3/5/2020 Week 8 Module 2
Analysis of molecular dynamics simulations 2

- This module will consist of
  - an explanation of dimensionality reduction, conformational clustering, and Markov State Models in the analysis of biological MD
  - a tour of an analysis of a ubiquitin simulation
    - based on the python package MDAnalysis
    - that you can follow for your own system
- At the end of this module, you should be able to answer the following questions:
  - What is dimensionality reduction? What is one way to do it?
  - What is clustering and why is it useful? What is one way to do it?
  - What are Markov State Models and why are they useful?
- Hopefully you can modify the scripts that I used to analyze your own systems.
- Let me know if you feel that the class is going too quickly

Installing MD analysis packages w/ conda

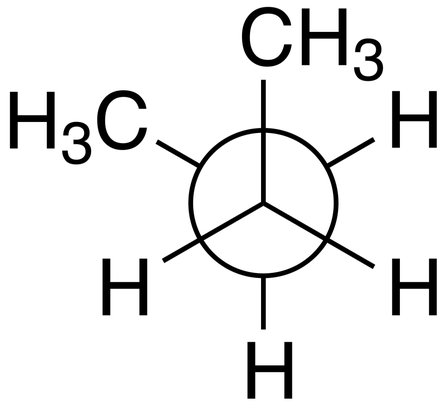
- conda create --name mdanalysis
- conda activate mdanalysis
- conda config --add channels conda-forge
- conda install jupyter pandas mdanalysis pymbar
  - juypter - for interactive coding notebooks
  - pandas - for data analysis
  - mdanalysis - for loading and analyzing MD trajectories
  - pymbar - for calculating free energies. also contains equilibration detection.

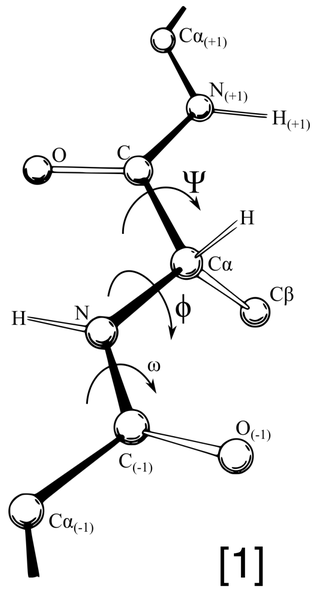
- If you want to try Markov State Models, you may need to downgrade python first
  - conda install python=3.7
  - conda install pyemma
    - pyemma - for Markov state models

Dimensionality reduction with principal component analysis

Dimensionality reduction

- Biomolecular simulations have 3N dimensions, but in practice, functionally relevant macromolecular motion can be described with a lot fewer.
- Cartesian coordinates (x, y, and z) of each atom are the most common description but molecular systems can be described by other coordinate systems
- You have probably already seen
  - the free energy of butane as a function of the dihedral angle
  - the Ramachandran diagram, where protein backbones are described by the ϕ and ψ angles



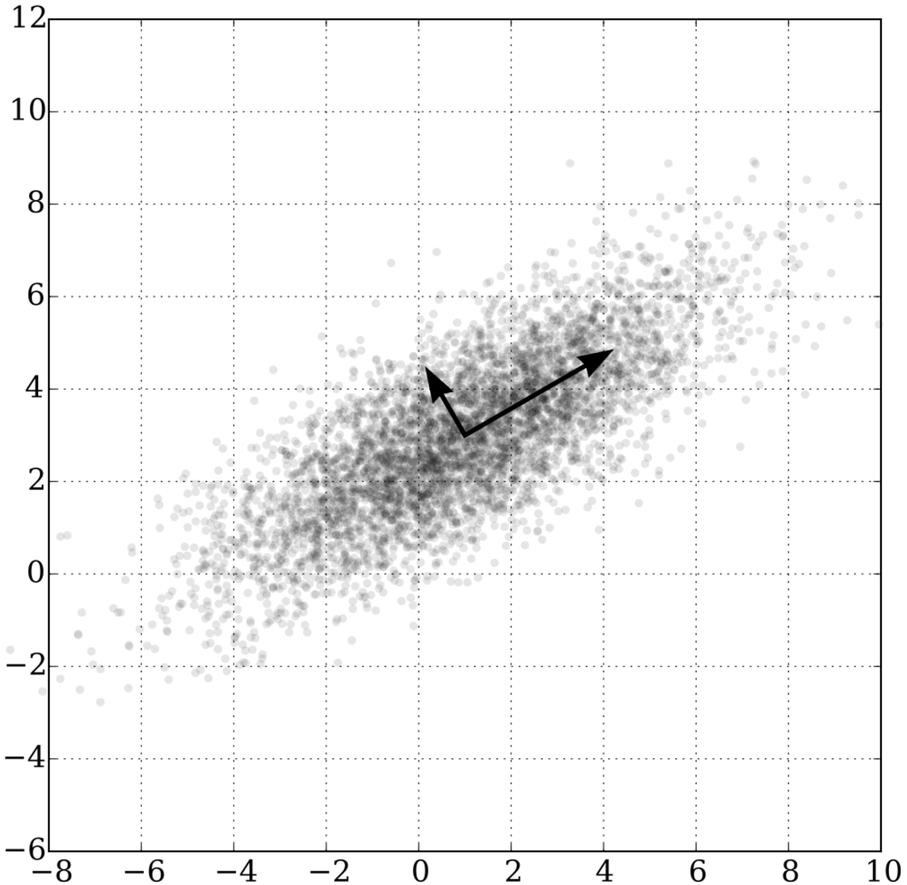


[1]

Principal component analysis (PCA)

- An *automated* way to do dimensionality reduction
- A linear transformation of coordinates in decreasing order of variance
  - First principal component has the largest variance
  - Second principal component has second largest variance
  - And so forth
- Dimensions can be reduced by keeping the highest-variance dimensions

- See https://en.wikipedia.org/wiki/Principal_component_analysis



[2]

The matrices of PCA

- **CV** = **PV**
- **C**: covariance matrix
  - C_{kl} is the covariance between dimensions k and l.
  - Usually empirically estimated from data.
- **V**: matrix of column *eigenvectors*
  - V_{kl} is the
    - importance of the original coordinate k
    - in the transformed coordinate l.
    - known as the principal components
    - the columns are orthonormal vectors
- **P**: diagonal matrix of *eigenvalues*
  - $P_{kl} = \lambda_{kl}$ if k = l
  - $P_{kl} = 0$ otherwise
  - variances in transformed coordinate system
  - scaling of the eigenvectors
- All three matrices have the same size

PCA analysis of molecular simulation

- In a molecular simulation, it could be helpful to visualize
  - the principal components themselves
  - the variance and cumulated variance
  - the projections onto the principal components in one or more dimensions
    - time series
    - histograms

- See PCA.ipynb, which shows PCA analysis for a simulation of ubiquitin

Time-lagged independent component analysis (TICA)

- Like PCA, based on eigenvector and eigenvalues of a matrix
  - autocorrelation instead of covariance matrix
  - isolates slow motions into coordinates
- Often used in Markov state models (MSMs)
- Slow is often but not always functionally relevant

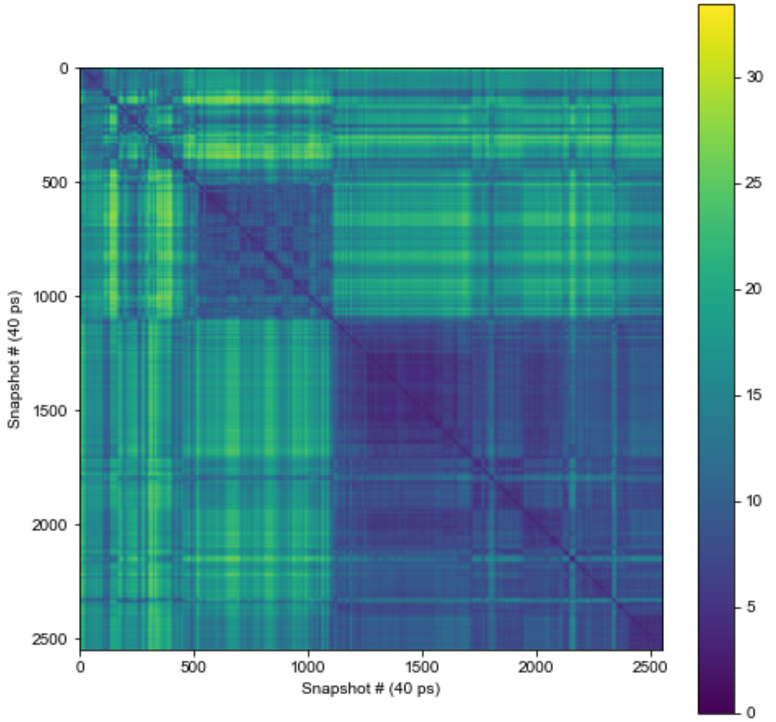
Conformational clustering

Clustering

- MD simulations yield configurations in continuous space
- Clustering methods group together similar configurations (or, in a more general data science context, observations)
- Clustering is useful
  - interpreting simulation results
  - calculating thermodynamic and kinetic properties
    - predicted populations of conformations
    - predicted rates of transitions (e.g. Markov state models [1, 2])
  - selecting representative configurations for molecular docking [3]

Distance matrices in clustering

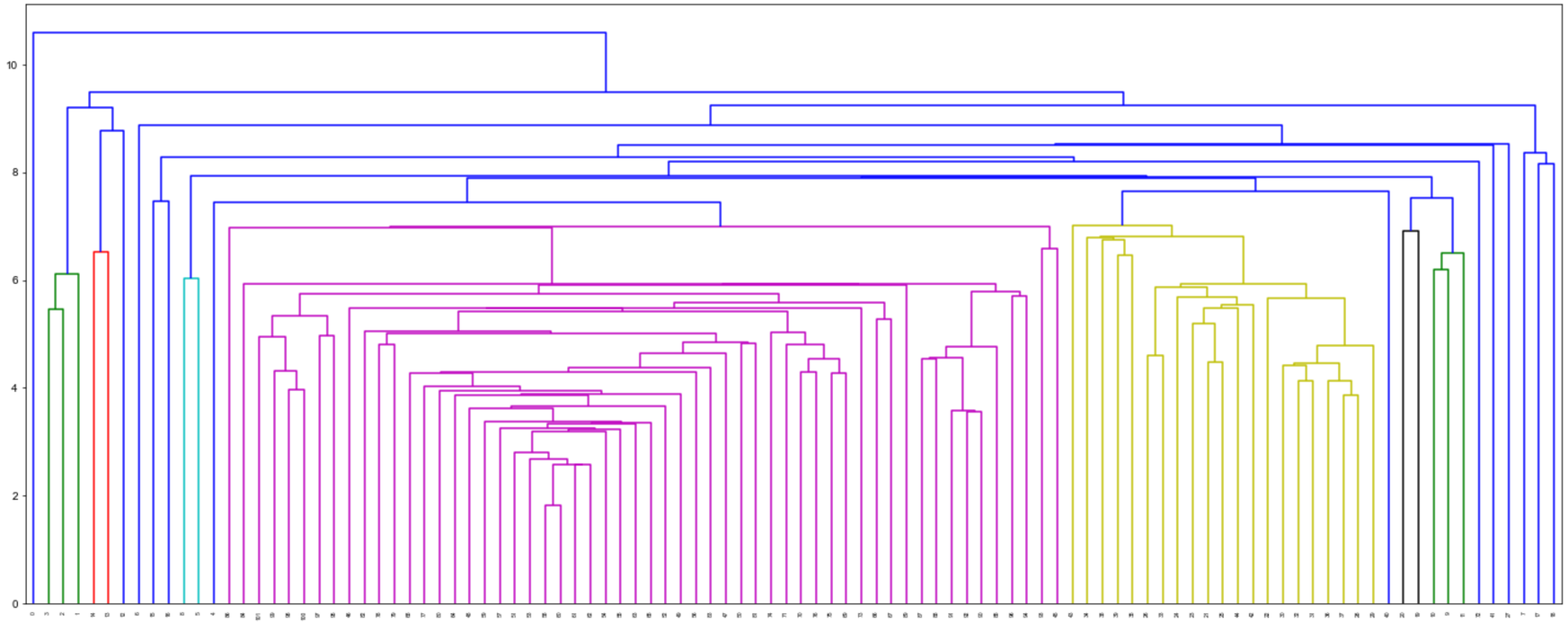
- Almost all clustering algorithms employ a distance matrix
- In a matrix **D**, D_{kl} denotes the distance between observation k and l
- Distance matrices include [3]
  - the RMSD
    - between alpha carbons/all heavy atoms
    - in a entire protein/in a region of the protein
  - based on occupancy fingerprints
    - a 3D grid with zero or one depending whether a point is close to an atom
    - If M_{ab} is the number of points where one grid has *a* and the second *b*,
      - the overlap is $M_{10} + M_{01}$
      - the Tanimoto similarity is $-\log_2[M_{11}/(M_{11} + M_{10} + M_{01})]$
      - the Jaccard distance is $[(M_{11} + M_{01})/(M_{11} + M_{10} + M_{01})]$
  - Euclidean distance between principal components (like the RMSD, PCA can be based on different subsets of coordinates)



Heat map of Euclidean distances between top 20 principal components in a simulation of ubiquitin

Agglomerative hierarchical clustering

- Closest pair of observations (or clusters) are grouped together until all observations are in groups
- There are
  - Different definitions of distances between observations and clusters
  - Different ways to go from linkage matrix to clusters
- See Clustering.ipynb, which shows clustering analysis for a simulation of ubiquitin



Dendrogram of hierarchical clustering for every 1 ns for a simulation of ubiquitin

Markov State Models

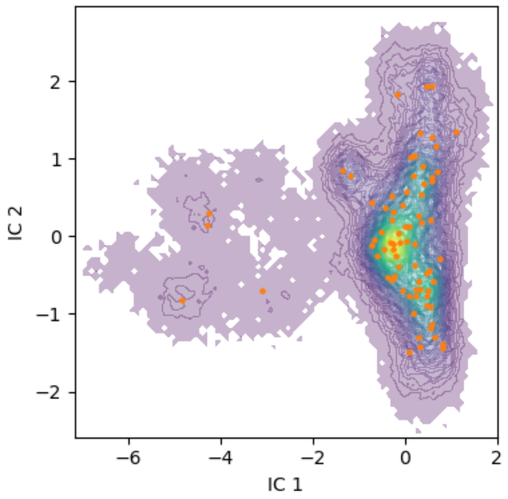
Markov State Models

- MSMs
  - are similar to chemical kinetics models, where there are states and rates of transitions between states
  - there are states and *transition matrices* - which describe the probability of transitions to other states after a certain amount of time
  - are based on Markov chains, which assume that the future of a system only depends on its current state
- They can be used to calculate
  - the rates of transitions between any pair of states
  - the most probable pathways between any pair of states
  - the equilibrium probability of any state
- MSMs are useful because they
  - can combine information from short MD trajectories
  - piece together local equilibria into a global picture

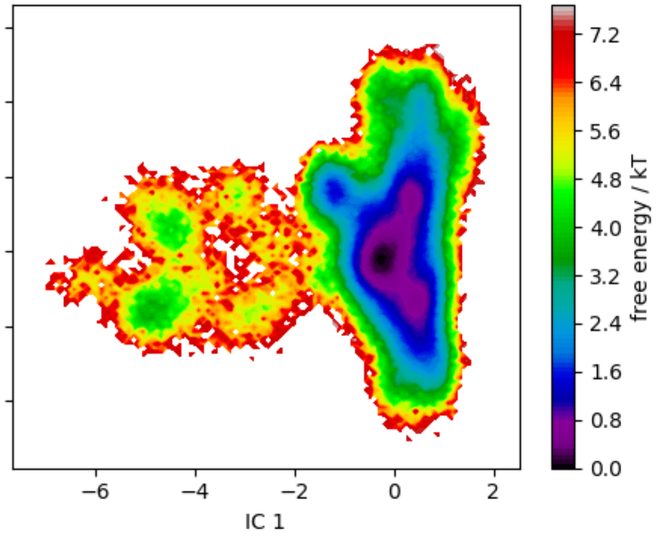
MSM of T4 lysozyme

- 1 μs MD trajectories initiated from 4 crystal structures: open, 1qud (blue); closed, 2otz (pink); intermediate, 183l (gold); intermediate, 3dn3 (green)
- Clustering using TICA of protein backbone (from folding)

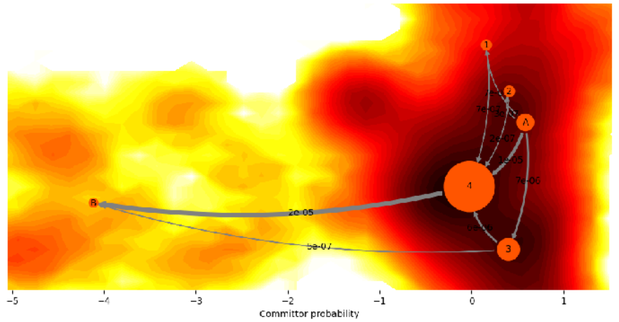
Cluster centers



Reweighted free energy surface



Kinetic description



15

References

- [1] Pande, V. S.; Beauchamp, K. A.; Bowman, G. R. Everything You Wanted to Know about Markov State Models but Were Afraid to Ask. Methods (San Diego, Calif.) 2010, 52 (1), 99–105. https://doi.org/10.1016/j.ymeth.2010.06.002.
- [2] Chodera, J. D.; Noé, F. Markov State Models of Biomolecular Conformational Dynamics. Current Opinion in Structural Biology 2014, 25, 135–144. https://doi.org/10.1016/j.sbi.2014.04.002.
- [3] Xie, B.; Clark, J. D.; Minh, D. D. L. Efficiency of Stratification for Ensemble Docking Using Reduced Ensembles. Journal of Chemical Information and Modeling 2018, 58 (9), 1915–1925. https://doi.org/10.1021/acs.jcim.8b00314.

Some software

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

- For Markov State Models
  - MSMBuilder: http://msmbuilder.org/3.8.0/
  - PyEMMA: http://emma-project.org/latest/