian Optimization** to assist with experiment planning
d) **Density Functional Theory (DFT)** simulation manager works with our high performance computer cluster to run DFT experiments
e) **PostgreSQL database** to keep track of both simulation and experimental results
f) Uses the **SiLA2 client/server protocol** to control laboratory instruments





The implementation of a web application enables ChemOS 2.0 to be released to the public sometime in the future, or at least certain academic institutions or people outside of the University of Toronto. Currently, we just run the online platform locally when we need to use it. However, we could **deploy the online platform**, which would allow the website to be visible to public. Currently, we have a system in place for creating a new account. New users must either register with a **voucher code or get approved by the admin** before their account is created.

The diagram to the left demonstrates a **closed-loop workflow** for the synthesis of $BSBC_Z$ derivatives, with the goal to maximize the experimental gain cross section and the simulated lasing gain factor[1] :

a) Based on known observations, the **Atlas Bayesian Optimizer (Atlas)** makes a recommendation for which $BSBC_Z$ derivative to synthesize.
b) **Chemspeed** is instructed to **synthesize** and this **simulated fluorescence spectrum** is produced.
c) After the synthesis, the product is injected into the **high performance liquid chromatography - mass spectrometry (HPLC-MS)** to identify the product in the unrefined mixture.
d) The purified $BSBC_Z$ derivative is transferred into the **optical characterization platform** to assess the sample's fluorescence and absorbance spectra, along with the emission lifetime to calculate the gain cross section.
e) These are the extracted **Ion Chromatogram plots** for the HPLC-MS
f) The results from the previous 2 steps are used as inputs to **re-train Atlas and select another candidate**, which happens to be g) in this example.

In our chemistry lab, every machine has their own **SiLA2 server**. Each client can **interact** with every machine's server and **access their available actions**, through the ChemOS 2.0 GUI. This GUI along with each machine's queue act as **device-specific managers**. It is important to note that all servers are linked to ChemOS 2.0, allowing it to act as an orchestration tool to **delegate the experiment among the lab machines**. This is significant as chemistry experiments typically involve **multiple steps** to complete so there would need to be some manager to **oversee the machines in the lab** and their statuses. Thus, once a machine is done its part of the experiment, the product can be **passed to another machine** for it to complete the next step.

Job Analysis

As seen from the bottom diagram, ChemOS can **measure various pieces of data throughout the reaction**. The purpose of this page is to provide a central place to **visualize all of this information**. This allows chemists to analyze **various properties of the molecule** and the **quality of the recommendation**, provided by ChemOS 2.0. In the future, this tab could contain more information, depending on what the chemists in our lab think would be useful.

For the job analysis page of the web application, we are working on improving how the data gets analyzed and visualized. Our goal is to make the tab have an **embedded Jupyter Notebook** that would allow users to type code directly on our website to have it visualized. We also hope to provide users an option to **download this code** that they have typed in case they may want to use it in the future. As a result, we also want to provide users to **upload a jupyter notebook (.ipynb) file**. Then, the web application would simply run the file directly and then display the visualizations.

Scan To Learn More

https://www.matter.toronto.edu/
Learn about:
- Our team
- Our research
- Our publications
- Our open software and data

References

(1) Malcolm Sim, Mohammad Ghazi Vakili, Felix Strieth-Kalthoff, Han Hao, Riley Hickman, Santiago Miret, Sergio Pablo-García, Alán Aspuru-Guzik https://doi.org/10.26434/chemrxiv-2023-v2khf

Acknowledgements