MD Simulations with OpenMM

Toni Giorgino
National Research Council of Italy
toni.giorgino@cnr.it
www.giorginolab.it

@giorginolab

Thesis projects available

https://github.com/giorginolab/
OpenMM-Tutorial-UniPD-2023

University of Padova c/o Prof. Fuxreiter
Mar 16, 2023

This class

- Molecular dynamics is a powerful tool for studying molecular systems
- OpenMM is a software library that allows for efficient and customizable MD simulations
- It's exemplary of a modern well-maintained open-source library:
  - CI infrastructure, developed on GitHub
  - C++ w/ Python bindings
- We'll use the latter, testing *live* on Google Colab.

Molecular Dynamics

What is MD?

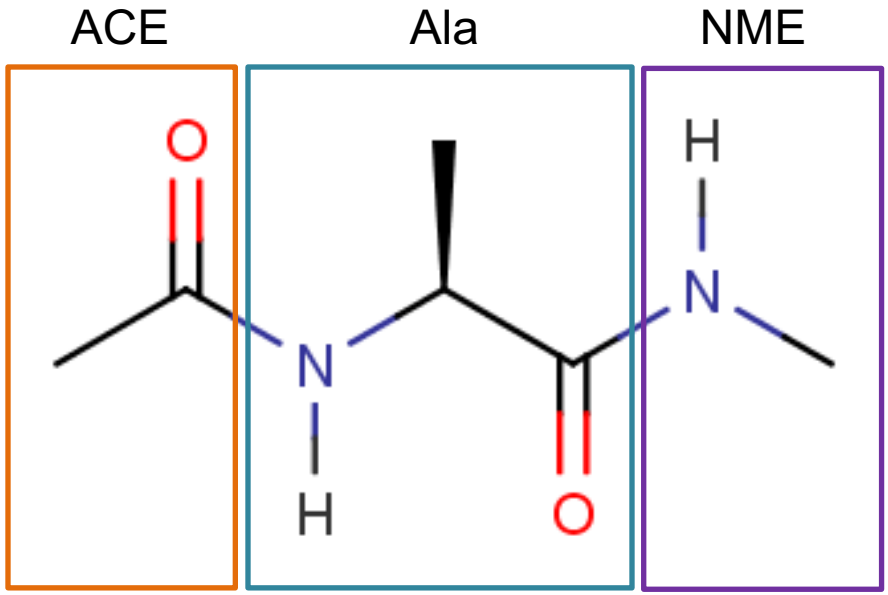
- Attempt the most detailed description of a system which is
  1. atomistic
  2. classical
- Model the internal *forces*…
- …in order to *integrate* the motion
- Hope in convergent *sampling*

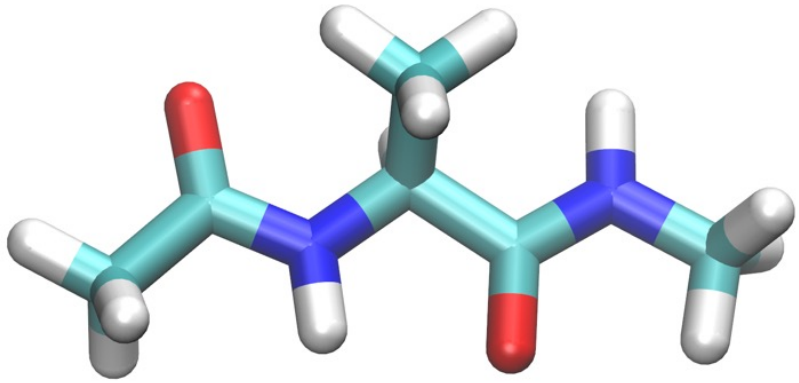
$$\vec{F}_i(\mathbf{x}) = m_i \ddot{\mathbf{x}}_i$$

Assumptions

- In this tutorial we shall deal with **unbiased** sampling approaches with **explicit** solvent, i.e.
  - no added forces except the "physical" ones in your system;
  - all of the system (including water molecules) have atomic resolution.
- Also, current classical MD does not address, by design, the following:
  - Chemical reactions, e.g. catalysis, phosphorylation, ubiquitination etc.
  - Protonation changes
- Finally, small molecules pose distinct challenges and need a separate, expensive **parameterization** step.

Alanine “dipeptide”

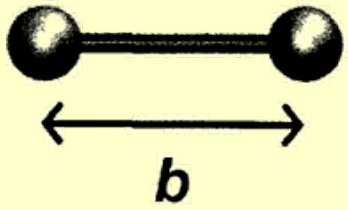


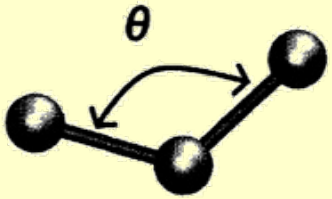


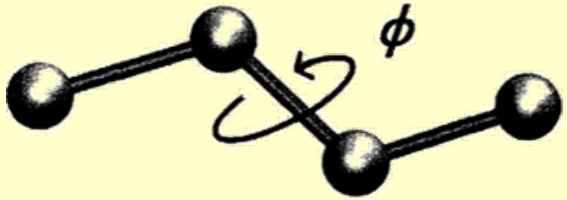
Bonded energy terms

+ Electrostatics

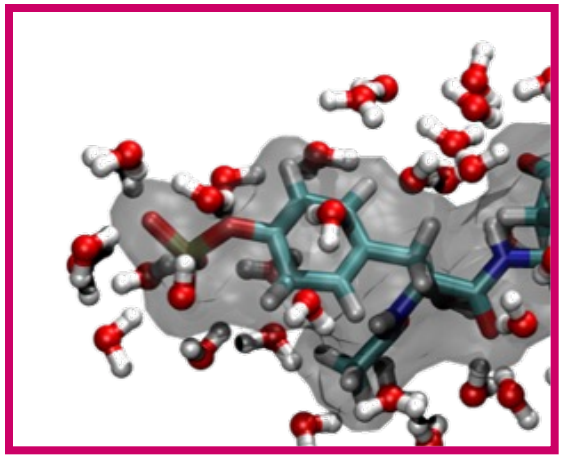
+ VdW



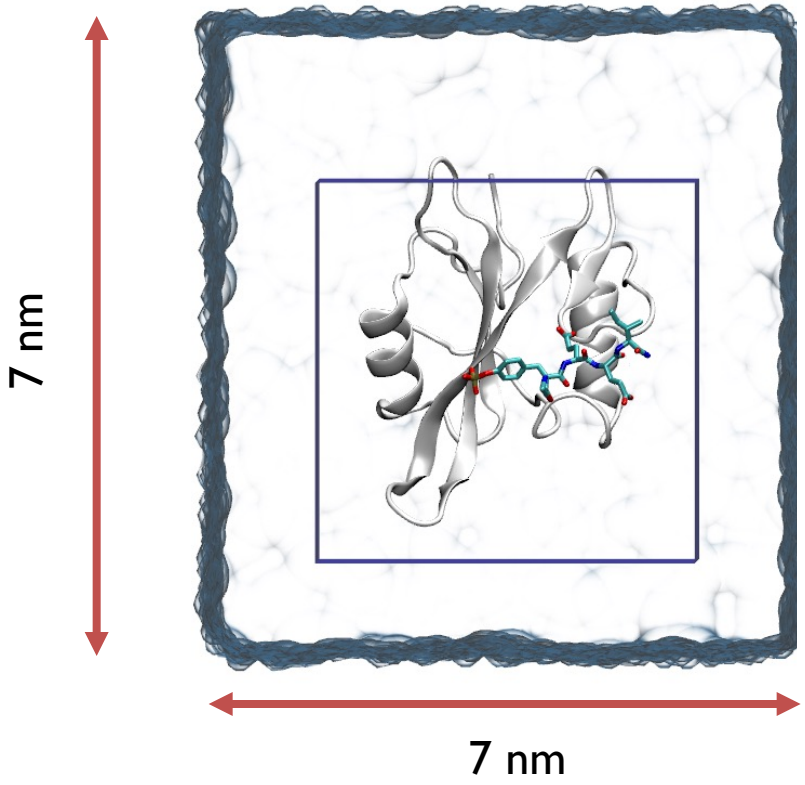


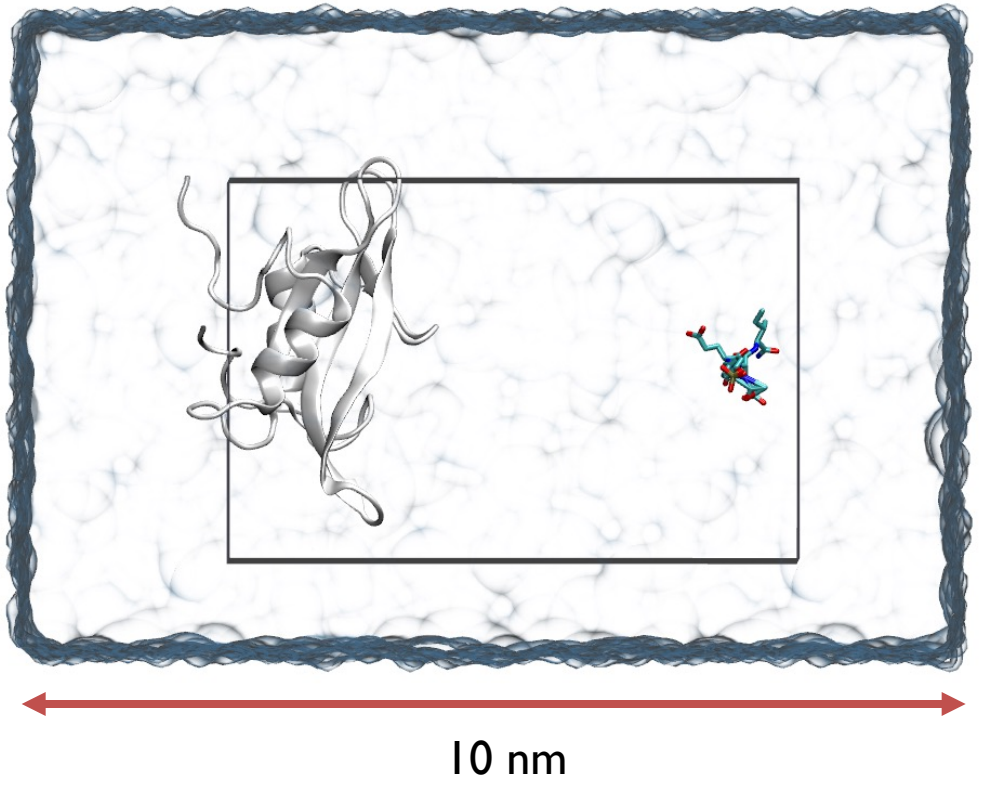


The forcefield is a database of interatomic parameters



- **Explicit solvation**
- → O(10^5) atoms
- **Unbiased dynamics**
- Update every 10^{-15} s (1 fs)





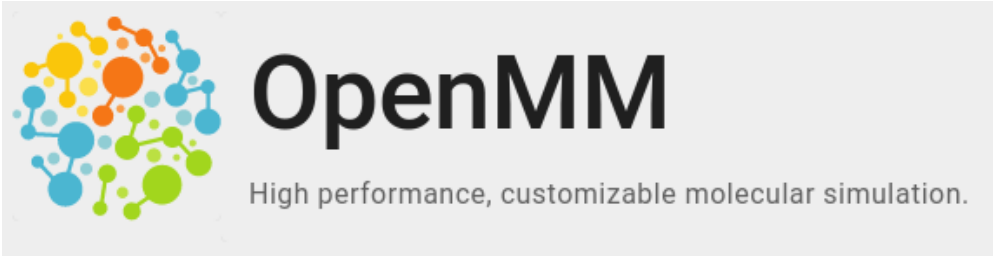
MD is entirely about timescales

- Your ability to obtain quantitative results is severely limited by the sampling ability you have. You will only be able to reach phenomena occurring on the sampled timescales, or shorter.
  - Sidechain rearrangements, diffusion-limited processes: usually possible *
  - Local flexibility: usually possible *
  - Membrane environments: ok-ish
  - Binding: hard but not impossible
  - Folding: very hard but not impossible
    - [*] Unless there are significant barriers.

Patience and limits

- The following factors affect the running speed (usually expressed in ns per simulation day, ns/day)
  - System size. Reasonable is 100 AA ~ 30,000 atoms.
  - Computer speed. Forget laptops.
  - Definitely use GPUs.
  - Software.

OpenMM



.org

- OpenMM is a molecular dynamics simulation toolkit that allows for high-performance simulations of biomolecules.
- Allows for simulation of a variety of molecular systems, including proteins, nucleic acids, and small molecules
- OpenMM supports a wide range of force fields and integrators and can run on CPUs and GPUs.
- Open source, written in C++ with Python and other language bindings available

Basic Workflow (object-oriented)

1. Download, complete and edit the structure:
   - **Topology** (i.e. the identity of atoms, bonds, etc)
   - **Positions** (i.e. the starting coordinates)
2. Create the **system** object.
3. Create the **integrator** object.
4. Create and add custom **forces** to system if needed.
5. Define the **simulation** object.
6. Set the initial positions and velocities.
7. Minimize.
8. Run the simulation.
9. (Analyze the results.)

Integrators

- …are algorithms that solve the equations of motion for a system
- OpenMM includes several integrators, e.g. Langevin dynamics, Verlet integrator, and Monte Carlo barostat
- Different integrators are appropriate for different types of simulations and conditions (e.g.: NPT vs NVT)

Simulating a system

- Once a system has been defined and the force field and integrator selected, it can be simulated
- The simulation (run) involves running a series of steps, where each step involves calculating the forces on each atom, integrating the equations of motion, and updating the system's coordinates
- After the simulation, data analysis can be performed to obtain information about the system's behavior and properties

Let’s pick a test system

6H1F: Gelsolin G2+nanobody

Structure Summary | 3D View | Annotations | Experiment | Sequence | Genome | Versions

Biological Assembly 1

3D View: Structure | 1D-3D View | Electron Density | Validation Report | Ligand Interaction

Global Symmetry: Asymmetric - C1
Global Stoichiometry: Hetero 2-mer - A1B1

Find Similar Assemblies

Biological assembly 1 assigned by authors and generated by PISA (software)

Biological Assembly Evidence: gel filtration

Macromolecule Content

- Total Structure Weight: 28.49 kDa
- Atom Count: 1,896
- Modelled Residue Count: 229
- Deposited Residue Count: 259
- Unique protein chains: 2

Display Files | Download Files

6H1F

Structure of the nanobody-stabilized gelsolin D187N variant (second domain)

PDB DOI: 10.2210/pdb6H1F/pdb

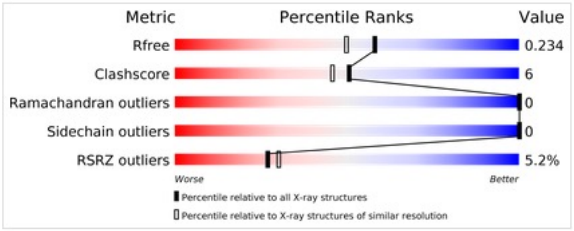
Classification: STRUCTURAL PROTEIN
Organism(s): Lama glama, Homo sapiens
Expression System: Escherichia coli
Mutation(s): Yes

Deposited: 2018-07-11 **Released:** 2019-01-23
Deposition Author(s): Hassan, A., Milani, M., Mastrangelo, E., de Rosa, M.
Funding Organization(s): Amyloidosis Foundation

Experimental Data Snapshot

Method: X-RAY DIFFRACTION
Resolution: 1.90 Å
R-Value Free: 0.233
R-Value Work: 0.199
R-Value Observed: 0.202

wwPDB Validation | 3D Report | Full Report



This is version 1.0 of the entry. See complete history.

Literature | Download Primary Citation

Nanobody interaction unveils structure, dynamics and proteotoxicity of the Finnish-type amyloidogenic gelsolin variant.

Giorgino, T., Mattioni, D., Hassan, A., Milani, M., Mastrangelo, E., Barbiroli, A., Verhelle, A., Gettemans, J., Barzago, M.M., Diomede, L., de Rosa, M.
(2019) Biochim Biophys Acta Mol Basis Dis **1865**: 648-660

PubMed: 30625383 Search on PubMed
DOI: 10.1016/j.bbadis.2019.01.010
Primary Citation of Related Structures:

3D Protein Feature View: 6H1F

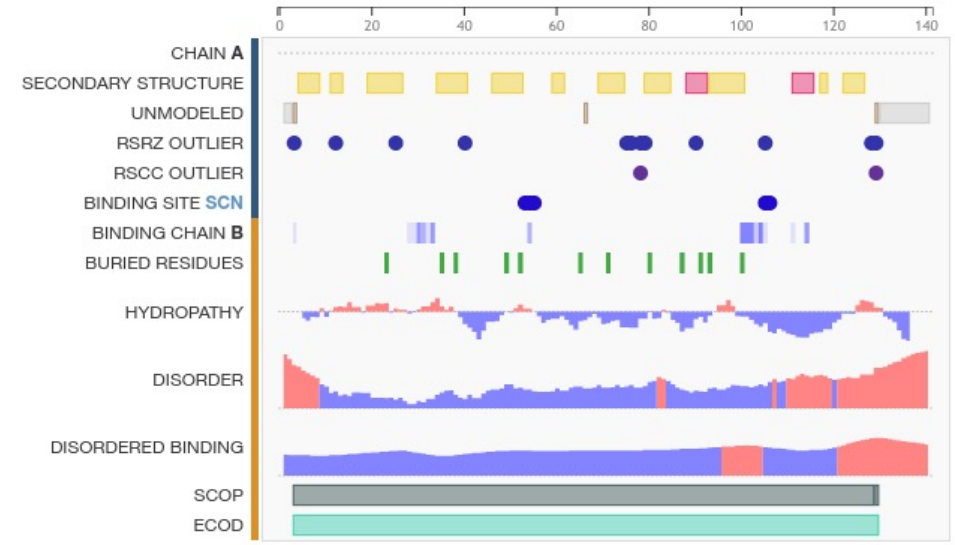
Help Back

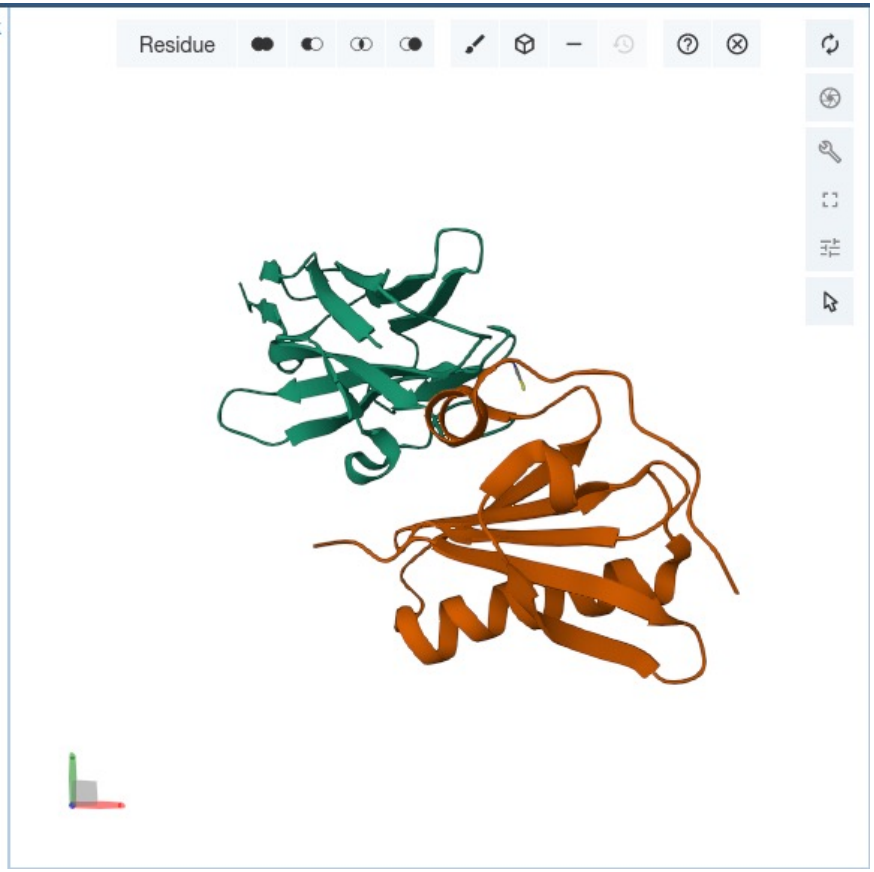
Structure of the nanobody-stabilized gelsolin D187N variant (second domain)

Chain

A

Gelsolin nanobody - Lama glama





In practice

Using OpenMM on Google Colab

- We'll test OpenMM on Google Colab to run molecular dynamics simulations without the need for installing any software on your local machine.
- **Google Colab** is a free Jupyter environment that allows you to run Python code in the cloud. GPUs runtimes are available.
- To use OpenMM on Google Colab or locally, open the provided notebook (read the comments)

Search or jump to... / Pull requests Issues Codespaces Marketplace Explore

giorginolab / OpenMM-Tutorial-UniPD-2023 Public

Edit Pins | Unwatch 1 | Fork 0 | Star 0

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main | 1 branch | 0 tags | Go to file | Add file | Code

tonigi test editing locally 5264d95 8 minutes ago 12 commits

results	import results	2 hours ago
OpenMM_2023.ipynb	test editing locally	8 minutes ago
README.md	Create README.md	yesterday
gsn-nb.py	preview	yesterday
openmm_fuxreiter_2023_r1.pdf	preview	yesterday

README.md

OpenMM-Tutorial-UniPD-2023

OpenMM class, prof. Fuxreiter's course at UniPD

About

OpenMM class, prof. Fuxreiter's course at UniPD

Readme
0 stars
1 watching
0 forks

Releases

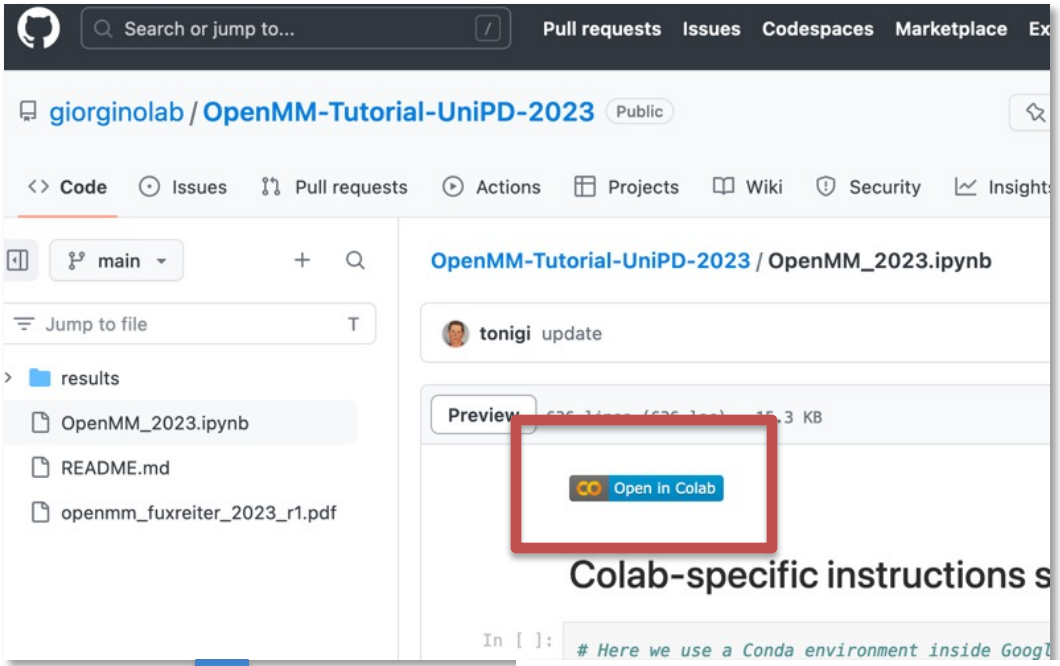
No releases published
Create a new release

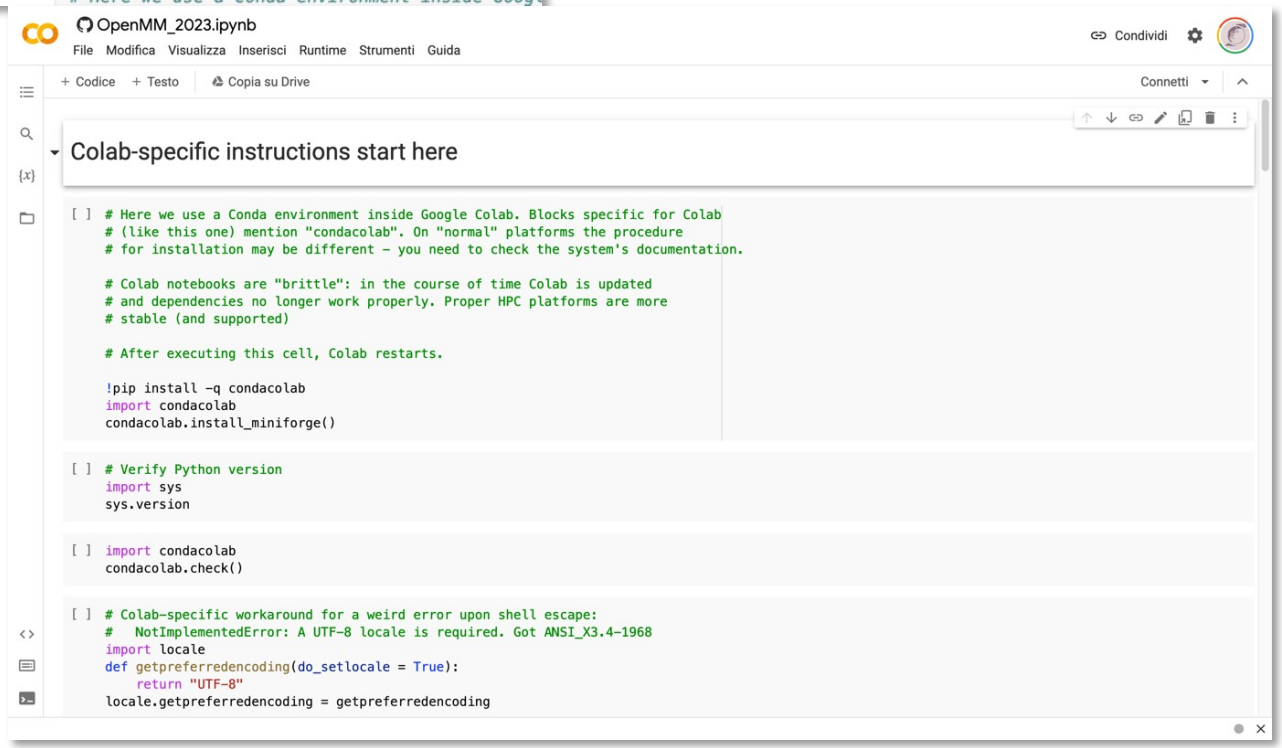
Packages

No packages published
Publish your first package

Languages

Jupyter Notebook 84.4%
Python 15.6%


Search or jump to...
Pull requests
Issues
Codespaces
Marketplace
giorginolab / OpenMM-Tutorial-UniPD-2023
Public
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
main
Jump to file
results
OpenMM_2023.ipynb
README.md
openmm_fuxreiter_2023_r1.pdf
OpenMM-Tutorial-UniPD-2023 / OpenMM_2023.ipynb
tonigi update
Preview
Open in Colab
Colab-specific instructions s
In [ ]:
Here we use a Conda environment inside Googl


OpenMM_2023.ipynb
File Modifica Visualizza Inserisci Runtime Strumenti Guida
Condividi
+ Codice + Testo
Copia su Drive
Connetti
Colab-specific instructions start here
[ ] # Here we use a Conda environment inside Google Colab. Blocks specific for Colab
(like this one) mention "condacolab". On "normal" platforms the procedure
for installation may be different - you need to check the system's documentation.
Colab notebooks are "brittle": in the course of time Colab is updated
and dependencies no longer work properly. Proper HPC platforms are more
stable (and supported)
After executing this cell, Colab restarts.
!pip install -q condacolab
import condacolab
condacolab.install_miniforge()
[ ] # Verify Python version
import sys
sys.version
[ ] import condacolab
condacolab.check()
[ ] # Colab-specific workaround for a weird error upon shell escape:
# NotImplementedError: A UTF-8 locale is required. Got ANSI_X3.4-1968
import locale
def getpreferredencoding(do_setlocale = True):
return "UTF-8"
locale.getpreferredencoding = getpreferredencoding

…when done…

Visualize

- After you have done the simulation, load the minimized PDB and output.dcd in PyMOL
- What about PBCs? Fix with: `pbc_unwrap` …

Questions

- How many atoms?
- How many residues?
- Disulfide bridges?
- How many trajectory frames?
- Simulation length in *actual* time?

More questions

- Does density change? Should it?
- What is the box size? Is it appropriate?
- Relaxation time?
- Plot the log file

Conclusion

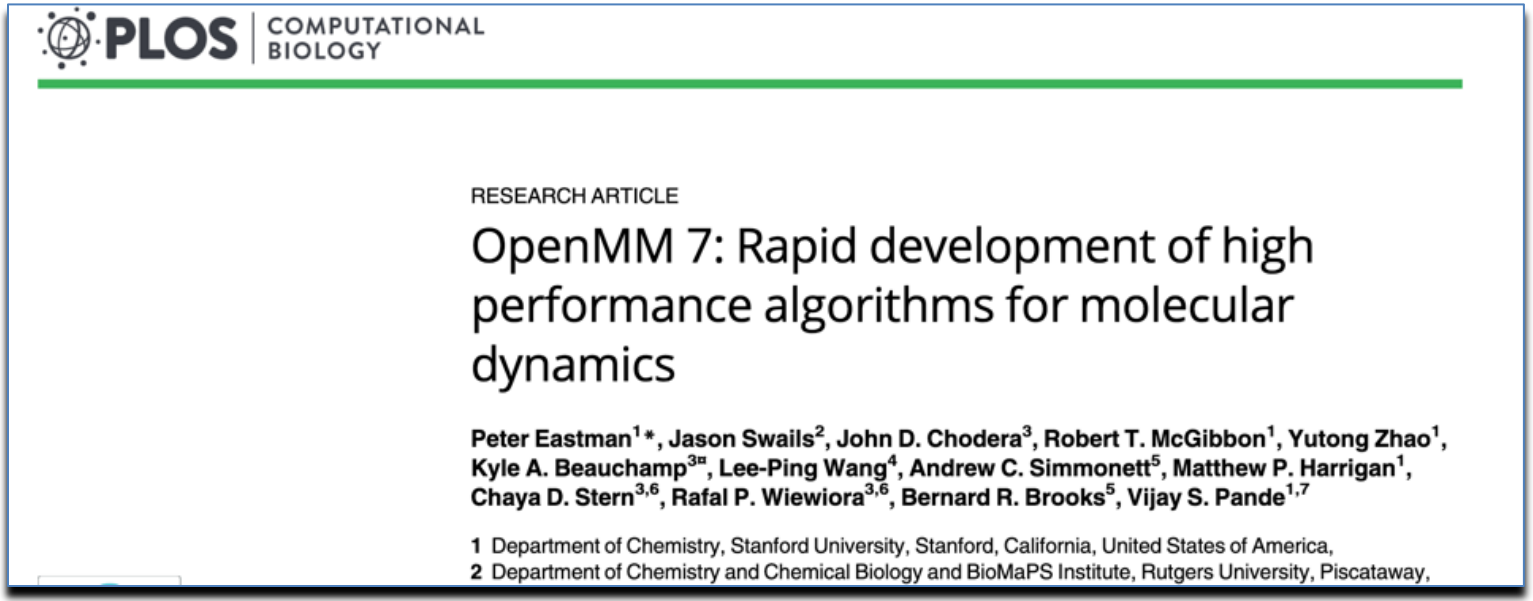
Conclusion

- OpenMM is a powerful tool for molecular dynamics simulations
- Good, if fragmented, documentation
- With its customizable force fields and integrators, it can be used to study a wide range of atomistic systems, e.g.
  - "toy" polymers
  - all-atom MD with major FFs
  - ANN potentials

Resources for learning OpenMM

- OpenMM.org website and documentation
- GitHub repository with examples and tutorials
- Community forums and mailing lists for support and discussion

- See also
  - OpenMMtools
  - https://openforcefield.org/
  - HTMD, ACEMD
  - https://github.com/openmm/pdbfixer
  - Charmm-GUI



PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

OpenMM 7: Rapid development of high performance algorithms for molecular dynamics

Peter Eastman[1]*, Jason Swails[2], John D. Chodera[3], Robert T. McGibbon[1], Yutong Zhao[1], Kyle A. Beauchamp[3¤], Lee-Ping Wang[4], Andrew C. Simmonett[5], Matthew P. Harrigan[1], Chaya D. Stern[3,6], Rafal P. Wiewiora[3,6], Bernard R. Brooks[5], Vijay S. Pande[1,7]

1 Department of Chemistry, Stanford University, Stanford, California, United States of America,
2 Department of Chemistry and Chemical Biology and BioMaPS Institute, Rutgers University, Piscataway,

End