3.1.1 Basics: Analyzing molecular simulations

- This module will period will consist of a mini-lecture and jupyter notebook exercises about
  - equilibration versus production
  - aligning trajectories
  - visualizing trajectories in VMD
  - root mean square properties
  - estimating thermodynamic expectations
- After this module, you should be able to
  - answer the following questions
    - what is the purpose of ignoring an equilibration period?
    - what is a potential of mean force?
    - what is true regarding the joint and marginal probabilities of independent events?
  - perform the following tasks
    - load a trajectory into VMD
    - calculate the RSMD and RMSF of a molecular simulation
    - estimate the average of an observable in a molecular simulation

What is MD used to calculate?

- MD simulations may be used to
  - predict events or sequence of events that are physically possible
  - estimate statistical averages of
    - configurational properties, e.g.
      - average distance between two residues
      - histogram of an angle between three domains
      - populations of certain conformations
    - rates
- Statistical estimation is based on the assumption of ergodicity - that the time average is equal to the ensemble average

Some python packages analyzing MD simulations

- juypter - for interactive coding notebooks
- pandas - for data analysis
- mdanalysis - for loading and analyzing MD trajectories
- pymbar - for calculating free energies. also contains equilibration detection

- For examples of using these packages, see
  - github.com/daveminh/Chem456/static_files/tutorials/ubq_wat-md/3-analysis for a simulation of ubiquitin
  - github.com/CCBatIIT/modelingworkshop/exercises/mpro/RIKEN_trajectory/analysis for a simulation of MPro

What is equilibration?

- The early part of a simulation is biased by the initial configuration
  - macromolecule structures start in a local minimum
  - water and ions are placed somewhat arbitrarily
  - box size is probably too large
- Equilibration is the time a system takes to reach a representative configuration
- Generally, simulation results from the equilibration period are ignored
  - The samples actually used to calculate averages are known as the production
  - It is not completely essential to ignore equilibration

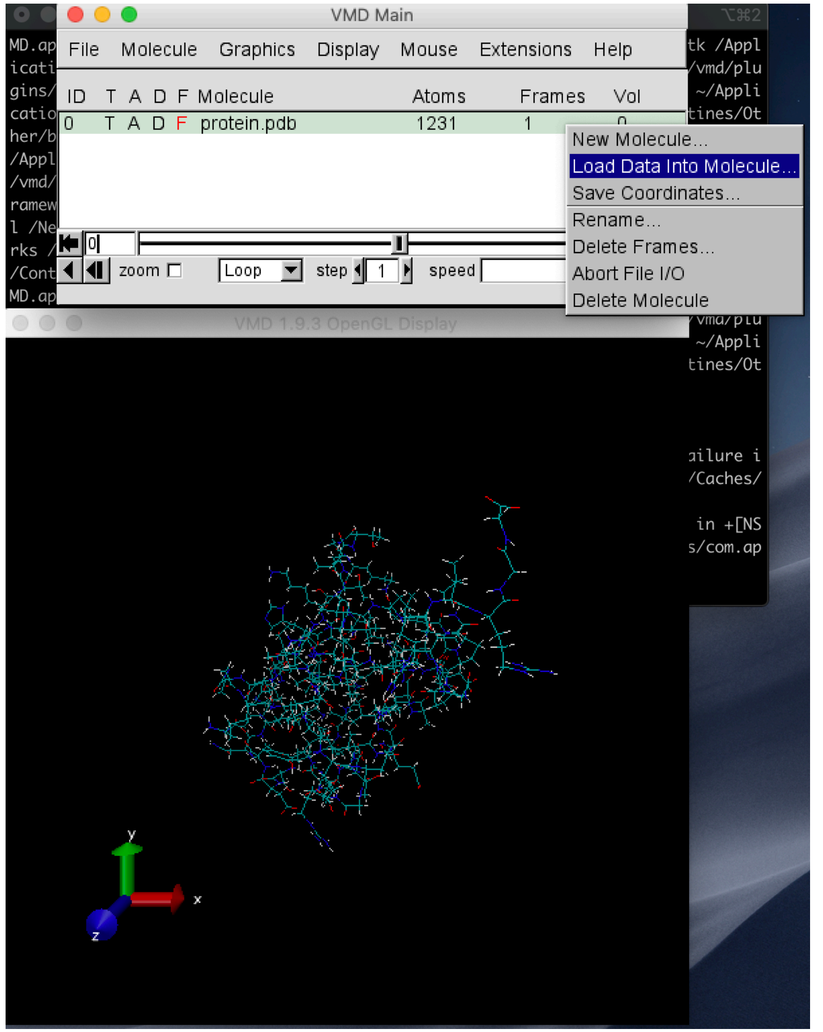
- See Equilibration.ipynb, which illustrates these points for a simulation of ubiquitin

How is equilibration time determined?

- Arbitrarily
- Once a key property, e.g. the root mean square deviation (RMSD), is stabilized
- By maximizing the effective sample size [Chodera, 2016]
  - A short equilibration leads to a long estimate of the time for the sample to be independent
  - A long equilibration reduces the number of samples
- Not at all
- Equilibration time
  - may look different for different properties
  - if properties are independent, slow equilibration of one may not affect estimation of another

An unaligned trajectory of ubiquitin

- can be visualized by
  - loading a model into VMD
  - loading the trajectory into the model
- molecules, especially water, can be split across a periodic box
- you probably don't need to see all water
- the molecule freely diffuses



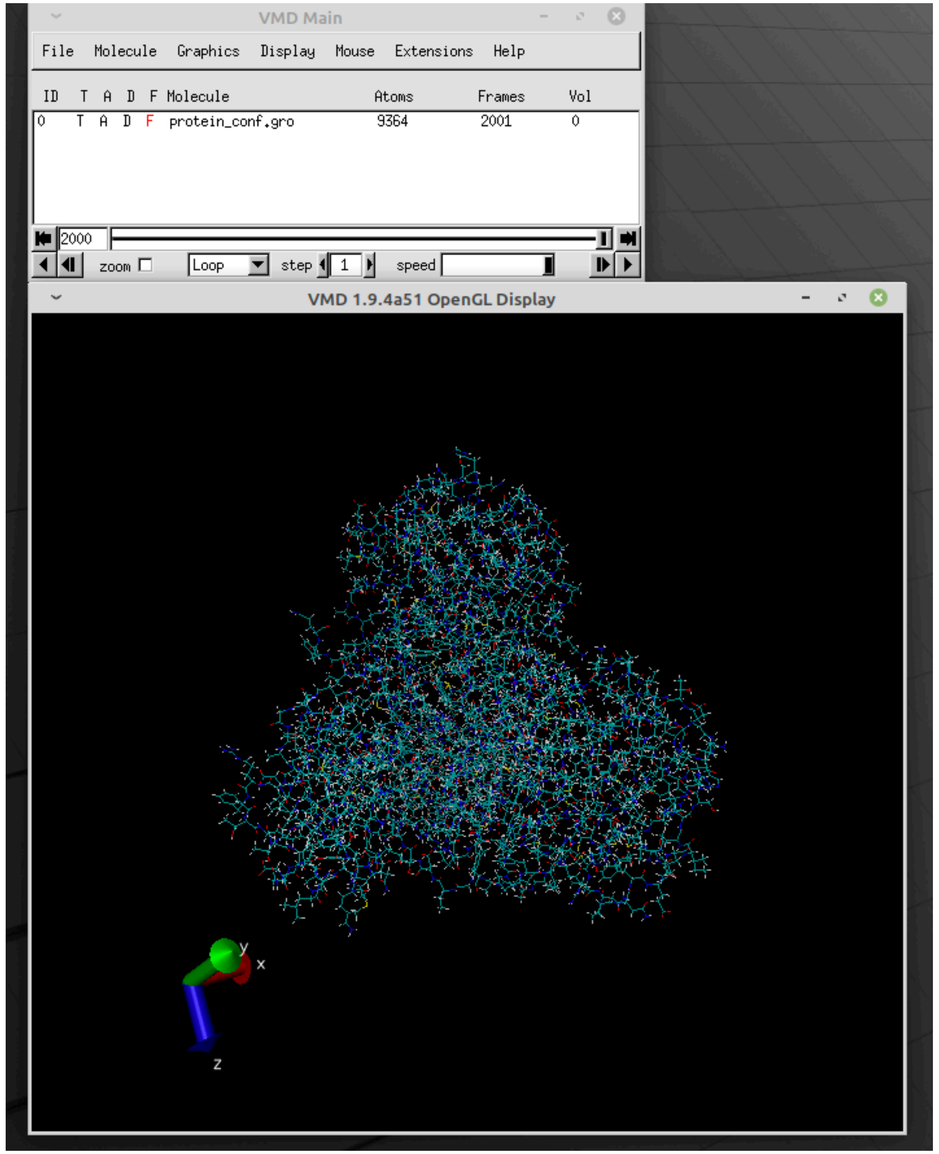
Structural alignment

- Other than identifying equilibration time, structural alignment is usually one of the first tasks of MD analysis
- Why?
  - In most MD simulations, molecules freely diffuse around the box
  - We are usually
    - uninterested in the overall translation and rotation,
    - interested in fluctuations relative to the macromolecule
- Alignment is often based on a rigid-body translation and rotation to minimize the root mean square deviation (RMSD)

- See Alignment.py, which performs a structural alignment for a series of simulations of ubiquitin and outputs a trajectory of the protein by itself.

Visualizing the main protease of SARS-CoV-2

- Let's visualize a trajectory of the main protease of SARS-CoV-2
- You can download it onto your virtual machine with the following script:
  - (base) chemuser@ChemBox:~/Documents/modelingworkshop/exercises/mpro/RIKEN_trajectory$ source get_trajectory.sh
- After starting VMD,
  - create a molecule from sarscov2-10921231-structure/protein_conf.gro, a Gromacs GRO file
  - load data into the molecule from traj_protein_snap_every1ns/protein_snap_every1ns_*N*to*M*us.xtc, a Gromacs XTC Compressed Trajectory



Root mean square analysis

$$\mathrm{RMSD}(v, w) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\, v_i - w_i \,||^2}$$

- Root mean square deviation (RMSD)
  - describes the difference between two structures
  - usually based on a subset of atoms
  - *i* is an index over atoms
- Root mean square fluctuation (RMSF)
  - describes the fluctuations of a specific atom, e.g. alpha carbon, over the course of a simulation
  - usually described per residue, identifying relatively flexible regions of a protein
  - *i* is an index over configurations
- Both require structural alignment

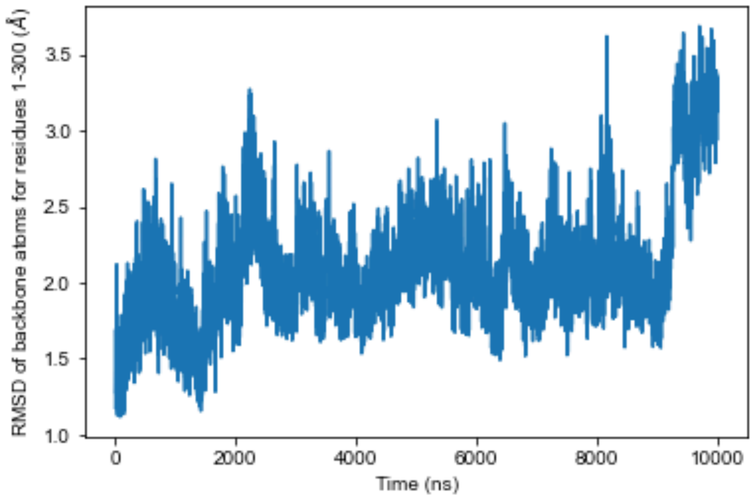
Exercise: RMS calculations with MDAnalysis

- Load and run the jupyter notebook https://github.com/CCBatIIT/modelingworkshop/exercises/mpro/RIKEN_trajectory/analysis/RMS.ipynb, which shows different types of RMS analysis for the RIKEN simulation of the main protease of SARS-CoV-2
- Based on the notebook
  - what is a reasonable number of frames to discard as equilibration?
  - how do the RMSD between alpha carbons and backbone atoms compare?
  - how does the flexibility of the protein vary between monomers
  - which part of the protein is most flexible?
- Based on the included examples and the MDAnalysis atom selection language, write your own code snippet to calculate the RMSD fo backbone atoms for residues 1-300

Solution to code snippet

In [19]:
```
# Evaluate the RMSD of backhone carbons
# with respect to the reference frame
from MDAnalysis.analysis import rms
R = rms.RMSD(sim, ref, select="backbone and resid 1-300")
R.run()
rmsd_backbone_to300 = R.rmsd

plt.plot(rmsd_backbone_to300[:,1]/1000, rmsd_backbone1_70[:,2])
plt.xlabel('Time (ns)')
plt.ylabel('RMSD of backbone atoms for residues 1-300 ($\\AA$)');
```



Thermodynamic Expectations

- One key reason to do a molecular simulation is to estimate expectation values
- The definition of an expectation value of the observable A is,

  $\langle A \rangle = \int A(x)p(x)dx$, where p(x) is the probability of x
- For an unbiased MCMC simulation, the typical way to estimate an expectation value is the sample mean, $\bar{A} = \frac{1}{N}\sum_{n=1}^{N} A(x_n)$, over N samples after

  equilibration

What A?

- The observable A can be anything that can be measured from a configuration
  - distance between two atoms
  - an angle between two protein domains
  - an indicator function H(x) that is 1 if $x_l < x < x_r$ and 0 otherwise
    - this expectation is the probability of x being in this range
    - it is useful for a histogram

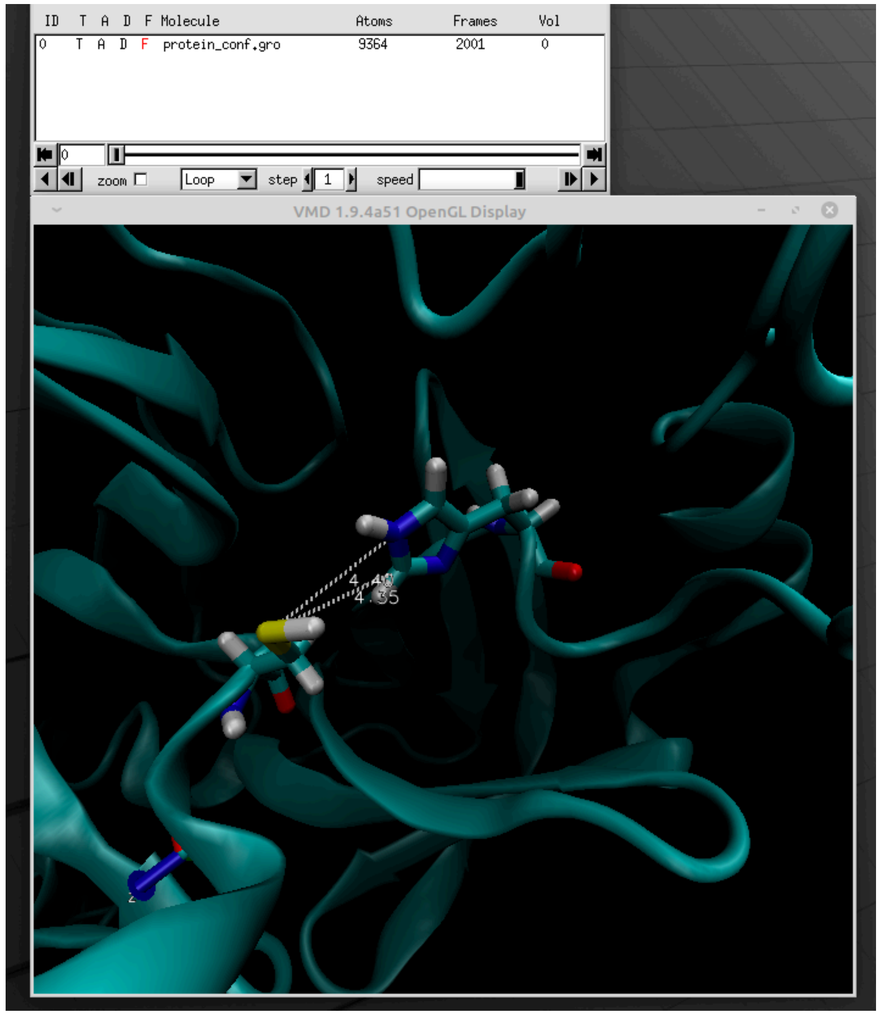
Potential of mean force

- A potential of mean force
  - $p(x) = -RT \log \langle H[z(x)] \rangle$
    - H[.] is an indicator function
    - z(.) is an order parameter, a function of the configuration
    - x is the configuration
  - is the probability of a certain order parameter
  - a potential energy whose gradient would give rise to the mean force on a particle
- A histogram is a simple way to estimate the expectation for a set of bins
- Let's be more concrete and practical…

The catalytic dyad of MPro

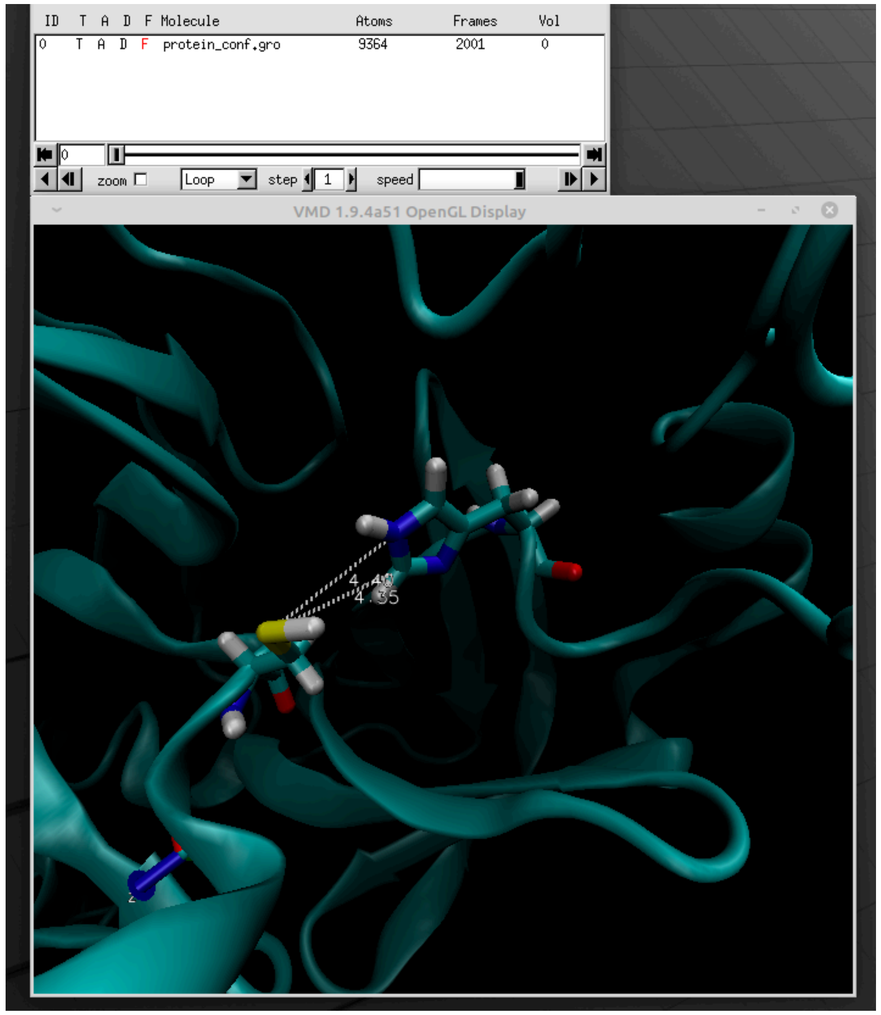
- The main protease of SARS-CoV-2 is
  - a dimer
  - believed to have half-of-the-sites activity
- Chen et al (2006)
  - ran 4 10 ns simulations of MPro from SARS-CoV-1
  - reported the fraction of frames with hydrogen bonds between Cys145 SG and His41 ND1/NE1
  - suggested that during dimerization one subunit is active and the other inactive
- At 10 μs, the RIKEN simulation is ~1000x longer!

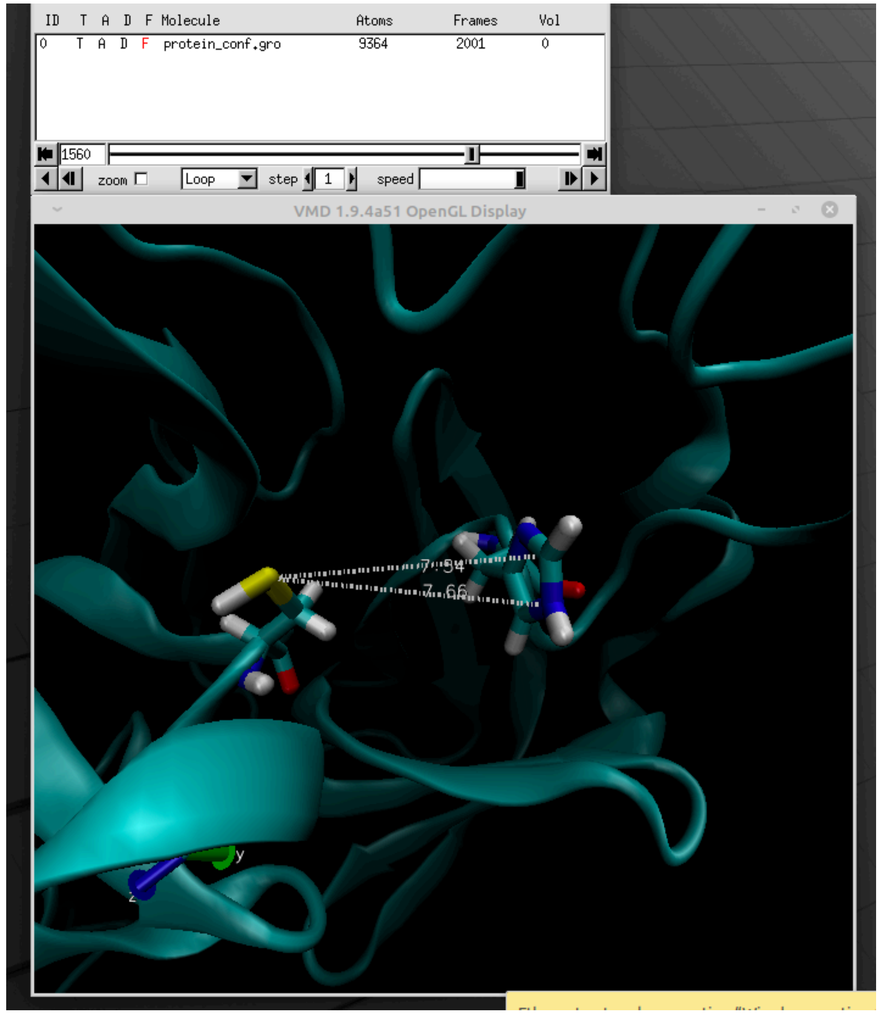
- Exercise: Use VMD to label the distance between Cys145 SG and His41 ND1/NE1



An alternate His41

- In some MD simulation frames, the histidine ring seems to flip away from Cys 145. This could be important!
- Did you notice anything else that I didn't?





Exercises: Distances between the catalytic dyad

- Load and run the jupyter notebook https://github.com/CCBatIIT/modelingworkshop/exercises/mpro/RIKEN_trajectory/analysis/dyad-stability.ipynb
- Answer these questions:
  - Are probability densities of the dyad distance the same in each monomer?
  - Is there any apparent correlation between subunits in the dimer?
  - Are probabilities of forming a short contact in each subunit independent? Write a code snippet based on these hints
    - Use `d_cutoff = 4.25`
    - `pA = np.sum(d_monomerA<d_cutoff)/len(d_monomerA)` is the fraction of frames with a short contact in monomer A
    - `pAB = np.sum(np.logical_and(d_monomerA<d_cutoff, d_monomerB<d_cutoff))/len(d_monomerB)` is the fraction of frames with a short contact in both monomer A and monomer B
    - If two events are independent, $p(A, B) = p(A)p(B)$.

Solution to the code snippet

Joint probabilities of short dyad distances

In [18]:
```
d_cutoff = 4.25
pA = np.sum(d_monomerA<d_cutoff)/len(d_monomerA)
pB = np.sum(d_monomerB<d_cutoff)/len(d_monomerB)
pAB = np.sum(np.logical_and(d_monomerA<d_cutoff, d_monomerB<d_cutoff))/len(d_monomerB)
print('The fraction of frames with a short dyad distance')
print('in monomer A is', pA)
print('in monomer B is', pB)
print('in both monomers is', pAB)
print('If the monomers were indepedent, the probabilities would be', pA*pB)
```

```
The fraction of frames with a short dyad distance
in monomer A is 0.0475
in monomer B is 0.0824
in both monomers is 0.005
If the monomers were indepedent, the probabilities would be 0.003914
```

- How conclusive are these results?

Review

- what is the purpose of ignoring an equilibration period?
- what is a potential of mean force?
- what is true regarding the joint and marginal probabilities of independent events?

References

- Chodera, J. D. A Simple Method for Automated Equilibration Detection in Molecular Simulations. Journal of Chemical Theory and Computation 2016, 12 (4), 1799–1805. https://doi.org/10.1021/acs.jctc.5b00784.
- Chen, H.; Wei, P.; Huang, C.; Tan, L.; Liu, Y.; Lai, L. Only One Protomer Is Active in the Dimer of SARS 3C-like Proteinase. J. Biol. Chem. 2006, 281 (20), 13894–13898. https://doi.org/10.1074/jbc.M510745200.

Some other software

- MDTraj: http://mdtraj.org/1.9.3/index.html
- ProDy: http://prody.csb.pitt.edu/tutorials/trajectory_analysis/trajectory.html