Macrostates from clustering (ideas)

June 22, 2017

1 Josh's method, as reported in the SI of our paper (JCTC 2016, 12, 3473)

This is not even a clustering procedure: the idea here was to simply use RMSD thresholds to define the folded and unfolded macrostates consistent with the MFPTs reported by the Shaw group. One technical issue was how to assign a given MSM 'microstate' to a macrostate if the microstate straddled a threshold. Josh did the following:

- Select threshold values R_{min} and R_{max} (Folded=$\{$RMSD $< R_{min}\}$, Unfolded=$\{$RMSD $> R_{max}\}$) so that the MFPTs computed *directly from the Shaw trajectories* (i.e., no use of a model) are consistent with the Shaw values. We also checked that those thresholds make sense when we visualize the evolution of the RMSD vs time.
- Cluster the snapshots into the microstates suitable for a MSM.
- Decide the "membership" of the microstates based on their distribution on the coarser set $\{$Folded, Intermediate-Region, Unfolded$\}$. If $p($Folded $> 0.9)$ then it is considered folded, similarly if $p($Unfolded $> 0.9)$ is considered Unfolded, everything else would belong to the intermediate region.
- Recompute the MFPTs *directly from the Shaw trajectories, now discretized via MSM microstates assigned to macrostates* to check if after that classification the MFPTs are still consistent with the values reported.

Clearly, a very careful definition of the macrostates based on the microstates was not our goal here, since that could be a complex problem by itself.

2 Clustering based on the MFPT matrix

Here we propose a more systematic, but fairly simple idea based on a procedure used in some previous work [J. Chem. Theory Comput. 2010, 6, 3048–3057, which in turn was influenced by a paper of John's]. The idea is to perform hierarchical kinetic clustering of MSM microstates while ensuring a separation of timescales. We can use as a distance metric the MFPT, or more precisely, the "round-trip" time (MFPT(ij) + MFPT(ji)). A slight limitation is of the implementation suggested below is the use of (inexact) Markovian MFPTs, because the *direct* estimation would be very noisy and the non-Markovian estimate very expensive (we need to re-analyze the trajectory in every cycle of the clustering procedure and for every pair of microstates).

The simplest way would be a hierarchical (or progressive) clustering based on a cutoff t_{cut} and a factor m defining the targeted separation of timescales. If the round-trip time (t_{ij}) is less that t_{cut} then we merge the states. Here is a possible procedure:

1. Compute MFPT matrix M and add it to M^T to obtain the round-trip times $\{t_{ij}\}$
2. While $\min(\{t_{ij}\}) < t_{cut}$:
    - Merge the corresponding states
    - Recompute $\{t_{ij}\}$ (step 1) for merged states
3. Increase t_{cut} until the maximum remaining element of t_{ij} becomes less than a pre-defined multiple m (e.g., 10) of t_{cut}. Use as macrostates the two clusters from the prior step with the maximum $t_{ij} > m\, t_{cut}$

3 Clustering based on target milestones (JCP 2016, 145, 024102)

This procedure does not need any model and that is probably its strongest point. It is based on the identification of target microstates (core set) that optimize the metastability index among a set of trial microstates (we have to predefine the size of the core set). Once we identify the target states we can use them as cluster "centers". A given trial microstate S_i will belong to the cluster defined by the target state B if B is the most likely target state to be visited by a trajectory after hitting S_i. The general procedure will be:

1. Initial clustering of snapshots to obtain the trial microstates.
2. We have to decide how many target microstates we want, since that will be the number of clusters at the end.
3. Select the subset of target states that optimize the metastability index (the core set). We do not know exactly how many cores to use. But we can experiment with different numbers from 2 to 20. In other words, we can always go back to [2] and start again.
4. Compute the committor probabilities that map the trial states with the cluster "centers" (target states)

5. Check how close we are from the ideal partition into {Folded, Unfolded and Intermediate region}. There is no universal rule for this. It is easier to identify the folded state than define what is unfolded. So after clustering we might have to experiment with different criteria (for instance number of native contacts) and see how robust the observables are. We also may want to check for separation of timescales.

6. Go back to [2] if needed.