simcodes-isu.github.io

Automating the Fragmentation of Proteins

By **Devarsh Shroff**

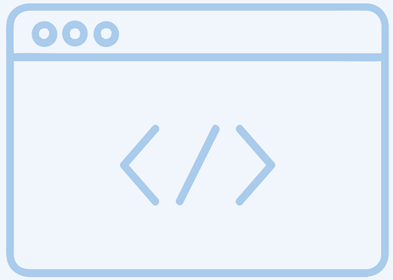
Affiliation: SIMCODES, NSF REU 2025

Project Overview

Problem statement : Accurately predicting enzymatic reactions requires reliable energy calculations for proteins, ligands, and their environment. Traditional quantum chemical methods are too slow and computationally intensive for large systems. This creates a need for faster approaches that balances speed with accuracy.

- We scoped down to peptide-level fragmentation due to computational and database constraints.
- As molecular structures resemble graph like structure and properties, we decided to use graph clustering algorithms/models to generate fragments. To make chemical sense we also capped the fragments with hydrogens.
- Once we were able to generate consistent results we shifted our focus to make the accuracy of the model higher and in the given threshold.

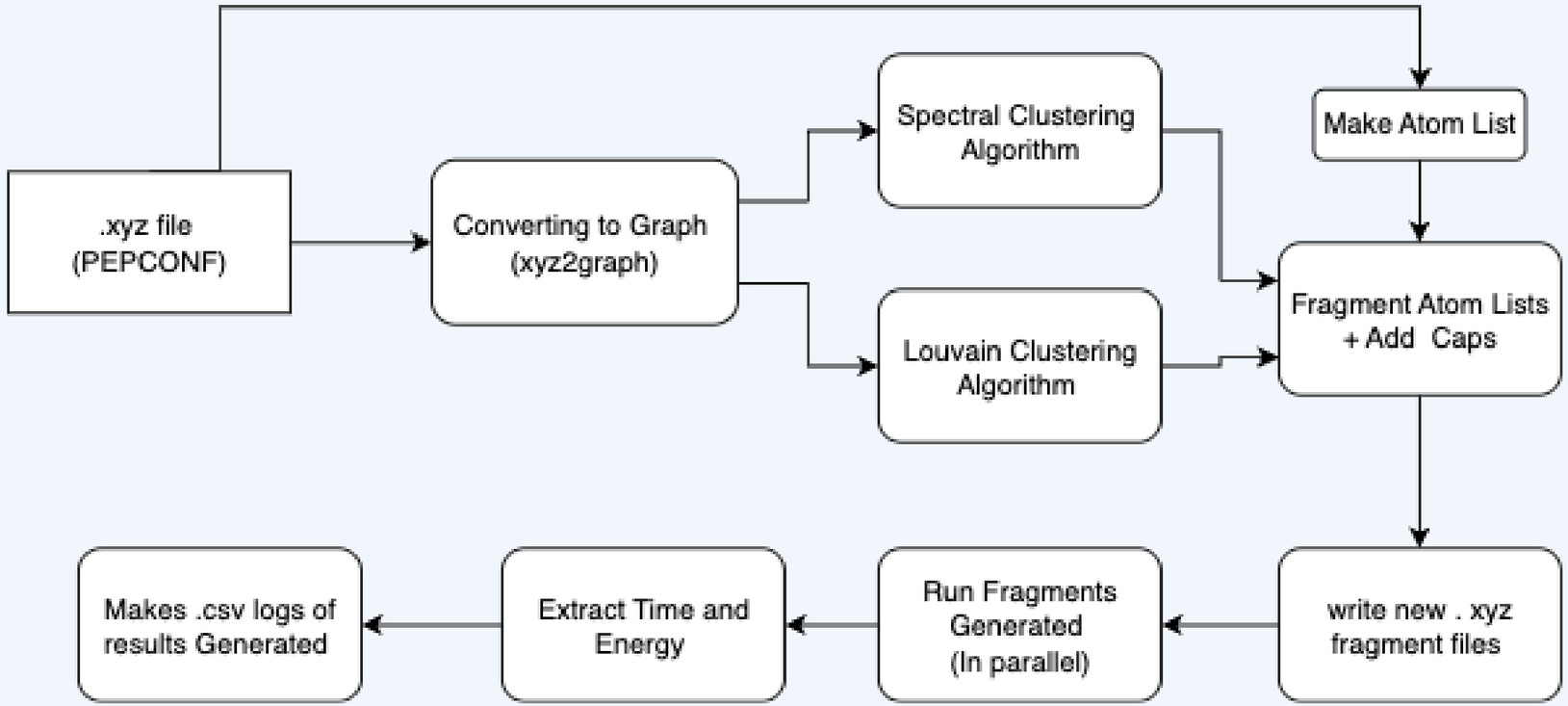
My Role



As a research intern in the **SIMCODES program**, I contributed to improving and automating fragmentation-based energy calculation workflows. My work centered on identifying optimal graph-based clustering algorithms/models (Louvain, Spectral) to fragment biomolecules efficiently and consistently, enabling accurate **xTB** energy predictions across a range of peptide sizes.

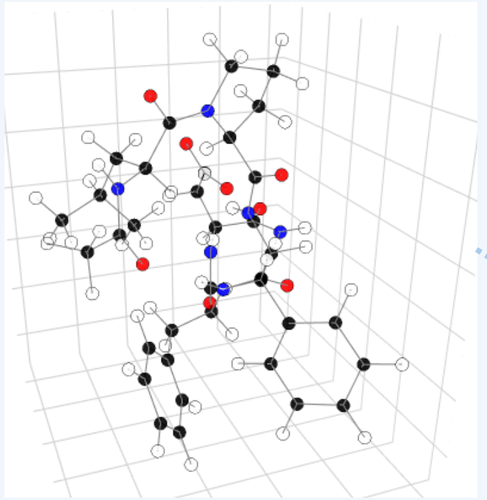
I developed and validated a full computational pipeline in Python, integrating tools such as **xTB**, **NetworkX**, and **xyz2graph**. The pipeline converted .xyz files into graphs, ran clustering algorithms to define fragment boundaries, and automated steps such as hydrogen capping, energy correction, file I/O, and job execution through xTB. Also running jobs for hyperparameter tuning with different fragmentation models and visualized results to evaluate the trade-off between hyper-parameters and energy error.

Pipeline Developed



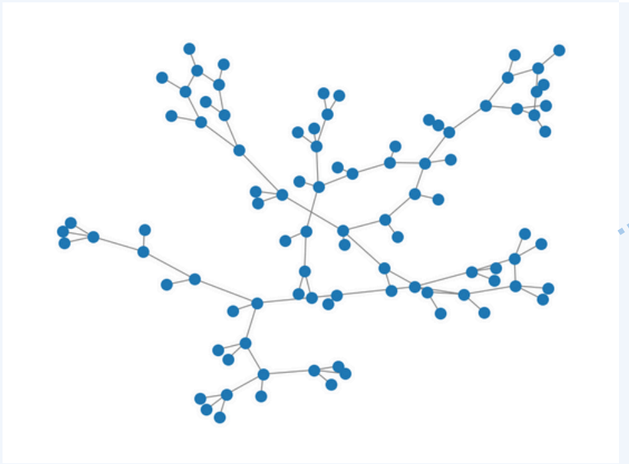
Code implementation: clustering_V1_3.py

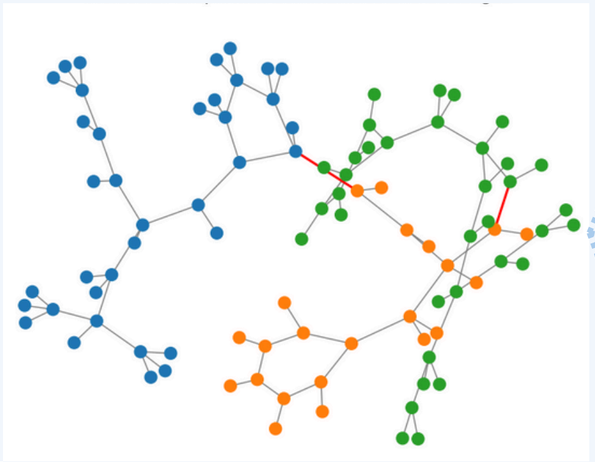
simcodes-isu.github.io



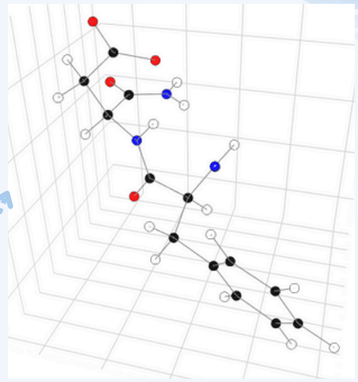
Molecular graph of .xyz file

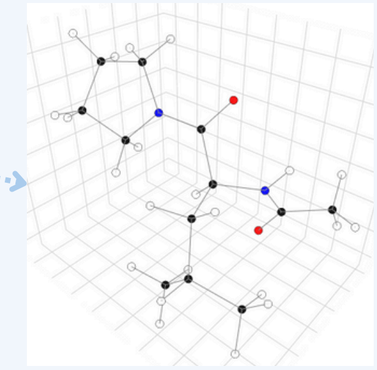
networkX graph of .xyz file

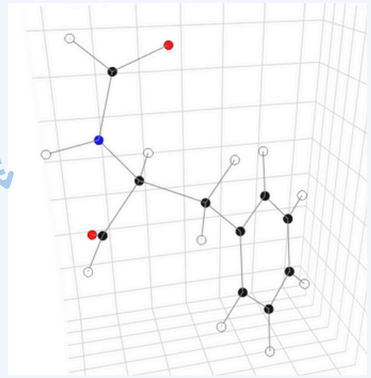




Clustering Visualisation







Molcular Graph of fragments

Affiliation: SIMCODES, NSF REU 2025

simcodes-isu.github.io

Progress Made

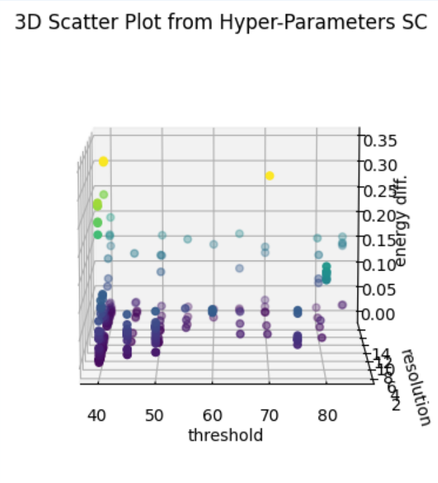
01 **H-capping technique :** initially hand capped and fragment peptides to get a proof of concept that H capping will work and provide similar energies using xTB

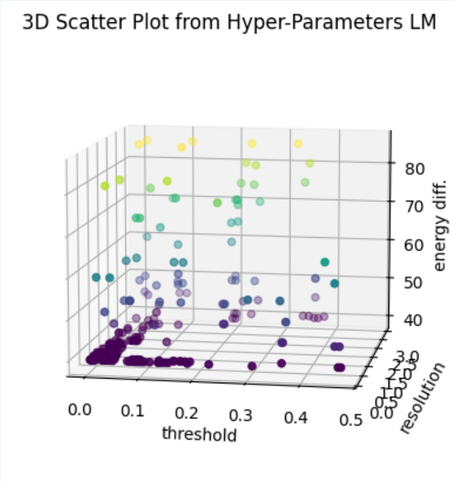
02 **Built scripts for fragmentation :** Implemented Spectral Clustering and Louvain Method and made pipeline around it to use the output to make fragment files

03 **Integrated xTB:** Using Sub-process can xTB calculation on original peptides and fragments files and log the data in .csv files.

04 **Added Parallelization :** we parallelised xTB jobs, drastically reducing the time taken to run peptides.

05 **Hyper-parameter tuning :** hyperparameter tuning module, to check if the energy difference between ground truth and fragments total energy, logging the optimal parameters. Ran it for 235 peptides and for 2 clustering algorithms and compared their outputs.

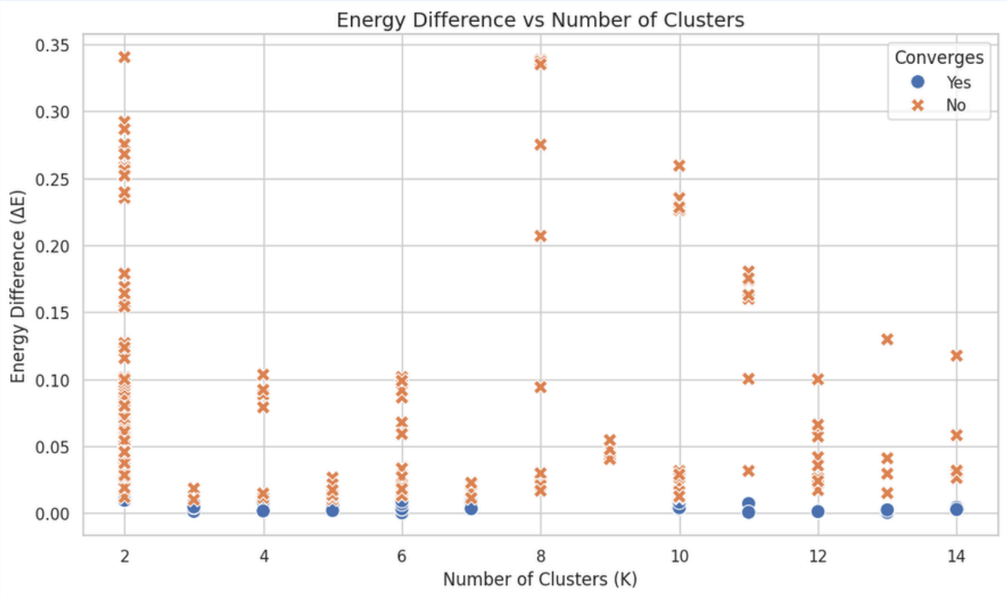




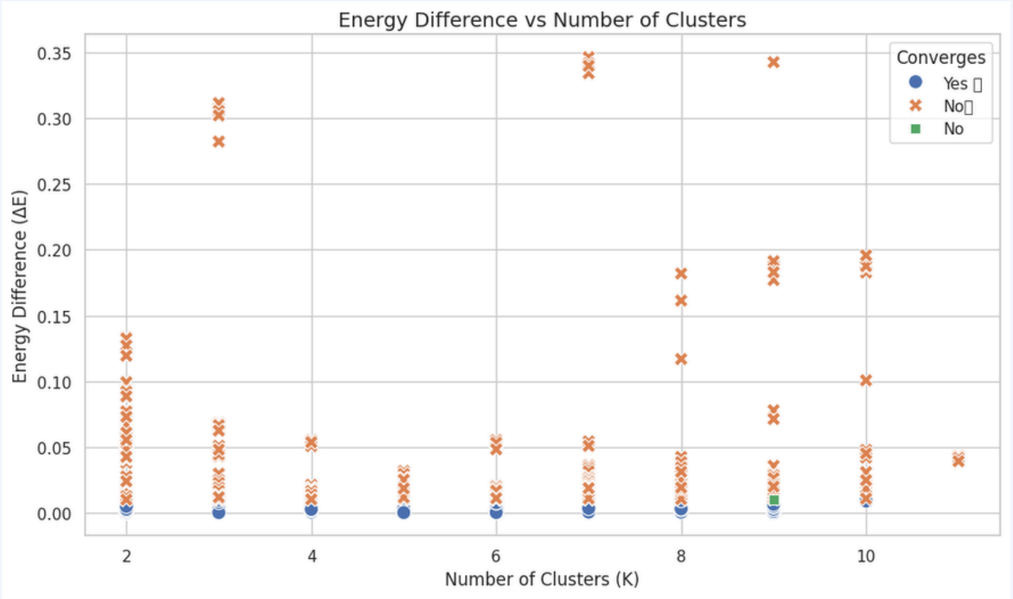
06

Spectral Clustering tuning : corrected the redundancies found in the spectral Clustering and made it more efficient and ran for 2000**+ peptides**. Graphing the hyper-parameter and how it affected the energy difference.

Current Stats: 973 out of 1494 where in optimal energy difference threshold that's about 65% and the times is reduced by about 95% on average after fragmenting. The plot below shows how energy difference changes with number of clusters for spectral clustering technique :



Bioactive : Clusters 2–7 and 12–14 have lower energy difference, but no single best parameter.



Dipeptides : A concave-up trend, suggesting the sweet spot is around 4–6 clusters.

Affiliation: SIMCODES, NSF REU 2025

Learning Highlights

- Gained hands on experience with xTB and xyz2graph, Set up environments using Conda/Miniforge, BASH scripting to streamline workflow.
- Explored Graph Neural Networks (GNNs) through Stanford and FreeCodeCamp resources.
- Learned to access Nova using Slurm job submission scripts and joblib for parallelizing energy calculations.
- Built data visualization, using Matplotlib, Seaborn, and 3D plotting to analyze energy differences trends.
- Developed a understanding of hyperparameter tuning, learned to balance computational efficiency with model fidelity when searching for optimal fragmentation strategies.
- Learned to use Python's subprocess library to automate system-level scripting (e.g., running xTB) .
- Through repeated experimentation and data analysis, I learned to identify systematic errors introduced by incorrect fragmentation and how computational cost scales with system size. Highlighting the critical balance required between accuracy and automation, especially when preparing workflows for integration of ML driven models.

simcodes-isu.github.io

Challenging Aspects

- **Handling edge cases in fragmentation:** Designing reliable capping for fragmented molecules proved especially difficult when bond cuts occurred at complex junctions.
- **Balancing chemical accuracy with abstraction:** Simplifying molecular systems for computational feasibility while maintaining chemically meaningful fragmentation.
- **Combinatorial complexity of fragmentation:** Exhaustively exploring all possible fragmentations is computationally infeasible, so we end up ranking the possible fragmenationations.
- **Memory and runtime constraints:** Some clustering configurations led to runtime issues and requiring optimization of the algorithm .
- **Working with large and opaque systems:** As peptide sizes scaled up, it became increasingly difficult to visually or manually validate fragmentations and energy outputs.

Affiliation: SIMCODES, NSF REU 2025

What's Next

- **Make Clustering Chemistry Aware:** incorporating valency checks and bond weights to make some cuts less likely and more chemistry-aware.
- **Upgrade energy calculations:** Use more accurate quantum chemical methods to assess how outputs generalize.
- **Introduce testing :** Implement unit tests using pytest to ensure reliability and correctness of the automated pipeline.
- **Apply ML regression model:** Train regression models to predict energy differences and runtimes based on molecular graph features.
- **Graph-informed tuning:** Refine clustering by analyzing molecular graph properties to guide fragmentation and improve prediction quality.
- **Scale to full proteins:** Extend current clustering pipeline on full-length proteins to evaluate performance and generalizing beyond peptides.

Thanks to :

- SIMCODES program and ISU
- NSF (award no. 2348724)
- Mentors: Dr. Qi LI, Dr. Ryan Richard, Dr. Theresa Windus and Dr. Myra Cohen
- All code and data hosted on GitHub for reproducibility on Devarsh_repo