Fast and Efficient Particle Trajectory Analysis with the freud Library
Tommy Waltmann¹, Bradley Dice¹, Vyasa Ramasubramani², Joshua Anderson², Sharon C. Glotzer¹²³

¹Department of Physics, University of Michigan
²Department of Chemical Engineering, University of Michigan
³Department of Materials Science and Engineering, University of Michigan
⁴Biointerfaces Institute, University of Michigan

Introduction

Modern day computational resources have allowed the expansion of research utilizing computer simulation to study larger and larger system sizes. Systems with millions of particles can now be simulated for hundreds of thousands of timesteps in a matter of days due to large amounts of effort into developing software which can run these simulations as fast as possible. However, without tools fit to perform analysis on these large systems, the research bottleneck shifts from the simulation side to the analysis side of the workflow. To this end, we present freud: a powerful and efficient particle trajectory analysis library which exposes an easy-to-use python API.

Standard Analysis Methods

The freud library computes many common analysis methods, such as radial distribution functions (RDF) and mean-squared displacements (MSD).

Order Parameters

The freud library computes a wide variety of topological order parameters such as the Steinhardt order parameters and their variants, nematic, translational, solid/liquid, and others.

Neighbor Finding

Whether explicit or implicit, many applications of the freud library require neighbor finding, which can be accelerated by using specialized data structures for querying points.

Scattering Analysis

In more recent freud developments, many new scattering analysis features, such as diffraction patterns and static structure factors, have been added to the codebase.

Other Analysis Methods

The freud library also contains other analysis methods rarely or not seen elsewhere such as PMFTs, environment matching, and interface detection.

Machine Learning Features

In more sophisticated applications, order parameters from the freud library have been used as features for machine learning applications.

Performance

The freud library was built to compute results quickly. The figure to the right shows a system of 1,000,000 particles (blue) randomly distributed in a 2D box at density 0.01. Freud computed all the neighbors (red) for this system in 3.18 seconds using 12 threads.

Python Ecosystem

The freud library’s NumPy array interfaces allow it to integrate tightly within the scientific python ecosystem.

Code: github.com/glotzerlab/freud
Documentation: freud.readthedocs.io