ATOMS Lab Rotation Plan

Prof. Tyler R. Josephson

Spring 2023

E-mail: tjo@umbc.edu
Web: **cbee.umbc.edu/josephson/**
WebEx Personal Room: umbc.webex.com/meet/tjo
Office Hours: By Appointment

Welcome to the ATOMS Lab!

I'm excited to guide you through a rotation in the ATOMS Lab (AI & Theory-Oriented Molecular Science) over the next 6-8 weeks.

The ATOMS lab seeks to:

1. Predict adsorption and catalysis in nanoporous materials and in solution using computational chemistry tools
2. Develop artificial intelligence tools (especially symbolic regression and automated reasoning) to learn complex relationships in material performance and thermodynamic properties
3. Apply these tools to gain molecular-scale insights on processes in the environment to enable new solutions for cleaner air and water, in collaboration with experimental partners

To get a feel for what we do, your rotation will consist of a number of mini-projects that will expose you to quantum chemistry and molecular simulation software. You will write some code from scratch for data management, fitting, and visualization; transferable skills that will pay dividends whether you join this group or not. As you finish each mini-project, please record your time spent - I'll appreciate these measurements as I plan the future of this rotation program.

In the ATOMS Lab, we focus on Python and Lean, popular and up-and-coming languages for scientific computing. We access HPC (high-performance computing) resources through a Linux environment, so bash commands are also helpful (though it's good practice to limit the scripting you do with bash; Python is much more flexible). Though not required for your rotation, typesetting reports and papers in LaTeXwill be required if you join the group.

Finally, I encourage you to spend *at least* 3 distraction-free hours every week reading textbooks or papers (see the final section of this document), no matter how excited you are about writing code and running calculations. Whether deep-diving into textbooks and theory or assessing review papers on a surface level, it's important to read, and read *consistently* to grow and benefit from the work of others.

Rotation Objectives

By the end of this rotation, successful students will be able to produce quality results for three of the following tasks (selected in consultation with Prof. Josephson). Students can pursue stretch goals of their interest during their rotation, keeping in mind their relevance to funded projects.

- Pick a dataset from the NIST Adsorption database, fit the Langmuir and dual-site Langmuir isotherms to it, and plot using Python
  - Why? Coding and using theories of adsorption is essential in the ATOMS Lab
  - Stretch goal: Rediscover Langmuir using symbolic regression
- Using a chemical science database of your choice, produce a high-quality visualization with Python or Julia, write a descriptive caption, and store the data in tabular, machine-readable format (like JSON or CSV)
  - Why? Visualization is essential to understand the data we produce, and quality figures are essential for science communication. To promote data transparency and reproducibility, papers from the ATOMS Lab must include tabulated files for all results in the Supporting Information (if file sizes are reasonable for journals to host), and generating plots and tables using scripts is MUCH safer than using spreadsheets.
  - Stretch goal: Perform a meaningful regression or classification task on the dataset
- Simulate vapor-liquid equilibrium in the Gibbs ensemble using MCCCS-MN (Monte Carlo for Complex Chemical Systems-Minnesota)
  - Why? MC is a fundamental technique in the ATOMS Lab for simulating phase coexistence, gas- and liquid-phase adsorption, and predicting thermodynamic properties
  - Stretch goal: Calculate critical properties, and contribute to TraPPE validation data
- Define Raoult's Law and Dalton's Law in Lean 4, and prove the relationship obtained when both are true
  - Why? Lean is a powerful programming language that enables the computer to reason about its own code. The ATOMS Lab aims to pioneer scientific computing in Lean
  - Stretch goal: Port a theorem from mathlib to Lean 4
- Calculate the heat of combustion of methane using Gaussian
  - Why? Cluster-based electronic structure calculations are useful when we don't have accurate force fields, and are especially important for force field parameterization, chemical reactions, and solvation
  - Stretch goals: Fit the torsion potential for butane, find transition states for F–F–H and decarboxylation of butenoic acid, calculate solvation free energy and pKa of phenol

Resources

- Recommended textbooks and readings:
  - Molecular simulation
    - **Understanding Molecular Simulation: From Algorithms to Applications, by Daan Frenkel and Berend Smit, 2002**
      - For a good introduction to the field, check out chapters 1, 2, 3, 4, 5, and 8
    - Computer Simulation of Liquids, by M. P. Allen and D. J. Tildesley, 1991
  - Statistical mechanics
    - Statistical Mechanics, by Donald A. McQuarrie, 1975
    - Introduction to Modern Statistical Mechanics, by David Chandler, 1987
  - Quantum chemistry
    - Exploring Chemistry with Electronic Structure Methods, by James B. Foresman and Æleen Frisch, 2015. (Tutorial/manual for Gaussian)
  - Machine learning
    - Machine Learning Refined, by Jeremy Watt and Reza Borhani https://github.com/jermwatt/machine_learning_refined
    - Dive Into Deep Learning, by Aston Zhang, Zach Lipton, Mu Li, and Alex Smola, 2019-2020. Jupyter notebooks with working code are available at https://d2l.ai/
    - Deep Learning, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, 2016. http://www.deeplearningbook.org/
  - Python for Everybody by Charles R. Severance https://www.py4e.com/book
  - Data visualization
    - Fundamentals of Data Visualization by Claus O. Wilke (https://clauswilke.com/dataviz/)
    - The Visual Display of Quantitative Information by Edward Tufte, 2001
  - Logic: see https://en.wikipedia.org/wiki/First-order_logic and follow the links
- Essential software:
  - Anaconda (Python, data science apps)
    - https://www.anaconda.com/download/
    - https://docs.anaconda.com/anaconda/
  - Jupyter
    - http://jupyter.org/documentation
  - Gaussian
  - MCCCS-MN, Monte Carlo for Complex Chemical Systems, Minnesota
- Force fields
  - TraPPE, Transferable Potentials for Phase Equilibria http://trappe.oit.umn.edu/
  - OPLS, Optimized Potentials for Liquid Simulations http://zarbi.chem.yale.edu/ligpargen/
- Lean
  - Check out this in-progress document for resources for learning Lean

Connecting to UMBC High-Performance Computing Facility (HPCF)

UMBC researchers share supercomputer resources through the `taki` supercomputer. Learn more about the resources here. The faculty who manage this cluster have done an **excellent** job filling the website with helpful information. If you're new to working with a supercomputer, you'll find this guide to be a helpful explanation of what it means to be a good community user. More technical information about the supercomputer hardware is available here.

To get started, please request an account by filling out this form. You can use the following description when creating your account for doing rotations:

- Research Title: AI & Theory-Oriented Molecular Science
- Abstract: The ATOMS lab at UMBC uses molecular simulation and machine learning to gain molecular-scale insights on processes in the environment to enable new solutions for cleaner air and water.
- People with departmental and institutional affiliations: Prof. Tyler R. Josephson, CBEE
- Contact information for PI: tjo@umbc.edu ; 410-455-2474
- Comments: CPU resources will be used to perform Monte Carlo simulations using MCCCS-MN software and DFT calculations using ASE and GPAW.

Once you have an account, you can connect using Terminal from a Mac or using Putty from a Windows computer (download Putty here). Detailed connection instructions are available at hpcf.umbc.edu/cpu/using-your-taki-account/, along with some basic instructions and exercises for transferring files, changing permissions, etc.

When you connect to the supercomputer through the command line, you're actually writing code in the `bash` programming language. Every time you hit `Enter`, the line of code you've written is executed. `bash` isn't the most powerful computer language to write programs, but it's almost always used to interface with HPC systems, so learning several `bash` commands will help you navigate. To learn more, check out this tutorial, and of course, Google is your friend when you have questions about how to do one thing or another. Later on, you can automate these commands by writing your own `bash` scripts. However, I generally recommend using a more versatile language like Python when you're writing programs to set up and analyze your jobs, and only reverting to `bash` right before you submit each job to the cluster.

Installing MCCCS-MN

Get the most recent copy of code from Prof. Josephson (it is maintained by the Siepmann group of the University of Minnesota, and the latest version is not available to the public). Once you understand the code, if you want to add new features, fix a bug, or clean up the manual, feel free! Prof. Josephson can connect you with the Siepmann group and get you access to the master version. You'll be credited as an author of the code in all future releases. But most of the time, we won't need to change the code in order to do neat things with it.

In MCCCS-MN/doc/, you'll find manual.tex, which contains the most updated version of the user manual. The first chapter has installation instructions. I've modified those ever-so-slightly below, to show you the commands that will work to compile the code on `taki`.

After you've got the code, create a folder **in your home directory, not in your scratch directory** to store your executable. Note that it's helpful to date the folder, to better track multiple compilations of MCCCS-MN.

```
mkdir CURRENTDATE-exe-MCCCS

cd CURRENTDATE-exe-MCCCS

module load CMake

module load intel

FC=ifort cmake -DDOUBLE_PRECISION=ON -DUSE_MPI=OFF -DUSE_OPENMP=OFF -DUSE_OWN=ON
-DCMAKE_BUILD_TYPE=RELEASE ~/tjo_user/MCCCS-MN/

make -j8 # build using 8 cores
```

Note there are multiple options to select in the FC=ifort line. Always use double precision (the speedup from single precision is not worth inaccurate calculations due to round off errors), and DUSE_OWN=ON (a neat option that uses a custom random number generator and trigonometry calculator that saves CPU time). The other options are described in the manual.

GitHub

GitHub hosts our public repositories for the software we create, as well as for each paper the group publishes. Learn more about how to use GitHub in this tutorial.