ATOMS Lab Policies

Prof. Tyler R. Josephson

Fall 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!

The AI & Theory-Oriented Molecular Science (ATOMS) Lab at UMBC 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. Use automated theorem proving tools to write software for molecular simulations (and scientific computing more generally) that's free of bugs
4. 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

Rotations

See the ATOMS Lab Rotation Plan for details on rotations. New undergraduate researchers may also use these projects as a guide for which skills to develop and tools to learn.

Coding Languages

In the ATOMS Lab, we use Python and Julia, popular and up-and-coming languages for scientific computing, respectively. We are also exploring Lean 4, a unique programming language that also enables automated theorem proving. 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). Manuscripts must be typeset in LaTeX prior to submission – get started with LaTeXon Overleaf.

Working hours, productivity, and time off

- Grad school is not a sprint. It's not even a marathon; it's a quest. Pace yourself, take care of yourself, brace for the unexpected.
- Track your time! How much time do you spend each week 1) reading papers, 2) writing, 3) attending meetings, 4) attending class and working on homework, 5) doing TA responsibilities, 6) responding to emails, and 7) doing research on each of your projects (this means writing code, setting up, starting, and analyzing simulations, and making figures)? Keeping track of the time you spend each day and each week is the first step to identify ways to improve your time management.
- I recommend Clockify, a free online tool for tracking your time. This can make electronic time sheets easier for group members paid by the hour, and also as a general productivity tool for **everyone** to use, myself included. Your future employer in industry may use a tool like this to manage your projects. For example, you may be assigned 50% to Project A, 30% to Client B, and 20% to personal projects, and you would be expected to distribute your time accordingly.
- All researchers are entitled to two days of leave when they are on their periods. You don't have to take them if you don't want to, but they are there for you. You don't have to explain it to me or anyone else; you can rest at home.
- Anyone who is sick should stay home to keep others healthy. Participation in meetings or work is not expected; just let Prof. Josephson and your team members know.

Communication

WebEx Teams

Please use WebEx Teams, especially when discussing research projects! It enables natural organization around topics, and keeps your inbox from cluttering. It also helps new people to get up to speed on projects, since they can review all prior correspondence.

WebEx Teams is a great way to organize group communications and collaborate. This may feel like a mix of asynchronous communication (like emails) and synchronous communication (like instant messaging), and in many ways, it's a platform that handles both with ease.

However, I want to set clear expectations - this should be primarily viewed as a medium for asynchronous communication. If you receive a message, you are neither obligated to read nor to respond immediately (and you shouldn't expect this when you're sending, too). Of course, if you happen to be online at the same time as another person and you enjoy a quick back-and-forth, that's wonderful, but the expectation should not be that you are constantly connected. I also encourage you to "batch" your emails and messages into certain times of day, so at other times of day, you can more easily focus when you want to dig deep in writing code or preparing a manuscript or reading the literature.

Feel free to use this tool how you prefer. Please know that I have this installed on my phone, but I manage my notifications, so I'm not notified unless I'm already looking at my phone. You don't need to be as connected as me, however; it's really okay if you are not pestered by notifications from work when you're not working.

Weekly reports

All group members are to prepare brief, weekly reports on their research progress, and share them with Prof. Josephson via the Notion platform.

Authorship

Authorship and order of authors for each manuscript will be determined by Prof. Josephson and any additional Principle Investigators involved in the work. Authorship is earned by someone who conceives of the project, performs simulations or experiments, analyzes results, produces figures, or writes the paper. All authors must read, proofread, and sign off on the final version of the manuscript.

Group meetings

Please send materials the night before

This makes meetings much more efficient, because reading, learning, and reviewing can be done individually, and group time can be spent discussing and collaborating.

Ask questions

The "price of admission" to group meeting is that you ask one question. This helps you learn, and it helps the presenter understand how to present better. By doing this, you'll also gain valuable practice in preparing and asking questions of presenters. This will become helpful when attending conferences and listening to talks outside the group. Indeed, some questions are sillier than others, and this is something you'll learn with practice and observation. But don't worry about it – the ATOMS Lab is a safe space to ask anything, especially considering the diverse areas of expertise of the people in the group.

Professional development

Social media

Group members should create and maintain a professional social media presence. This can be a tool to support your ongoing networking activities and your future job search. Here's a great article on how to take your LinkedIn profile to the next level (tailored to grad students in STEM).

I've also found Science Twitter to be a helpful way to learn about new papers and share my own work and ideas with the community. Check out #compchem, #ChemTwitter, #symbolic, and #MachineLearning to find people sharing content in our research areas. Conferences and events will also use hashtags to organize related content. The ATOMS Lab has a student-run Twitter account, @ATOMSresearch; reach out if you'd like to be involved.

Networking

Professional networking can sound intimidating, but it helps to frame it as "making professional friends." This is so important, that **I require group members to participate in at least one informational interview each semester**, starting with their first semester. Practicing this, and starting early, will help develop the skills you need to build your network later, and also get your professional network jump started. Here's a nice article on what an informational interview is, and how you should prepare for it.

Virtual Lab Notebook

All group members are expected to keep an electronic lab notebook. This is a ∼daily record of the research work that you do. Why is this important? Let me tell you a story:

Lilian Lam Josephson, while she was getting her PhD at the University of Delaware, performed microscopy experiments, collected videos, and processed the videos with an analysis code. After collecting the data, analyzing it, and plotting it, she found a very interesting result that was to form a core part of the narrative of a paper. Several months later, another student in the lab attempted to replicate this, but consistently found a different result after several attempts. So she needed to figure out what happened.

What did she do? She opened her lab notebook. She opened up her calendar. She saw that, on the day which she performed the experiments and analyzed the results, she had been training a student on using the analysis code. In the morning, she analyzed 'experiment A' and in the afternoon (after the training), she analyzed 'experiment B' (the one with the anomalous behavior. Remembering that she had demonstrated during training the ability to switch the code to analyze 2D or 3D trajectories (which shifts the data by a factor of 2), she realized she hadn't switched it back before analyzing the data in the afternoon, which caused the discrepancy. When analyzed properly, it could now be seen that the data was now reproducible. Everyone was happy, the paper was submitted and published.

What's the lesson here? **Be organized. Take notes. Your future self will thank you.** When you set up your simulations, changes you make will be reflected in your input files, and you can theoretically go back and retrace your steps, so long as you don't delete anything. But it's far better to make a note of what you're thinking and what you're doing while you're at it.

ELN Software

After surveying several electronic lab notebook (ELN) options, we recommend Notion for maintaining your ELN.

Data Management Plan

Molecular simulations, electronic structure calculations, and data analysis are performed on the taki supercomputer at the UMBC High Performance Computing Facility, as well as on local workstations. Data from completed jobs will be routinely backed up on local workstations and external hard drives.

Prior to manuscript acceptance, all files needed to reproduce the results will be made available to the public via a GitHub repository in the ATOMS Lab GitHub. For electronic structure calculations, optimized geometries and corresponding input and output files will be shared. For Monte Carlo simulations, the input and output files for representative jobs, as well as the source code for the version of MCCCS–MN used, will be likewise shared via GitHub. This enables reproducibility and transparency, without requiring online hosting of potentially many TB of raw simulation outputs. Raw simulation data will be retained on external hard drives, available to be shared upon request. An example of this from the PI's recent work is here. Code associated with submitted publications will be shared publicly on the ATOMS Lab GitHub, along with machine-readable files for the data on the data input and the equations output from the algorithm. All published data figures will be accompanied by tabular, machine-readable data, available as Supporting Information.

Researchers in the ATOMS Lab are required to maintain an electronic lab notebook, hosted via Notion, to maintain a record of their activities as they develop software and run simulations, just as physical lab notebooks document research activities for experimentalists.

Backups

Back up your data. Not much could be more tragic than losing several months of manuscript writing, simulation results, or code modifications.

Each ATOMS Lab member is responsible for maintaining their own data. We've identified Google Drive, as the preferred cloud storage solution for backing up data. We'll also invest in physical backups, as well, in the form of external hard drives.

When running simulations on Taki, be especially conscientious of the nature of the different places in which data is stored. We'll run most of our calculations using "Research Storage," and this is not automatically backed up. It's possible to write scripts that send data from completed jobs outside of Taki to your local machine to be instantly backed up, if that's a helpful solution for you.

Sharing large files

If you're working on UMBC's supercomputer, know that we have a group shared folder that can facilitate transferring files among group members.

GitHub

The ATOMS Lab has a GitHub account for the group to host 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.

This is also the location for our group website, which is hosted through GitHub pages.

Netiquette

Treat others as you would like to be treated.

Remember that you're communicating with a real person, and not merely a computer screen. Follow the same standards of behavior online that you follow in real life. As professional interactions increasingly become online, learning to be professional online is a critical career skill – the person behind the other screen will be your colleague, your boss, or your customer.

- Use respectful language
- Use normal capitalization, grammar, and spelling for professional communication.
  - Mistakes make you look bad! Spell-checkers and tools like Grammarly can catch distracting errors.
- Remember that tone of voice and gestures are not communicated through text. Especially refrain from sarcasm, which can be easily misinterpreted, and which usually conveys an unprofessional attitude.
  - Emoticons can be helpful and fun, but less-common ones are ambiguous, so know your audience. Different generations have different standards.
- Respect other people's time
  - Use online calendars and tools like WhenToMeet to coordinate online meetings.
  - Make an agenda, start on time, and end on time.
  - Respond to emails promptly; within 48 hours is usually acceptable. A short note "This will take me some time, I'll get back to you by X date" helps set expectations.
- Video chat like a pro
  - Become fluent in video conferencing software, so you don't contribute to any technical issues.
  - Mute your mic when you're not talking.
  - Only use video when everyone's bandwidth can handle it.
- Secure your workstation
  - Never share personal user ID or password information.
  - Don't use simple passwords, and don't reuse passwords. Password managers like LastPass are great tools for generating and storing highly secure, random passwords.

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
  - Workflow tools (good to learn to build your own for simple tasks)
    - MoSDeF, Molecular Simulation and Design Framework, https://mosdef.org/
    - signac, Data and Workflow Management for computational research, https://signac.io/
- 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/

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 ; Campus phone: 410-455-2474 (email is best)
- Comments: CPU resources will be used to perform Monte Carlo simulations using MCCCS-MN software, DFT calculations using ASE and GPAW, and for running ML programs with Python and Julia.

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.