Samuel D. Lotz – Curriculum Vitae

Contact Information

104 Pinehurst Dr.
Cranberry Twp., PA 16066
USA

Phone: (724) 421-6138
E-mail: samuel.lotz@salotz.info
LinkedIn: `linkedin.com/in/salotz`
Website: `https://salotz.info`
Github: `https://github.com/salotz`
ORCID: `https://orcid.org/0000-0001-6159-615X`

Education and Experience

Roivant Sciences, New York, NY

Senior Software Engineer, Roivant Sciences (acquired Silicon Therapeutics Feb 2021)
Feb 2021– November 2022

In this position I had a variety of responsibilities ranging from:

- Lead engineer and system integrator on computational drug discovery pipelines for prediction of degrader molecule efficacy and ternary complex formation.
- Scientific expertise and contributions to weighted ensemble (WE) molecular dynamics (MD) workflows as well as various cheminformatics and data analysis problems.
- Developed multiple internal Python libraries and applications for:
  - Highly-scalable distributed analysis framework to be used across a variety of enhanced sampling MD based data sources generated in computational discovery pipelines.
  - Hybrid cloud/on-prem HPC orchestration framework that enabled efficient and scalable workflows that were authored primarily by contributing researchers. Workflows were containerized and included highly-optimized builds of critical software such as Amber, Gromacs, RDKit, Numpy, igraph, etc.
- Lead DevOps engineer building out the custom CI/CD pipelines (Gitlab CI/CD) for testing, packaging, and library templates. Pipelines were partially containerized running on Gitlab managed runners, and internal Kubernetes cluster (GKE), and our on-prem HPC cluster. Challenges included managing builds for dozens of complex high-performance software dependencies, enabling development by both the software engineering as well as the domain scientists, targetting a wide variety of deployment targets (native binaries and packages via Pip, Conda, & Spack, Docker for cloud, and Singularity for HPC). In this work I wrote and managed pipelines for the following tasks: performing builds of software dependencies, deployment pipelines for general purpose "environments" used by exploratory analysis deployed to a shared HPC system for multiple package managers (conda and Spack), deployment and testing of a repository template used by a large team to independently develop individual software packages, and the generalized templated pipeline for performing QA, testing, packaging, and automated release managment of all software packages.
- Mentoring and educating scientific contributors on bespoke computational workloads to both translate scientific ideas into automatable code and allow them to scale to an industrial scale.
- Collaborating closely with the HPC operations team to ensure scientific contributors workloads were deployed without disruption. I also contributed to planning of new HPC operations priorities to make sure the scientific teams needs were being met and would be provided for given upcoming requirements.

Michigan State University, East Lansing, MI

PhD, Department of Biochemistry and Molecular Biology
July 2014– March 2021
Advisor: Dr. Alex Dickson

Thesis Title: Studies in Ligand Unbinding Transition State Plasticity for Kinetics-Oriented Drug Design
The goal of my thesis was to use enhanced sampling techniques alongside molecular dynamics (MD) of drug unbinding to facilitate kinetics oriented drug design and discovery. The main enhanced sampling techniques used were variations on weighted ensemble (WE) including WExplore and REVO. I have used other machine learning analysis methods such as Markov State Modelling (MSMs) and transition path theory (TPT) to identify unbinding transition states and identified intermolecular interactions likely to play a role in determining unbinding kinetics.
I completed analyses of ligand unbinding transition states that allowed for the development of novel drug targets for the target soluble epoxide hydrolase (sEH).
To achieve this I have authored, in addition to many custom analysis scripts, two major software modules (see Software Projects for more detail). The first is a framework for running WExplore type WE simulations (`Wepy`) to enable quick simulation set up and analysis. Another was `Mastic`, which is an extensible framework for profiling of intermolecular interactions (of any kind) that is suitable for large scale data or being embedded in other software. I also authored libraries such as `Geomm` which was a Numpy-style library used for high-performance calculation of various geometric features relevant to biomolecules.

Graduate Assistant, Department of Biochemistry and Molecular Biology
January 2015–December 2015
Advisor: Dr. David Kramer

- QTL mapping of photosynthetic traits in *Arabidopsis thaliana* ecotypes in response to cold stress
- Coauthored one publication

Plant Genomics Research Experience for Undergraduates
June 2013 – August 2013
Advisor: Dr. David Kramer

- Title: The Role of an NAD(P)H Dehydrogenase Complex Associated Protein in Regulating Cyclic Electron Flow Under Light-Limiting Conditions
- Presented results at the 16th International Congress on Photosynthesis Research at St. Louis, Missouri, USA and the Mid-SURE undergraduate research symposium at Michigan State University.

Slippery Rock University, Pennsylvania, USA, GPA: 3.76/4.00

B.S. Biology, (Minor: Mathematics, Chemistry; Specialization: Bioinformatics)
August 2010 – May 2014

Undergraduate Biology Research, Dr. Nicole Dafoe, Dept. of Biology

- Title: Drastic Systemic Protein Increase in Wounded Plant Stems
- Presented at CPUB (Commonwealth of Pennsylvania University Biologists) annual conference

Publications

- Dixon T, **Lotz SD**, *et al.*, Izaguirre JA* (2020) Predicting the structural basis of targeted protein degradation by integrating molecular dynamics simulations with structural mass spectrometry *Nature Communications* 13 (1): 1-24 https://www.nature.com/articles/s41467-022-33575-4

- **Lotz SD** (2021) Studies in Ligand Unbinding Transition State Plasticity for Kinetics-Oriented Drug Design.
https://salotz.info/resources/thesis.pdf
Also available on request.

- **Lotz SD** and Dickson A* (2020) wepy: A Flexible Software Framework for Simulating Rare Events with Weighted Ensemble Resampling *ACS Omega* 5 (49): 31608–31623 https://pubs.acs.org/doi/abs/10.1021/acsomega.0c03892

- Dixon T, **Lotz SD**, and Dickson A* (2018) Predicting ligand binding affinity using on-and off-rates for the SAMPL6 SAMPLing challenge *Journal of the American Chemical Society* 32 1001-1012 https://link.springer.com/article/10.1007/s10822-018-0149-3

- **Lotz SD** and Dickson A* (2018) Unbiased Molecular Dynamics of 11 min Timescale Drug Unbinding Reveals Transition State Stabilizing Interactions. *Journal of the American Chemical Society* 140 (2): 618-628 https://doi.org/10.1021/jacs.7b08572

- Dickson A*, **Lotz SD** (2017) Multiple Unbinding Pathways and Ligand-Induced Destabilization Revealed by WExplore and Conformation Space Networks. *Biophysical Journal* 112: 620-629. https://doi.org/10.1016/j.bpj.2017.01.006

- Oakley C*, Savage L, **Lotz SD**, Larson R, Thomashaw M, Kramer DM, Schemske DW (2017) Genetic basis of photosynthetic responses to cold in two locally adapted populations of Arabidopsis thaliana *Journal of Experimental Botany* erx437 https://doi.org/10.1093/jxb/erx437

- Dickson A*, **Lotz SD** (2016) Ligand Release Pathways Obtained with WExplore: Residence Times and Mechanisms. *Journal of Physical Chemistry B* 120: 5377-5385. https://doi.org/10.1021/acs.jpcb.6b04012

Scientific Software Projects

Github Profile: `https://github.com/salotz`
PhD Group Github Profile: `https://github.com/ADicksonLab`

Only project for which source code is open and available are listed here. See my work experience sections for more information on professional closed source projects.

Wepy (Python) **Spring 2017 – present**
`https://github.com/ADicksonLab/wepy`

`Wepy` (also implemented in `Python`) is a software package aimed at enabling weighted ensemble (WE) simulations very efficiently and with a highly modular architecture to allow for quick prototyping. It has been used to develop a number of published resampling algorithms (including WExplore and REVO) and simulation protocols used both in academia and in industrial drug-discovery pipelines. I have been contacted by a number of drug-discovery companies regarding its potential usage.

It implements a simple simulation manager which manages the running of parallel simulations (walkers) and the resampling of those walkers at a cycle frequency, as well as allowing for the addition of boundary conditions and any kind of data reporter. Currently, OpenMM is used for running molecular dynamics, and two cutting edge resamplers are implemented as defaults (including WExplore). Like most aspects of `Wepy` the creation of resamplers is modular and uncoupled from the rest of the software, allowing researchers to focus solely on the implementation of their resampling method and not the actual running of simulations or saving of data. `Wepy` also provides an optional framework for writing resamplers which makes the development of system specific distance metrics (among other parts of resamplers) simple. Additionally, an advanced HDF5 format for saving trajectory data was implemented. This format improves upon other trajectory data formats by allowing random access (as opposed to sequential access) which can improve run times of certain analyses by several orders of magnitude. Additionally, because weighted ensemble clones and merges trajectories, the stored data is usually not sequential in simulation time, which leads to a complicated data management problem which hamstrings analytic techniques using sliding windows, such as the creation of Markov State Models with long lag times. Because the `Wepy` HDF5 format keeps track of the clones and merges in an detailed way and directly couples that data to the trajectories this makes such analyses much easier (and faster) to implement.

Thesis Pipelines **Spring 2020 – Spring 2021**
`https://github.com/salotz/thesis-workflows`

This is a repository showcasing the software pipelines I wrote to achieve the last portion of my thesis work. It is missing a number of data assets which are not able to be stored in Github.

Geomm (Python) **Spring 2018 – present**
`https://github.com/ADicksonLab/geomm`

`Geomm` intends to be a small library of geometric functions that are commonly used in computational chemistry and biology. While there are many libraries that work explicitly on molecular dynamics trajectory data, however the design of `Geomm` intends to be decoupled from application specific details. Particularly, there is no dependence on topologies. This is because there are potentially many ways to represent topologies and querying them is better left to libraries that specialize in topologies. Instead `Geomm` works only on the coordinates of particles (as `Numpy` arrays) and selections on the indices of those particles. So while the routines in `Geomm` are integrated to be useful for macromolecular systems, they are not beholden to them. The benefits of such an architecture are that when used as a library in other projects (such as `Wepy`) it cuts down overhead on both computational efficiency and lines of code that other libraries incur by forcing the use of topologies for computing geometric properties. Additionally, exotic or nonstandard topologies, for say Lennard-Jones particle systems or course-grained systems, do not need special accomodations or schemas. A more overarching goal of `Geomm` is to provide reference implementations of the myriad computations used in computational chemistry/biology, that will hopefully clarify the meanings of terms and reduce the number of mistakes made in computations.

Mastic (Python) **Spring 2017 – 2019**
`https://github.com/ADicksonLab/mastic`

`Mastic` is an extensible framework for profiling intermolecular interactions. It is extensible because any sort of interaction can be defined and profiled using the same framework. This was accomplished using object-oriented programming in the `Python` language. `Mastic` fills a gap in the computational chemistry/biology field by providing this functionality as it's own stand-alone library which may be used in any manner. It can be parallelized and used on large MD trajectory datasets or it can be used to compute order-parameters for enhanced sampling simulations, (for instance in `Wepy`). It accomplishes it's task by outsourcing feature recognition of molecular structures (e.g. identifying hydrogen bond donors and acceptors) to a more domain specific software package. Currently, `rdkit` is used for feature recognition, but as knowledge and interest change concerning chemical functional groups new feature recognition software can easily be plugged into `Mastic`. `Mastic` also allows for more efficient profiling by allowing the user to only profile certain kinds of interactions or interactions of certain subsets of atoms.

Protein Folding Ant Colony Optimization (Java) **Fall 2013 – Summer 2014**
`https://github.com/salotz/protein-folding-ant-colony-opt-java`

An independent study project written in `Java` that performs Ant Colony optimization to fold and draw pictures of proteins on a 2D square lattice. The algorithm was from Alena Shmygelska's dissertation "Novel Heuristic Search Methods for Protein Folding and Identification of Folding Pathways" Sept 2006 University of British Columbia. I learned the basics of programming scientific applications as well as how to write structured programs in a statically-typed object oriented programming language.

Agent-based model of vehicle traffic (Java) **February 2014**
`https://github.com/salotz/MCM-2014-traffic-models`

This was a project written in Java for the 2014 International Mathematics Contest in Modeling.

It implements an agent-based model of vehicle traffic in order to predict the effect of various traffic rules on overall traffic flow.

I had no previous experience in Java programming and this work was completed in a three day period. As such the code quality is poor but showcases my ability to learn and deliver quickly when needed.

Misc. Software Projects

Github Profile: `https://github.com/salotz`

Note that most of these are not actively maintained or have a user base. Some I still use regularly for my own purposes and others are kind of experiments or solved particular problems I had at various points in time.

snailpacks **2021 – present**
`https://github.com/salotz/snailpacks`

This is a Spack (`https://spack.io`) repo for packages that I use in some of my hobby projects. A Spack repo is a collection of package recipes which can be used to build software in a variety of build configurations. The focus is mostly on C, C++, and Rust projects related to multimedia applications.

refugue (Python) **2020**
`https://github.com/salotz/refugue`

This is a `Python` utility meant to coordinate the movement of sets of files around to different machines and targets in a customizable and configurable manner. It provides an easily installable CLI to accomplish this.

It is roughly comparable to a decentralized Dropbox and works as a wrapper to rsync. It works by setting up a single configuration file defining all of the target locations, named sets of data, and sync policies between them.

I used it primarily to transfer data over an external hard drive when working in two different locations for which transfer over the internet was not feasible due to data volume.

bimhaw (Python) **2020**
`https://github.com/salotz/bimhaw`

This is a `Python` utility which acts as a frontend abstraction for shell startup files (e.g. `.profile`, `.bashrc`, etc.) and a shell configuration profile manager. It provides an easily installable CLI with few dependencies to accomplish this so that it is easy to use in a minimal environment (e.g. setting up a freshly installed operating system).

The `bimhaw` configuration uses a modular configuration organization so that your shell configurations can be managed by combining single modules and a configuration file. This allows you to turn on or off shell configuration without needing to (un)comment lines in your shell loading profiles.

I have uses this daily since I created it and has substantially decreased the time needed to set up new machines, debug problematic shell profiles, and has allowed for experimentation with new technologies without cluttering my profile configurations. It has also improved shell load times and experience by ensuring that the correct modules get activated at the right shell load time and environment (which is pretty universally a point of confusion even for advanced users).

I recommend this if you have complicated shell profile configurations that you are looking to tame without losing functionality.

bibby (Python) **2020**
`https://github.com/salotz/bibby`

This is a `Python` utility and CLI that I used when writing academic papers and my thesis that automates the download and conversion of bibliographic references from the `https://www.bibsonomy.org/` web application.

In my workflow I utilized Bibsonomy to bookmark, tag, and catalogue all the scientific references that I read and cited in my publications. This greatly increased my productivity, however it was still tedious to extract these to a single LaTeX BibTex file for use in my documents. I wrote this small utility to address this problem.

I have used this in all my publications since then.

Inkscape Pages (Python) **2020**
`https://github.com/salotz/inkscape_pages`

This is a `Python` utility and CLI used to create PDF slideshows from an Inkscape (`https://inkscape.org/`) document.

Inkscape is a great free tool for doing vector drawings and documents (similar to Adobe Illustrator). I use it for making custom figures, posters, and slideshows. One shortcoming is that Inkscape does (or did not) not directly support creating multi-page documents like slideshows.

I created this simple tool to take an Inkscape document and create pdf slideshows using each layer as a different slide. It could be used for any multi-page kind of document.

I have used this tool in numerous slideshows as a replacement for Microsoft PowerPoint. I continue to use it to this day whenever not constrained by e.g. company standards.

Python Serialize Test (Python) **2019**
`https://github.com/salotz/python_serialize_test`

This is a Python `Python` CLI utility which you can use to test the performance of a number of different pickle serialization backends on a test pickle file.

I used this to choose a pickle serialization backend for work in my thesis.

pytest-datadir-extras (Python) **2019**
`https://github.com/salotz/pytest-datadir`

This was an experiment in adding some features to the `pytest-datadir` pytest plugin. The original package does not support arbitrary scopes for different kinds of data directory fixtures and this package adds this ability.

The upstream maintainers were not interested in these changes when I made a PR for this.

Pymatuning (Python) **2018**
`https://github.com/salotz/pymatuning`

This is a small `Python` utility that can be used to generate an Emacs Org-mode checklist plaintext document from a Python module. I used this to make a checklist to audit each class and function in my project for docstrings. Nowadays I would use an existing docstring linting tool; but it was a fun exercise.

panda-peek (Python) **2017**
`https://github.com/salotz/panda-peek`

This is a small `Python` CLI utility for pretty printing CSV files using pandas.

Exotic Software Projects

I maintain an interest in a wide variety of other software technologies. In addition to the more mainstream tools I use I like to investigate, learn, and experiment with more exotic options. This has been instrumental in developing a good intuition for the strengths and limitations of mainstream technologies. It is also enlightening to keep an open mind and eye towards potentially ground-breaking technologies that are more than 5 years out from being viable or mainstream as these represent opportunities for growth and innovation.

I don't expect this section of to be professional importance but maintain it to showcase my ability to be innovative and to learn things quickly.

The Scopes Programming Language **2021** – present

Here are a few projects I have worked on related to the virtually unknown but extremely powerful programming language `Scopes` (`http://scopes.rocks/`).

`Scopes` visually looks and feels like Python, but semantically is more similar to C++ and Rust in its ability for high-performance low-level programming and borrow-checking safety features. It also shares a lot of similarities with Lisp and Scheme for its syntactic flexibility and macro system. It sports a pluggable, online, multi-stage compiler (via LLVM) to allow for dynamic execution and hot reloading. It supports ABI compatibility with C and has trivial interfacing with C libraries.

Scopes Newbs `https://github.com/salotz/scopes-newbs`

I have produced an incomplete, but extensive guide for newcomers to the language oriented towards those with experience in dynamic scripting languages like `Python`.

It is produced in a form using Emacs org-mode so that all code blocks are executable similar to a Jupyter notebook.

Scopes Chipmunk2D wrapper `https://github.com/salotz/scopes-chipmunk2d`

This is a light wrapper around the Chipmunk2D 2D physics library used in games.

Scopes Raylib wrapper `https://github.com/salotz/scopes-raylib`

This is a light wrapper around the Raylib game programming framework.

Scopes Demos `https://github.com/salotz/scopes-demos`

Various demos oriented towards graphical game-like applications. Includes a physics demo.

Additional Open Source Contributions

- Contributed bugfixes to the Arcan project (https://github.com/letoram/arcan)
- Contributed to the NetworkX documentation (https://github.com/networkx/networkx)
- Bug reports for OpenMM (https://github.com/pandegroup/openmm)
- Bug reports for MDTraj (https://github.com/mdtraj/mdtraj)
- Q&A on StackExchange (https://stackexchange.com/users/4314396/salotz)

Conference Presentations

American Chemical Society
San Francisco, California **Spring 2017**
Poster: *Unbiased Molecular Dynamics of 11 min Timescale Drug Unbinding Reveals Transition State Stabilizing Residues*

13th Midwest Conference on Protein Folding, Assemblies, and Molecular Motions
University of Notre Dame, South Bend, Indiana **Spring 2016**
Poster: *Conformational Network and Residence Time Estimation of Trypsin-Benzamidine Unbinding Pathways*

16th International Congress on Photosynthesis Research
St. Louis, Missouri **Summer 2013**
Mid-SURE Undergraduate Research Symposium
Michigan State University **Summer 2013**
Poster: *The Role of an NAD(P)H Dehydrogenase Complex Associated Protein in Regulatin Cyclic Electron Flow Under Light-Limiting Conditions*

Commonwealth of Pennsylvania University Biologists
Clarion University, Clarion, Pennsylvania **Fall 2013**
Poster: *Drastic Systemic Protein Increase in Wounded Plant Stems*

Awards

- 2014, Early Start Scholars Fellowship, Michigan State University
- 2012, Puntureri Memorial Science Scholarship, Slippery Rock University
- 2012 & 2013, John P. Phillips Memorial Scholarship, Slippery Rock University
- 2012, Leucojeum Award in Botany, Slippery Rock University
- 2012, Sophomore Biology Honors Scholarship, Slippery Rock University
- 2011, Achievement in Biology Scholarship, Slipery Rock University

Community Outreach

MSU Grandparents' University **2016-2017**
How Do Molecules Find Their Perfect Fit?
Assisted my Advisor Alex Dickson in activities with children and their grandparents using 3D printed models of proteins that helped them to conceptualize and concretize the microscopic principles of protein interactions which underpin physiology and disease.

Other Experiences

Teaching Assistant for Biochemistry and Molecular Biology Laboratory Course **Spring 2016**

As a a TA for the 400 level biochemistry laboratory class I guided students through a variety of experiments.

International Mathematics Contest in Modeling **February 2014**
As part of a three person team wrote an agent-based model of vehicle traffic in Java and wrote report of findings.

Our model can be found here: `https://github.com/salotz/MCM-2014-traffic-models`

- Scored in the top 15% out of 6755 teams.

Relevant Coursework

Science:

- Genetics
- Organic Chemistry I, II
- Botany
- Zoology
- Biochemistry I, II
- Forst Ecology
- Field Botany
- Ecology
- Plant Physiology
- Molecular Biology
- Cell Biology
- Microbiology
- Biometry/Biostatistics (SAS)
- Physical Chemistry I
- Molecular Biology (Graduate)
- Methods of Macromolecular Synthesis and Analysis (Graduate)
- Protein Design, Structure, and Mechanisms (Graduate)
- Plant Genomics (Graduate)
- Statistical Mechanics I (Graduate)
- Quantum Mechanics I (Graduate, audit)

Mathematics:

- Calculus I, II, and III
- Linear Algebra and Differential Equatoins
- Differential Equations I
- Mathematical Statistics
- Stochastic Processes in Finance (Graduate, audit)

Computer Science:

- Programming I, II (C++)
- Database Systems Management (SQL)
- Algorithms and Data Structures (Java)
- Machine Learning (Java)