Multi-Ensemble Markov Models and 🚋 TRAM 🚋

Fabian Paul

20-Feb-2019

Outline

- Free energies
- Simulation types
  - Boltzmann reweighting
  - Umbrella sampling
  - multi-temperature simulation
  - accelerated MD

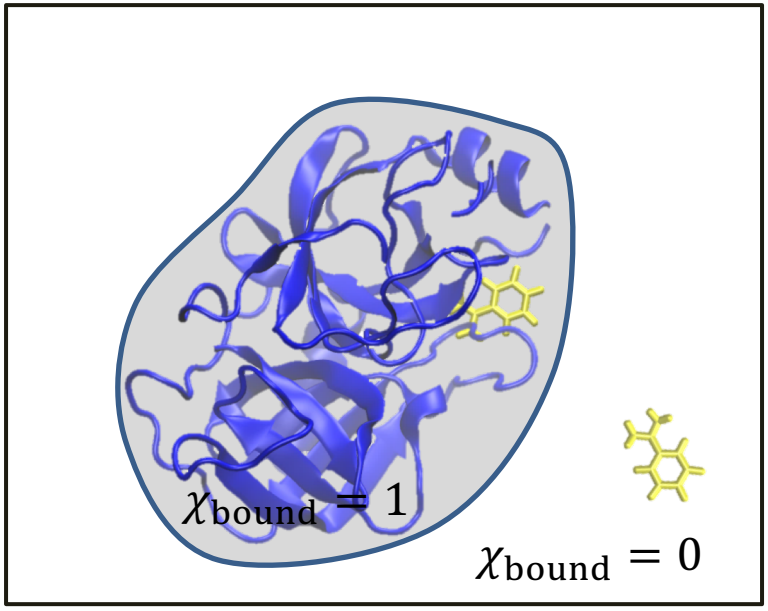
- Analysis methods
  - Weighted Histogram Analysis method + its problems
  - Multi Ensemble Markov Models and discrete TRAM

Free energy: definition and use

By “free energies” we mean :

A) – $k_B T$ times the logarithm of probabilities of different conformational states within one thermodynamic ensemble

$$P(\text{bound}) = \frac{\int \chi_{\text{bound}}(x) e^{-\beta U(x)} \mathrm{d}x}{\int e^{-\beta U(x)} \mathrm{d}x}$$

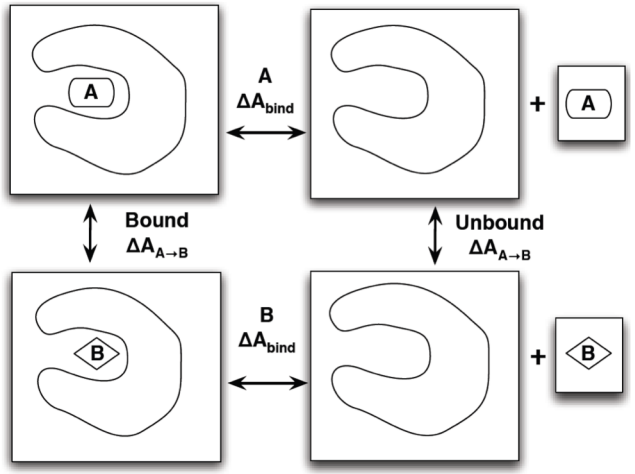


B) – $k_B T$ times the log of ratios of partition functions from different thermodynamic ensembles

$$e^{-\beta^{(1)} F^{(1)} + \beta^{(2)} F^{(2)}} = \frac{Z^{(1)}}{Z^{(2)}} = \frac{\int e^{-\beta^{(1)} U^{(1)}(x)} \mathrm{d}x}{\int e^{-\beta^{(2)} U^{(2)}(x)} \mathrm{d}x}$$

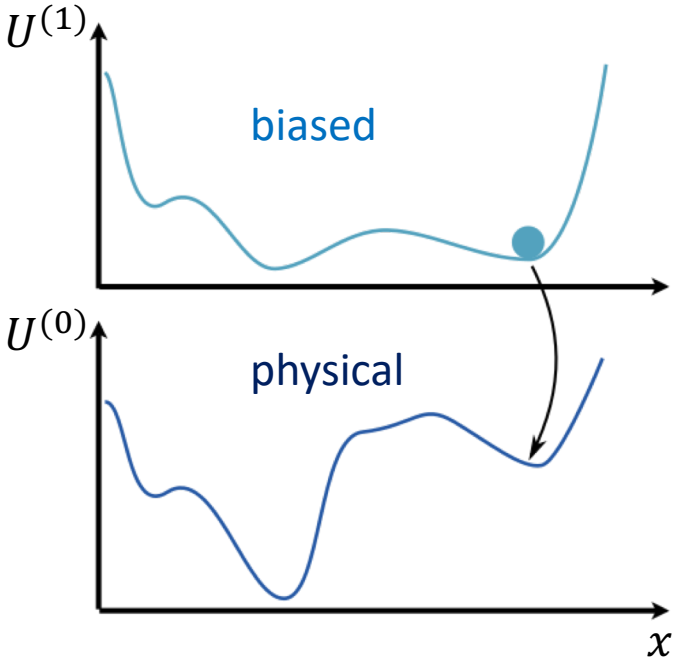
Uses:

- calculating entropy $S = -\left(\frac{\partial F}{\partial T}\right)_{V,N}$ or
- relative binding / solvation free energy



Boltzmann reweighting / importance sampling

Some systems have an interesting but improbable state or states that are separated by a high barrier. How can we investigate such states?



Expectation values in ensemble (0) are computed as:

$$\langle O \rangle^{(0)} = \int O(x) e^{-\beta U^{(0)}(x) + \beta F^{(0)}} \mathrm{d}x \approx \frac{1}{N} \sum_n^N O(x_n)$$

where $\boldsymbol{x}_n \sim p^{(0)}(\boldsymbol{x})$

$$\langle O \rangle^{(0)} = \int O(x) e^{-\beta U^{(1)}(x) + \beta F^{(1)}} \frac{e^{-\beta U^{(0)}(x) + \beta F^{(0)}}}{e^{-\beta U^{(1)}(x) + \beta F^{(1)}}} \mathrm{d}x$$

$$\approx \tfrac{1}{N} \sum_n^N O(x_n) e^{-\beta U^{(0)}(x) + \beta U^{(1)}(x) + \beta F^{(0)} - \beta F^{(1)}}$$

where $\boldsymbol{x}_n \sim p^{(1)}(\boldsymbol{x})$

- $U^{(0)}(x)$= the *unbiased* or the *physical* energy
- $U^{(1)}(x)$= the *biased* energy
- $U^{(1)}_{\mathrm{bias}}(x) = U^{(1)}(x) - U^{(0)}(x)$= the *bias* energy

Boltzmann reweighting / importance sampling

- $U^{(0)}(x)$= the *unbiased* or the *physical* energy
- $U^{(1)}(x)$= the *biased* energy
- $U^{(1)}_{\text{bias}}(x) = U^{(1)}(x) - U^{(0)}(x)$= the *bias* energy

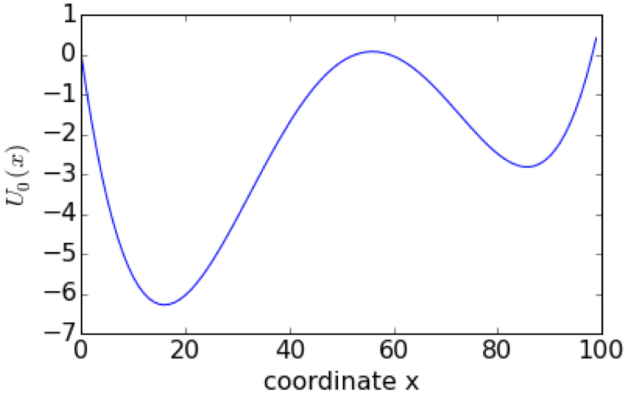
What is the optimal bias?

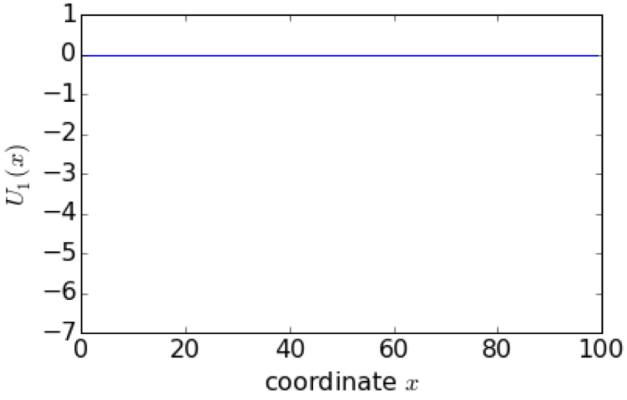
For a low-dimensional system, it would be efficient to sample from a flat energy landscape:

$$U^{(1)}(x) = 0$$

Allows good sampling of the minima and the barrier.

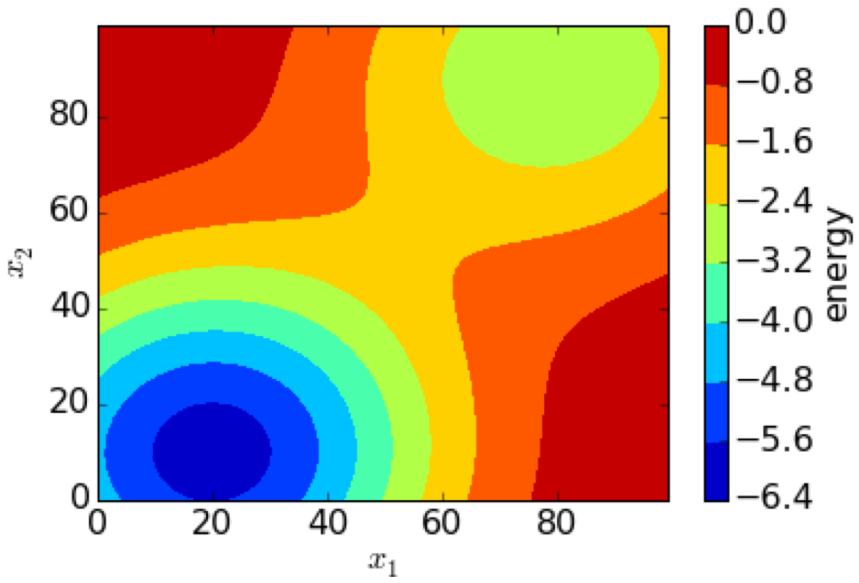
$$\Rightarrow U^{(1)}_{\text{bias}}(x) = -U^{(0)}(x)$$





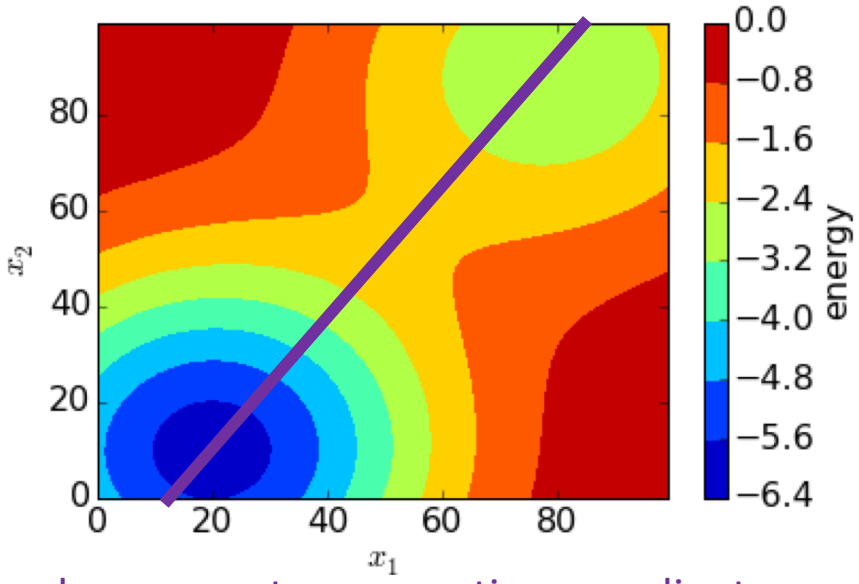
Importance sampling in high dimensions

- Sampling uniformly is not possible in high dimensional space like the conformational space.



Importance sampling in high dimensions

- Introduce an “order parameter” that connects the relevant minima in the energy landscape.

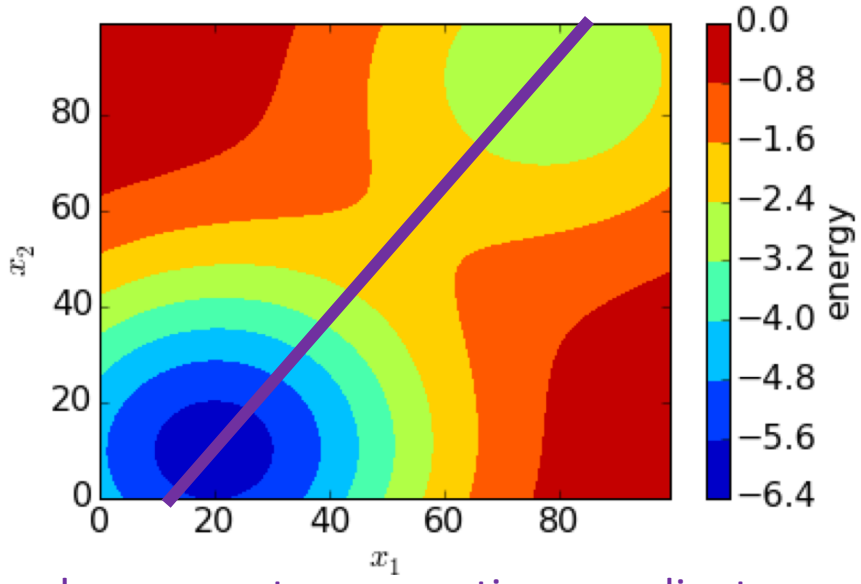


order parameter or reaction coordinate

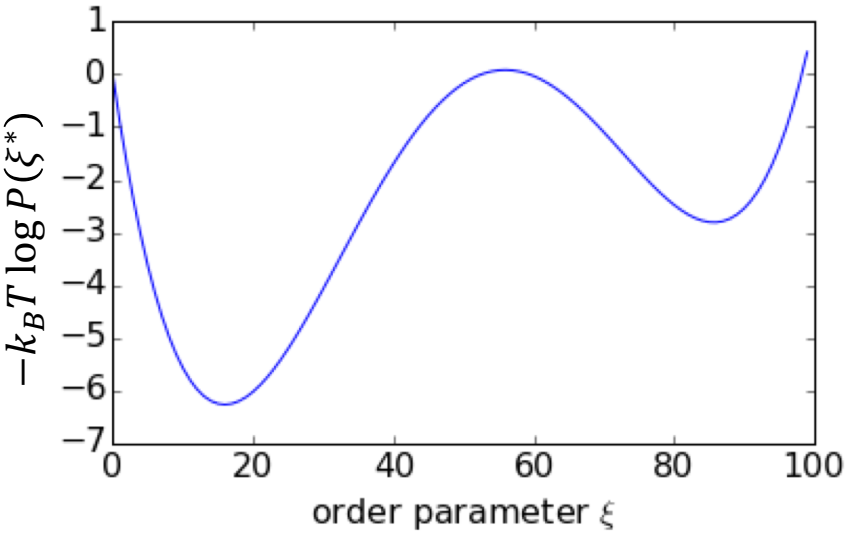
Importance sampling in high dimensions

- Sample uniformly along the order parameter.

$$P(\xi^*) \propto \int \delta(\xi(x) - \xi^*) e^{-\beta U(x)} \mathrm{d}x$$



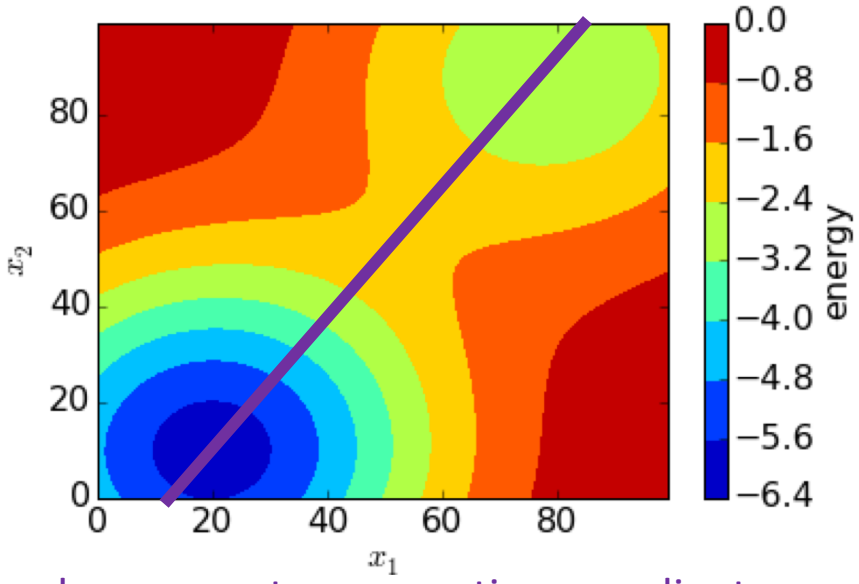
order parameter or reaction coordinate



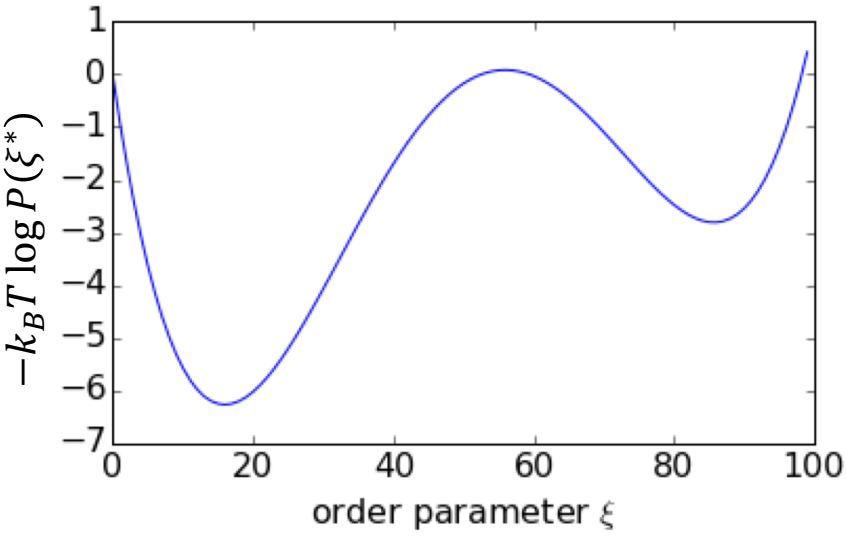
Importance sampling in high dimensions

- The ideal bias energy would be $k_B T \log P(\xi)$
  (minus the potential of mean force)
- Problem: computing $P(\xi)$ requires sampling from the unbiased distribution!

$$P(\xi^*) \propto \int \delta(\xi(x) - \xi^*) e^{-\beta U(x)} \mathrm{d}x$$

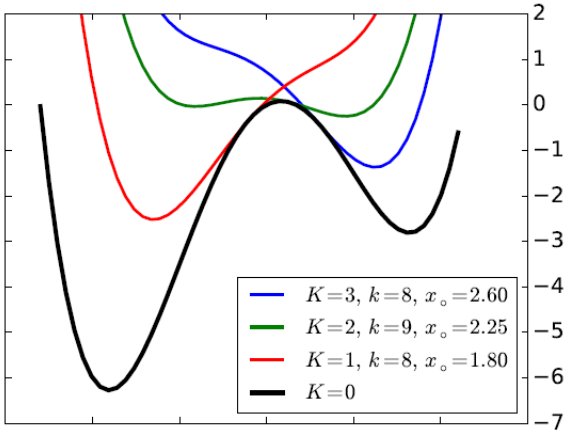


order parameter or reaction coordinate



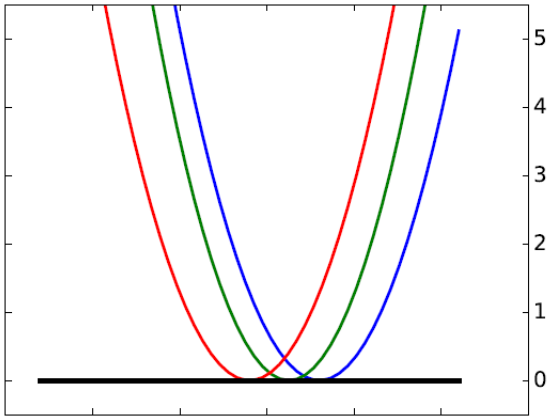
Umbrella sampling

- The ideal bias energy would be $k_B T \log P(\xi)$
- Problem: computing $P(\xi)$ requires sampling from the unbiased distribution!
- Instead of simulating with the ideal bias $k_B T \log P(\xi)$, we select a sub-optimal but *flexible* form of the bias. → umbrella sampling
- Use K different bias potentials that *jointly* allow uniform sampling.



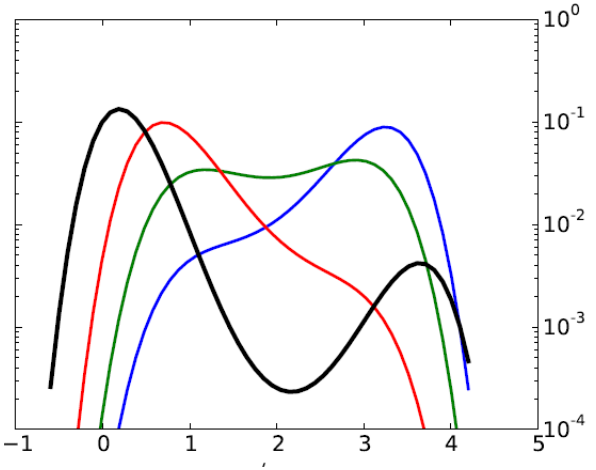
biased potentials

$U_0(x) + U_{\text{bias}}^{(k)}(x)$



bias potentials

$U_{\text{bias}}^{(k)}(x)$

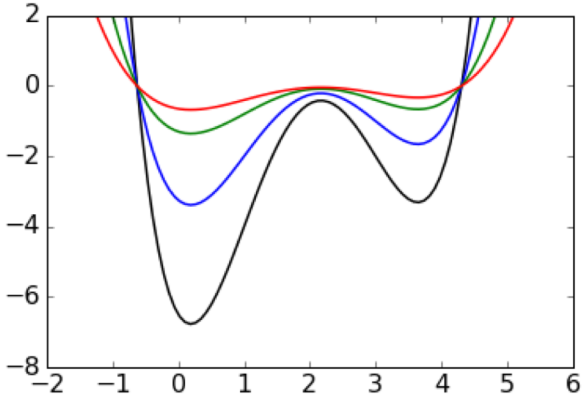


probability distributions

$P_{\text{biased}}(x) \propto e^{-\beta[U^{(0)}(x)+U_{\text{bias}}^{(k)}(x)]}$

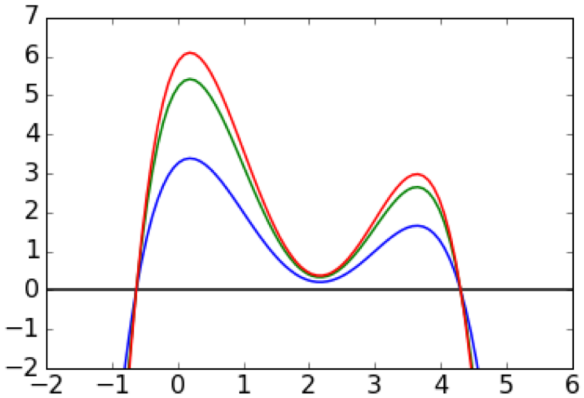
Multi temperature simulation

- Multi-temperature simulations is another way of approximately producing a flat biased distribution.



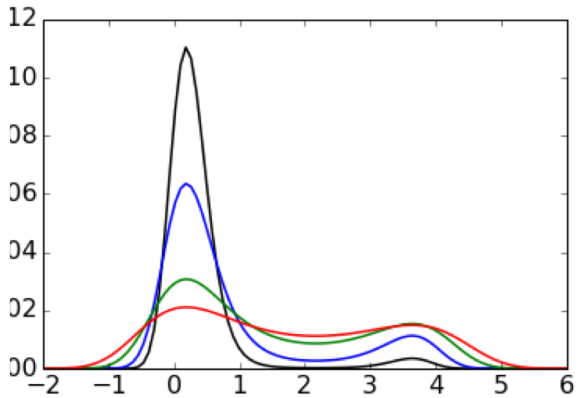
biased potentials

$\frac{\beta^{(k)}}{\beta^{(0)}} U^{(0)}(x)$



"bias potentials"

$\frac{\beta^{(k)}-\beta^{(0)}}{\beta^{(0)}} U^{(0)}(x)$



probability distributions

$P_{\text{biased}}(x) \propto e^{-\beta^{(k)} U^{(0)}(x)}$

- Idea has to taken with a grain of salt: order parameter and the minima that it connects are assumed to stay the same for all temperatures.

A bit of notation…

- Introduce “dimension-less bias“

$$b^{(k)}(x) \equiv \beta^{(k)} U^{(k)}(x) - \beta^{(0)} U^{(0)}(x)$$

  by picking the ensemble 0 as the reference ensemble.
- Assume that the energies in the reference ensemble are shifted, such that its Boltzmann distribution is normalized $\beta^{(0)} F^{(0)} = 0$.
- Introduce the log partition function $f^{(k)} = \beta^{(k)} F^{(k)}$

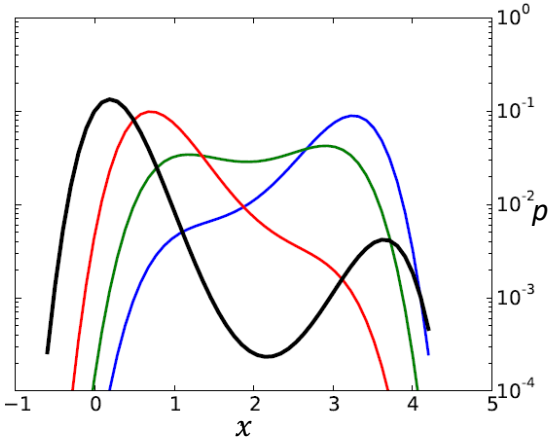
Then the reweighting factors become

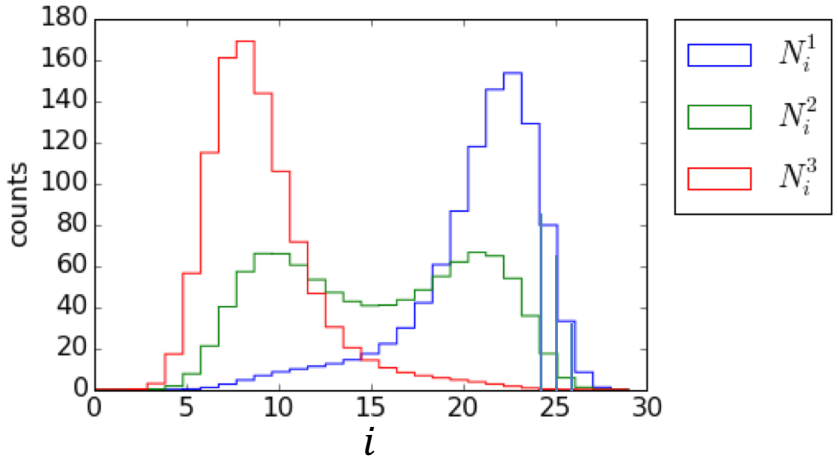
$$e^{-\beta^{(k)} U^{(k)}(x) + \beta^{(0)} U^{(0)}(x) + \beta^{(k)} F^{(k)} - \beta^{(0)} F^{(0)}} = e^{-b^{(k)} + f^{(k)}}$$

WHAM

Weighted **H**istogram **A**nalysis **M**ethod

The MD simulation gives us *realizations* or *samples*. How do we find *probabilities*?





Discretize the order parameter into a number of bins.

For every individual bin, we can do Boltzmann reweighting between ensembles.

$$\pi_i^{(k)} = \frac{\pi_i \exp[-b^{(k)}(i)]}{Z^{(k)}} \qquad Z^{(k)} = \sum_i \pi_i \exp[-b^{(k)}(i)]$$

where we have assumed that bias energy is constant over each bin.

But how to we find π_i?

Optimize likelihood: $L_{\mathrm{WHAM}}(\pi_i^{(k)}) = \prod_k \prod_i \left(\pi_i^{(k)}\right)^{N_i^{(k)}}$ (see next slide)

Maximum likelihood estimation

Start from basic definition of conditional probability:

$$Pr(\text{data}, \text{model}) = Pr(\text{ data} \mid \text{model}) \cdot Pr(\text{model})$$

$$= Pr(\text{model} \mid \text{data}) \cdot Pr(\text{data})$$

$$\max_{\text{models}} \; Pr(\text{ model} \mid \text{data}) = Pr(\text{data} \mid \text{model}) \frac{Pr(\text{model})}{Pr(\text{data})}$$

posterior

likelihood *L*

prior

Because we don't know better: $Pr(\text{model}) = const$

Compute:

$$\max_{\text{models}} \; Pr(\text{data} \mid \text{model})$$

Likelihood for WHAM

Likelihood: $$L_{\mathrm{WHAM}} = \prod_k \prod_i \left(\pi_i^{(k)}\right)^{N_i^{(k)}}$$

Example: set of samples $\{1,2,3,3,2\}$ form simulation with umbrella 1

$$Pr(\{1,2,3,3,2\}) = \pi_1^{(1)}\pi_2^{(1)}\pi_3^{(1)}\pi_3^{(1)}\pi_2^{(1)} = \left(\pi_1^{(1)}\right)^1\left(\pi_2^{(1)}\right)^2\left(\pi_3^{(1)}\right)^2$$

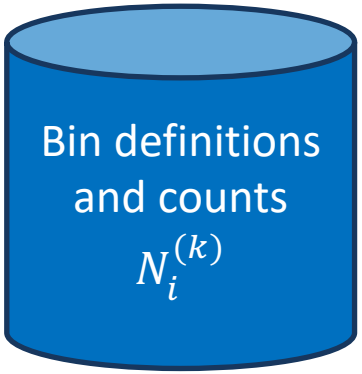
All simulations and all samples are statistically independent.
Inserting the Boltzmann reweighting relation into L_{WHAM} gives:

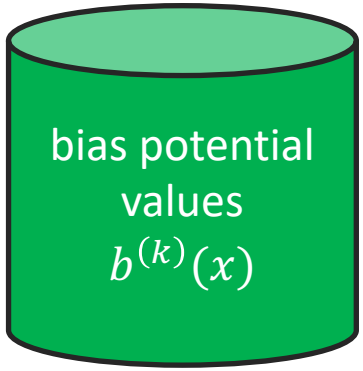
$$L(\pi_1, \dots, \pi_n) = \prod_k \prod_i \left(\frac{\pi_i \ \exp[-b^{(k)}(i)]}{\sum_j \pi_j \exp[-b^{(k)}(j)]}\right)^{N_i^{(k)}}$$

with the data $N_i^{(k)}$, $\exp[-b^{(k)}(i)]$ and the model parameters π_i.

Note: can make bins so small s. t. they contain only one x. $i \longrightarrow x$.

WHAM workflow





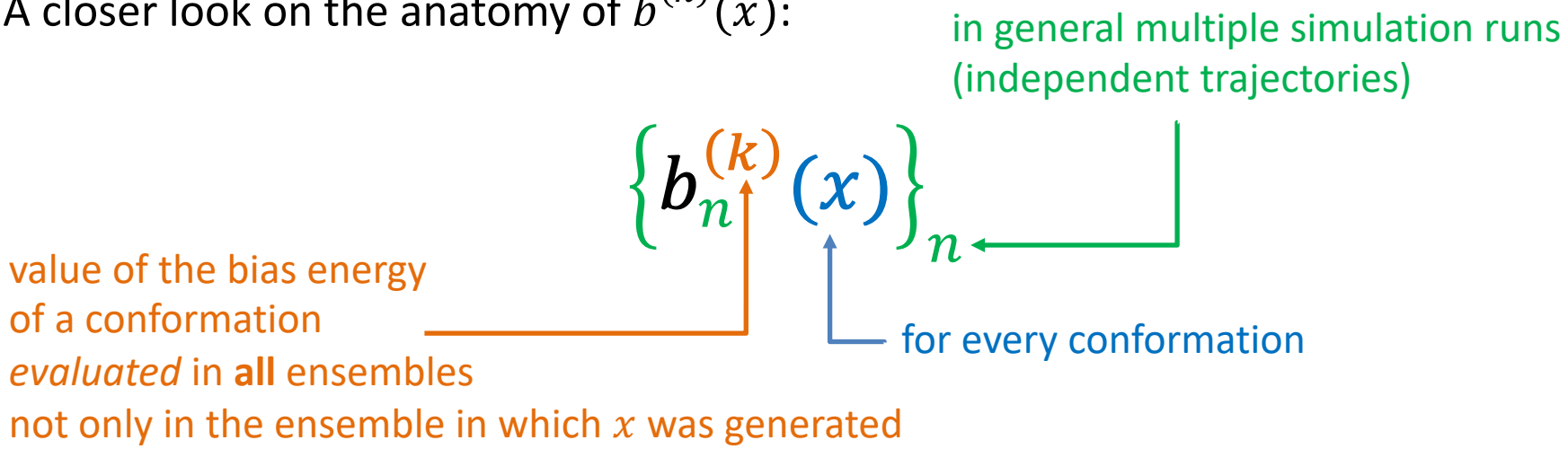
probabilistic model: $$L = \prod_k \prod_i \left(\frac{\pi_i \ \exp[-b^{(k)}(i)]}{\sum_j \pi_j \exp[-b^{(k)}(j)]} \right)^{N_i^{(k)}}$$

optimize model parameters π

stationary probabilities
(thermodynamics)

Computing the bias energies

A closer look on the anatomy of $b^{(k)}(x)$:



- This is 3-D data structure.
- Since the trajectories might have different lengths this is a jagged/ragged array and not a tensor. In PyEmma it's a list of 2-D numpy arrays:

```
btrajs = [
    np.array([[0.0, ...], [1.2, ...]]),
    np.array([[0.0, ...], [4.2, ...]]),
    ...
]
```

Computing the bias energies

Example: Umbrella sampling

- All temperatures are the same

$$\beta^{(k)} = \beta = 1/k_B T = 1/(0.00198\ \mathrm{kcal/mol\ K} \cdot 300\ K)$$

- The bias is a quadratic function of an order parameter $\xi(x)$

$$U^{(k)}(x) = \frac{1}{2}\kappa^{(k)}\left(\xi(x) - \xi^{(k)}_{\mathrm{center}}\right)^2$$

with the spring constants $\kappa^{(k)}$ and rest positions $\xi^{(k)}_{\mathrm{center}}$.

```
btrajs = []
for n in range(N_trajectories):
    b = np.zeros((N_frames[n], N_ensembles))
    for i in range(N_frames[n]):
        xi = compute_order_parameter(md_traj[n][i, :])
        for k in range(N_ensembles):
            b[i, k] = 0.5*kappa[k]*(xi-center[k])**2
    btrajs.append(b)
```

Computing the bias energies

Working with saved (pre-computed) order parameters:

```
btrajs = []
for n in range(N_trajectories):
    b = np.zeros((N_frames[n], N_ensembles))
    order_parameter = np.loadtxt('order_parameter_simulation_%d.dat'%n)
    for i in range(N_frames[n]):
        xi = order_parameter[i]
        for k in range(N_ensembles):
            b[i, k] = 0.5*kappa[k]*(xi-center[k])**2
    btrajs.append(b)
```

NOT computing the bias energies

`pyemma.thermo.estimate_umbrella_sampling(`*us_trajs, us_dtrajs, us_centers, us_force_constants, md_trajs=None, md_dtrajs=None, kT=None, maxiter=10000, maxerr=1e-15, save_convergence_info=0, estimator='wham', lag=1, dt_traj='1 step', init=None, init_maxiter=10000, init_maxerr=1e-08, width=None, **kwargs*)

This function acts as a wrapper for `tram()`, `dtram()`, `mbar()`, and `wham()` and handles the calculation of bias energies (`bias`) and thermodynamic state trajectories (`ttrajs`) when the data comes from umbrella sampling and (optional) unbiased simulations.

Parameters:

- **us_trajs** (*list of N arrays, each of shape (T_i, d)*) – List of arrays, each having T_i rows, one for each time step, and d columns where d is the dimensionality of the subspace in which umbrella sampling was applied. Often d=1, and thus us_trajs will be a list of 1d-arrays.
- **us_dtrajs** (*list of N int arrays, each of shape (T_i,)*) – The integers are indexes in 0, ...,n-1 enumerating the n discrete states or the bins the umbrella sampling trajectory is in at any time.
- **us_centers** (*list of N floats or d-dimensional arrays of floats*) – List or array of N center positions. Each position must be a d-dimensional vector. For 1d umbrella sampling, one can simply pass a list of centers, e.g. [-5.0, -4.0, -3.0, ...].
- **us_force_constants** (*list of N floats or d- or dxd-dimensional arrays of floats*) – The force constants used in the umbrellas, unit-less (e.g. kT per squared length unit). For multidimensional umbrella sampling, the force matrix must be used.

order parameter trajectories

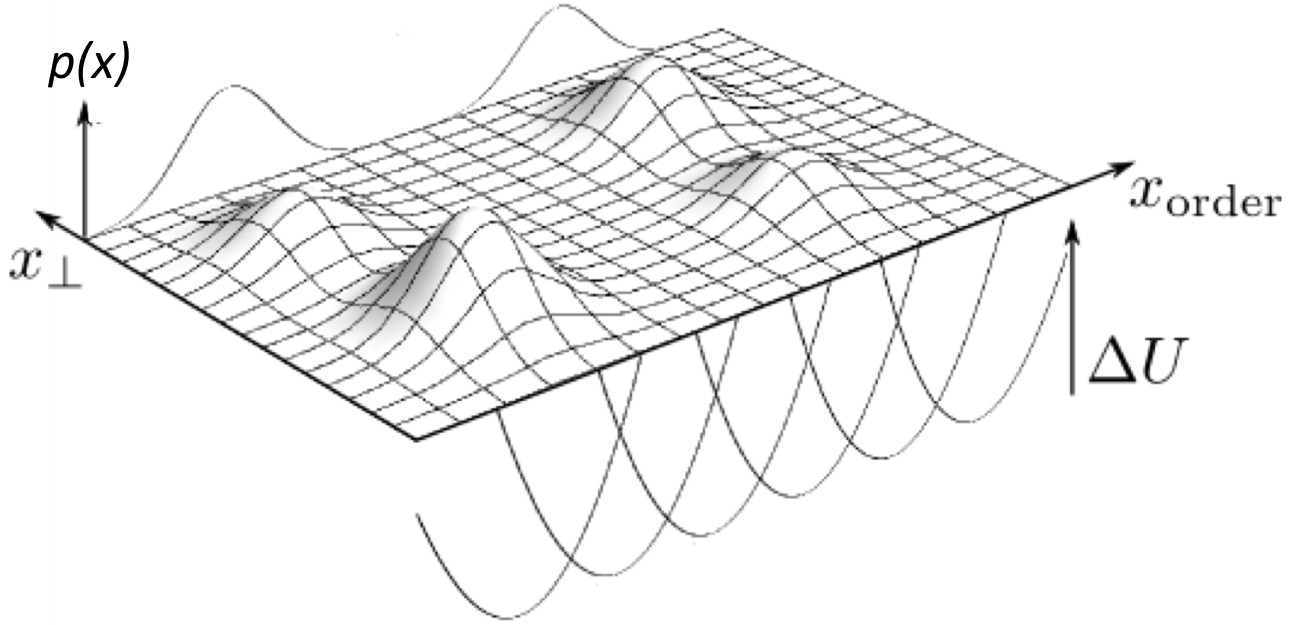
- Pyemma example

Combining free energy calculations with MSMs: Multi Ensemble Markov Models





Problems of Umbrella sampling: slow orthogonal degrees of freedom

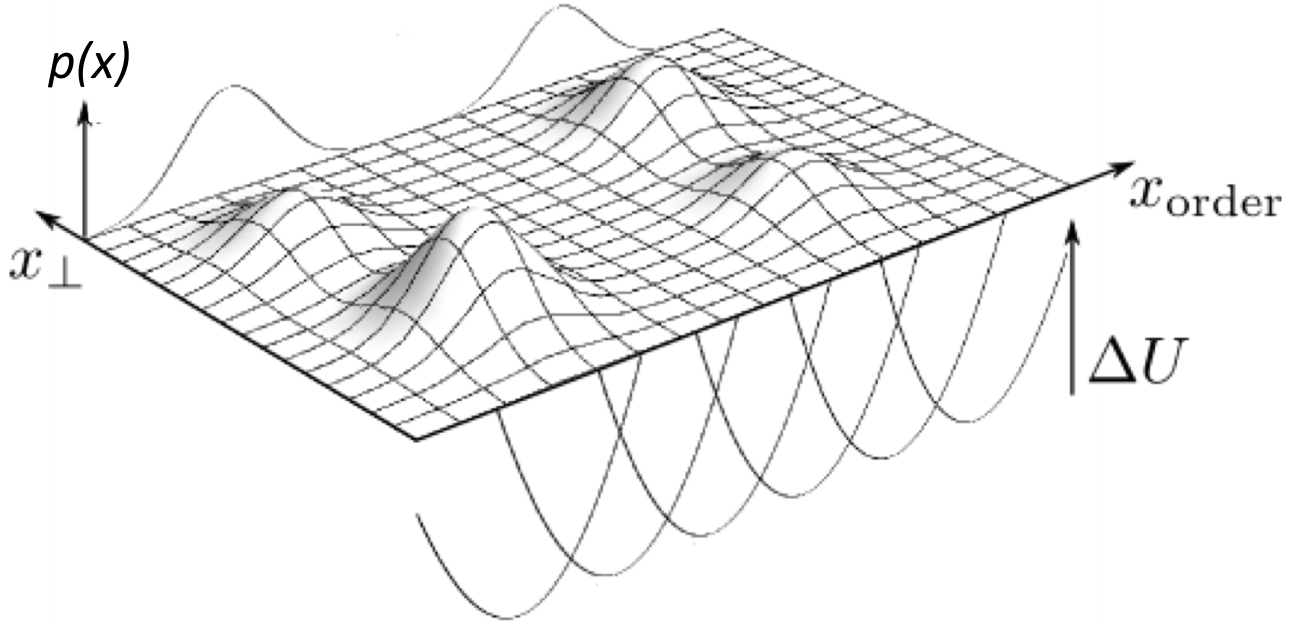


Remember the WHAM likelihood:

$$L_{\text{WHAM}} = \prod_k \prod_i \left(\pi_i^{(k)}\right)^{N_i^{(k)}}$$

Second product means that samples are drawn from the equilibrium distribution $\pi_i^{(k)}$.

Problems of Umbrella sampling: slow orthogonal degrees of freedom



In the energy landscape above, motion along $x_\perp$ can be highly autocorrelated. So the assumption of independent samples may be wrong. → systematic error

Since we know that MSMs can be used to compute free energies reliably form correlated data, can't we just somehow build an MSM along $x_\perp$?

MEMM

Multi **E**nsemble **M**arkov **M**odel $T_{ij}^{(k)}$

$$\begin{pmatrix} T_{11}^{(1)} & \cdots & T_{1n}^{(1)} \\ \vdots & \ddots & \vdots \\ T_{n1}^{(1)} & \cdots & T_{11}^{(1)} \end{pmatrix}$$

$$\begin{pmatrix} T_{11}^{(2)} & \cdots & T_{1n}^{(2)} \\ \vdots & \ddots & \vdots \\ T_{n1}^{(2)} & \cdots & T_{11}^{(2)} \end{pmatrix}$$

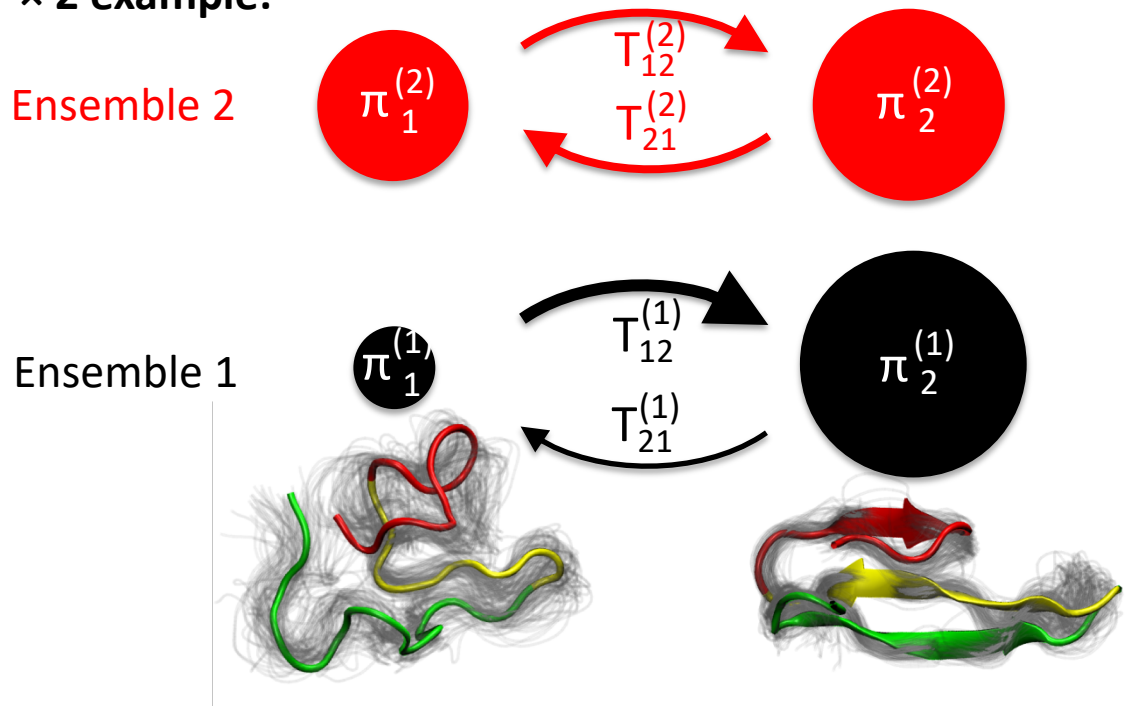
$$\vdots$$

$$\begin{pmatrix} T_{11}^{(K)} & \cdots & T_{1n}^{(K)} \\ \vdots & \ddots & \vdots \\ T_{n1}^{(K)} & \cdots & T_{11}^{(K)} \end{pmatrix}$$

index *k* = number of the Umbrella potential
= number of temperature in multi-temperature simulations

indices *i,j* = number of the discrete Markov state,
i.e. bin number along $x_\perp$
or any other sensible state discretization

2 × 2 example:



MEMM

Multi **E**nsemble **M**arkov **M**odel $T_{ij}^{(k)}$

- **How the individual MSMs in the MEMM are coupled together?**

- Part 1 of the answer:
  Boltzmann reweighting of stationary distributions (like in WHAM)

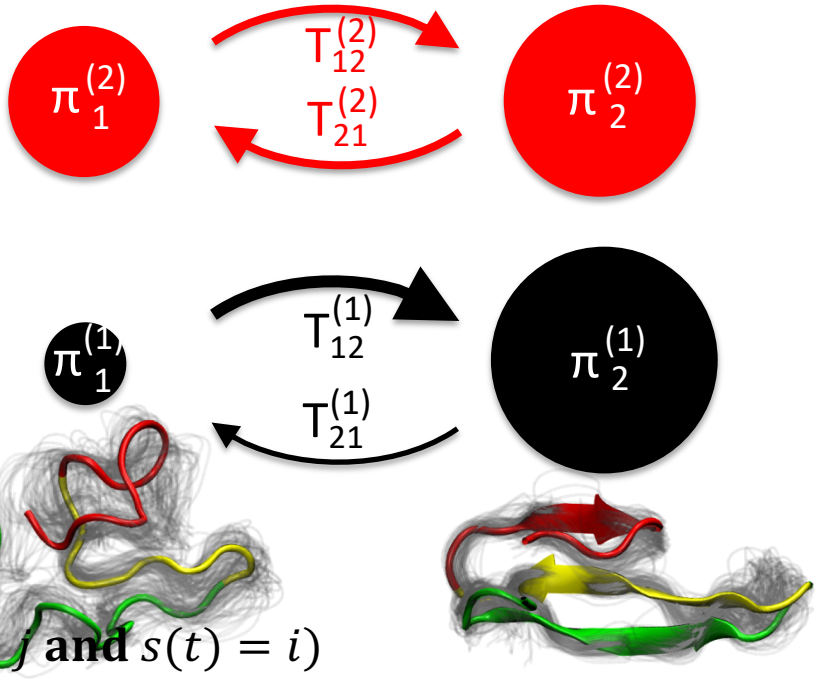
$$\pi_i^{(k)} = \frac{\pi_i \exp[-b^{(k)}(i)]}{Z^{(k)}} \qquad Z^{(k)} = \sum_i \pi_i \exp[-b^{(k)}(i)]$$

- Part 2 of the answer:
$\pi_i^{(k)}$ is the stationary distribution of $T_{ij}^{(k)}$.
We even require a stronger condition that $\mathbf{T}^{(k)}$ is reversible with respect to $\boldsymbol{\pi}^{(k)}$.

$$\pi_i^{(k)} T_{ij}^{(k)} = \pi_j^{(k)} T_{ji}^{(k)}$$

reversibility = detailed balance

$$\Pr(s(t+\tau) = i \textbf{ and } s(t) = j) = \Pr(s(t+\tau) = j \textbf{ and } s(t) = i)$$



(d)TRAM

(**d**iscrete) **T**ransition-based **R**eweighting **A**nalysis **M**ethod

- **How is the MEMM estimated?**
- **Reminder - estimation of MEMs:**

Likelihood for an MSM: $L_{\mathrm{MSM}} = \prod_i \prod_j (T_{ij})^{C_{ij}}$

Consider example trajectory $(1 \to 2 \to 2 \to 1 \to 2)$

$$\Pr(1 \to 2 \to 2 \to 1 \to 2) = \Pr(1) \cdot T_{12} \cdot T_{22} \cdot T_{21} \cdot T_{12}$$

$$\propto (T_{11})^0 (T_{12})^2 (T_{22})^1 (T_{21})^1$$

$$= (T_{11})^{C_{11}} (T_{12})^{C_{12}} (T_{22})^{C_{22}} (T_{21})^{C_{21}}$$

$$= \prod_{i=1}^{2} \prod_{j=1}^{2} (T_{ij})^{C_{ij}}$$

(d)TRAM

(**d**iscrete) **T**ransition-based **R**eweighting **A**nalysis **M**ethod

- **How is the MEMM estimated?**

Basically **an MEMM is just a collection of MSMs**.

$$L_{\mathrm{MEMM}}(\mathbf{T}^{(1)}, \ldots, \mathbf{T}^{(K)}) = \prod_k L_{\mathrm{MSM}}(\mathbf{T}^{(k)})$$

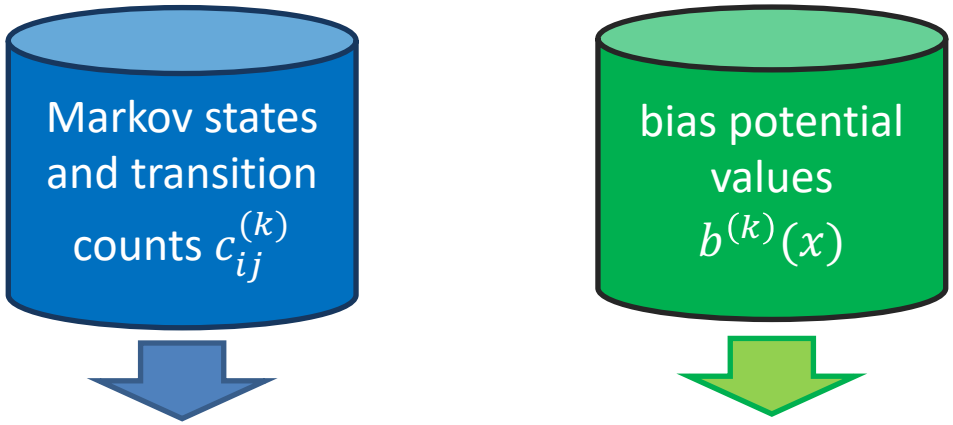
Inserting gives:

$$L_{\mathrm{MEMM}} = \prod_k \prod_i \prod_j \left(T_{ij}^{(k)}\right)^{C_{ij}^{(k)}}$$

Maximize L_{MEMM} under the constraints:

- $\pi_i^{(k)} T_{ij}^{(k)} = \pi_j^{(k)} T_{ji}^{(k)}$
- $\sum_j T_{ij}^{(k)} = 1$
- $\pi_i^{(k)} = \frac{\pi_i \exp[-b^{(k)}(i)]}{\sum_j \pi_j \exp[-b^{(k)}(j)]}$
- $T_{ij}^{(k)} \geq 0$

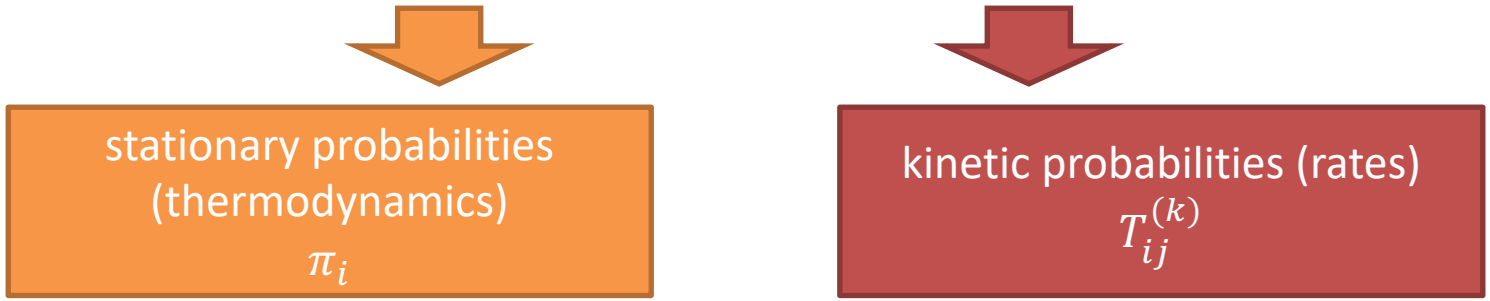
(d)TRAM: workflow



probabilistic model: $$L = \prod_k \prod_i \prod_j \left(T_{ij}^{(k)}\right)^{c_{ij}^{(k)}}$$

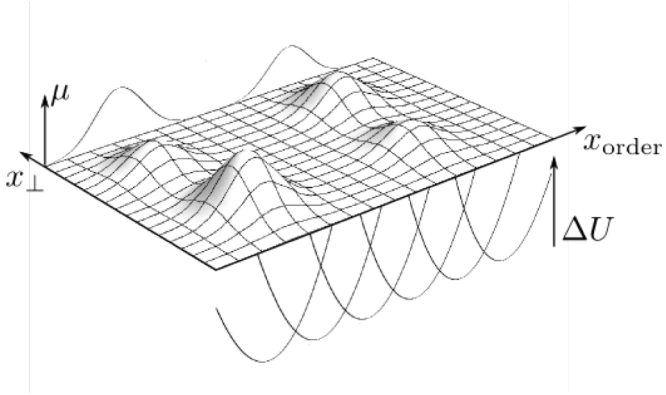
$$\pi_i \ \exp[-b^{(k)}(i)]\, T_{ij}^{(k)} = \pi_j \ \exp[-b^{(k)}(j)]\, T_{ji}^{(k)}$$

Optimize model parameters π and T.



Advantages of using (d)TRAM

- Better estimation of free energies along the unbiased orthogonal degrees of freedom.

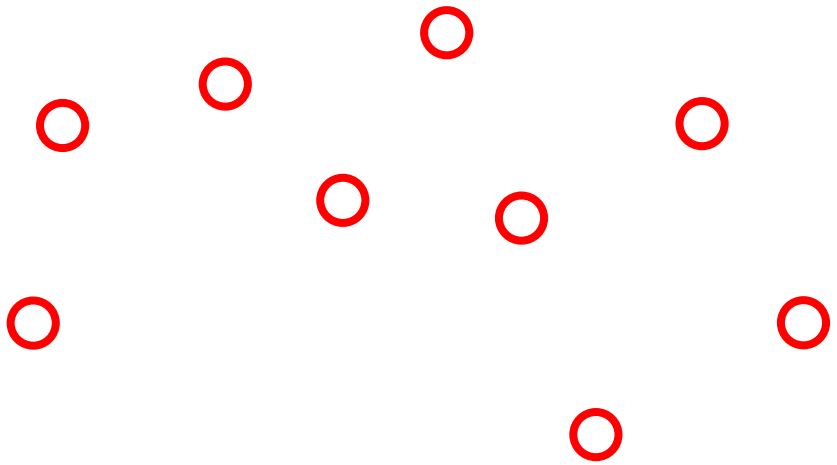


- There is no initial equilibration transient where the simulation have to relax to global equilibrium.

- Smaller de-correlation time (simulation time until one gets a new uncorrelated frame). More efficient usage of the data.

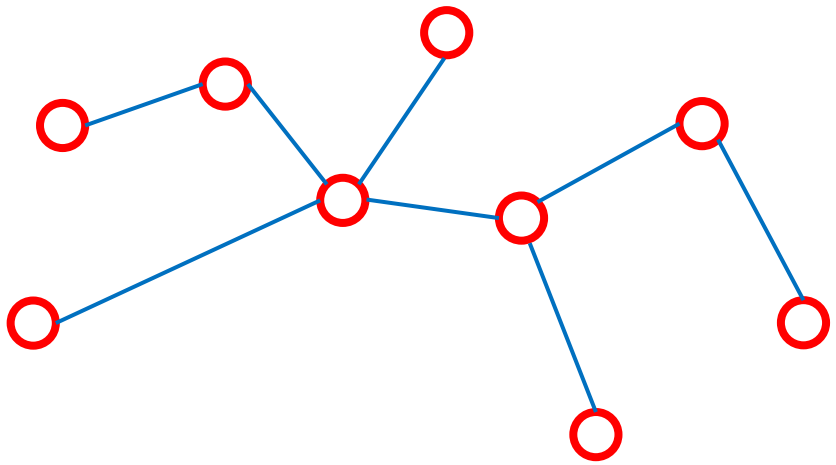
- Kinetics and free energies are inseparably related in reversible systems.
- Make use of detailed balance relation $\exp[-\beta f_i]\, T_{ij} = \exp\left[-\beta f_j\right] T_{ji}$

1. Define the Markov states.



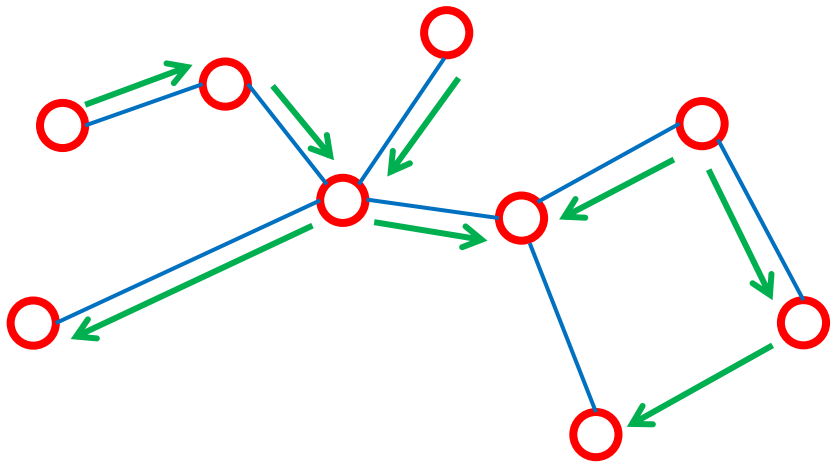
- Kinetics and free energies are inseparably related in reversible systems.
- Make use of detailed balance relation $\exp[-\beta f_i]\, T_{ij} = \exp\left[-\beta f_j\right] T_{ji}$

1. Define the Markov states.
2. Biased simulation provides information about the ΔF's between the states.



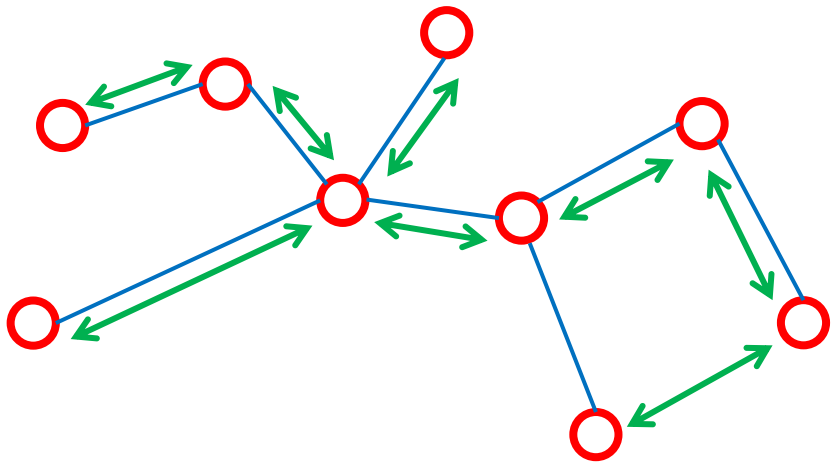
- Kinetics and free energies are inseparably related in reversible systems.
- Make use of detailed balance relation $\exp[-\beta f_i]\, T_{ij} = \exp\left[-\beta f_j\right] T_{ji}$

1. Define the Markov states.
2. Biased simulation provides information about the ΔF's between the states
3. Unbiased simulations provide information about the transition probabilities in one direction.



- Kinetics and free energies are inseparably related in reversible systems.
- Make use of detailed balance relation $\exp[-\beta f_i]\, T_{ij} = \exp\left[-\beta f_j\right] T_{ji}$

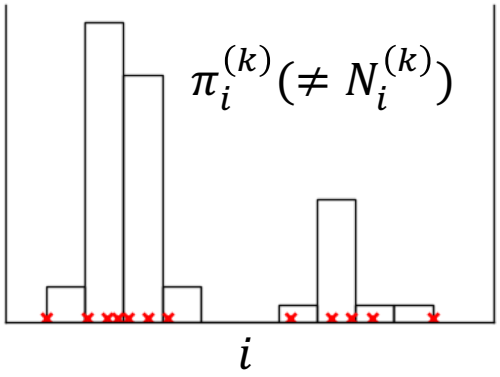
1. Define the Markov states.
2. Biased simulation provides information about the ΔF's between the states.
3. Unbiased simulations provide information about the transition probabilities in one direction.
4. TRAM yields the missing probabilities, **completing the model**.



Bin-less estimators

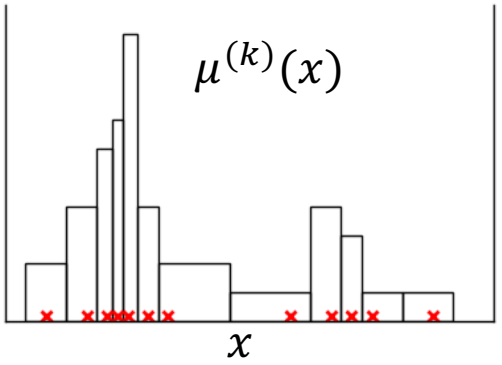
MBAR

Multistate **B**ennet **a**cceptance **r**atio



$$L_{\mathrm{WHAM}} = \prod_k \prod_i \left(\pi_i^{(k)}\right)^{N_i^{(k)}}$$

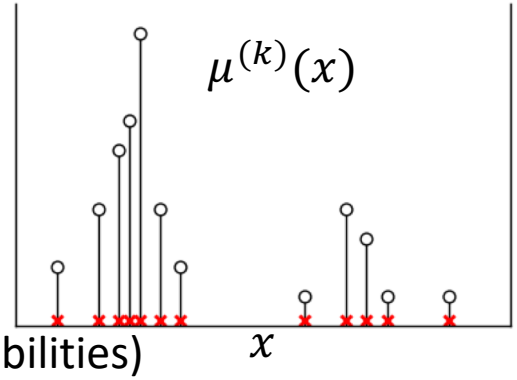
- Width of the bin is never used.
  Can put every sample in its own bin.
  Then $N_i^{(k)}$ is either 1 or 0.
  Ignore all factors of the form $\left(\pi_i^{(k)}\right)^0 = 1$.



$$L_{\mathrm{MBAR}} = \prod_k \prod_x \mu^{(k)}(x)$$

- $\mu^{(k)}(x) = \frac{e^{-b^{(k)}(x)}}{Z^{(k)}} \mu^{(\mathrm{ref})}(x)$

  instead of $\pi_i^{(k)} = \frac{e^{-b^{(k)}(i)}}{Z^{(k)}} \pi_i^{(\mathrm{ref})}$



- WHAM: binning -> reweighting
- MBAR: reweighting -> optional binning (for computing probabilities)

Bin-less TRAM

- How to combine the benefits an MSM with bin-less reweighting?
- For WHAM->MBAR we let go the bin-size to zero.
- For dTRAM->TRAM this doesn‘t work. MSM with a high number of states are hard to handle.
- Introduce the local equilibrium distribution.

The local equilibrium distribution

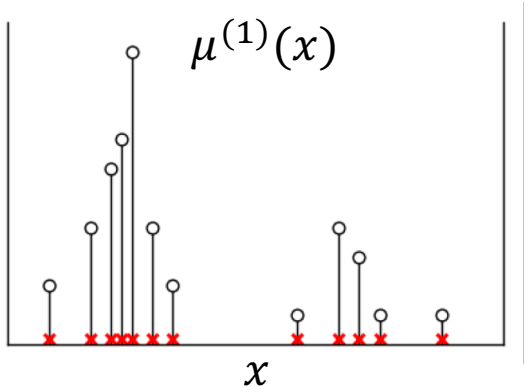
$\mu^{(k)}(x_j)$: contribution of the sample x_j to the Boltzmann distribution of ensemble k.

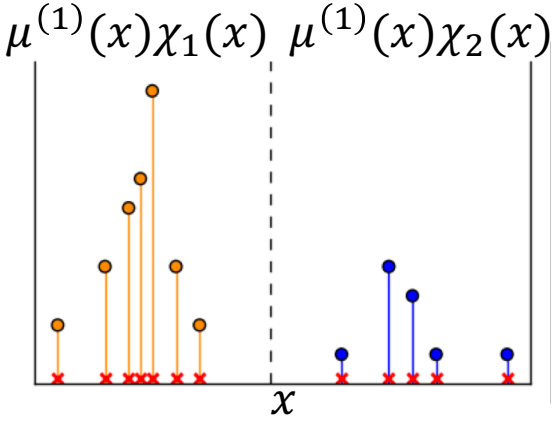
$\mu_i^{(k)}(x_j)$: contribution of the sample x_j to the Boltzmann distribution of ensemble k, given that the sample falls into Markov state s_i.

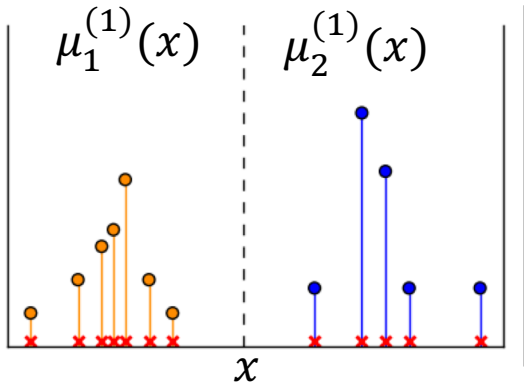
$$\mathbb{P}(\, x \mid \text{state } i \text{ and ens. } k\,) = \frac{\mathbb{P}(x \text{ and } x \in \text{state } i \text{ and ens. } k)}{\mathbb{P}(\text{state } i \text{ and ens. } k)}$$

$$\Longrightarrow \mu_i^{(k)}(x_j) = \frac{\mu^{(k)}(x_j)\chi_i(x_j)}{z_i^{(k)}}$$

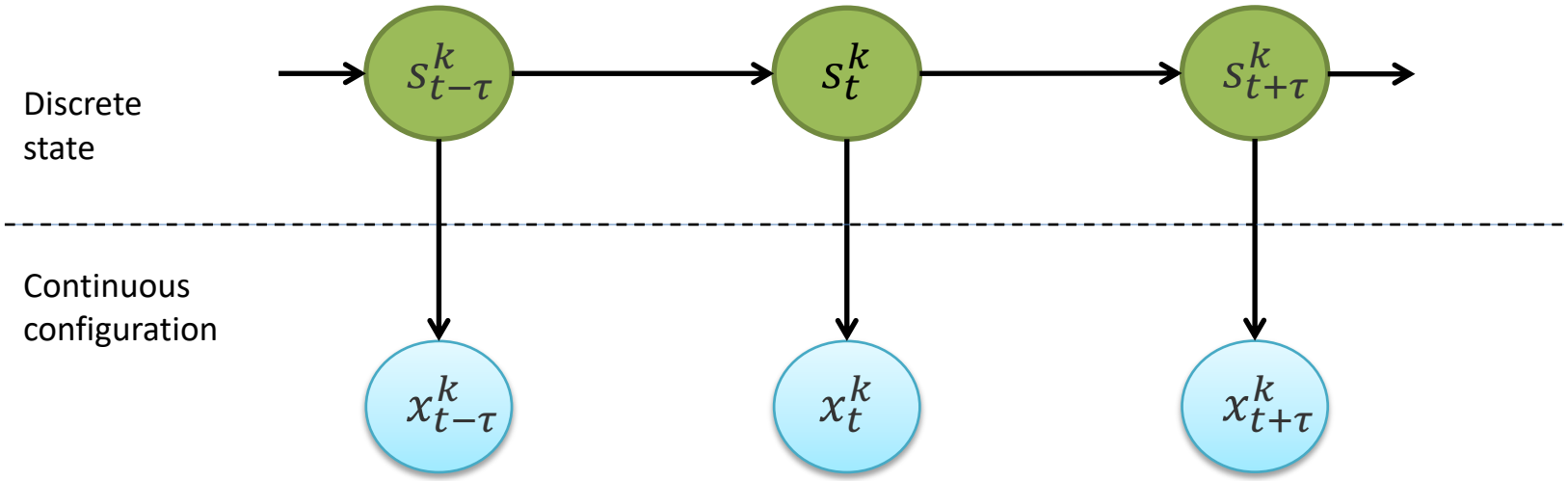
$$= \frac{\mu(x_j)\exp\left[-b^{(k)}(x_j)\right]\chi_i(x_j)}{z_i^{(k)}}$$





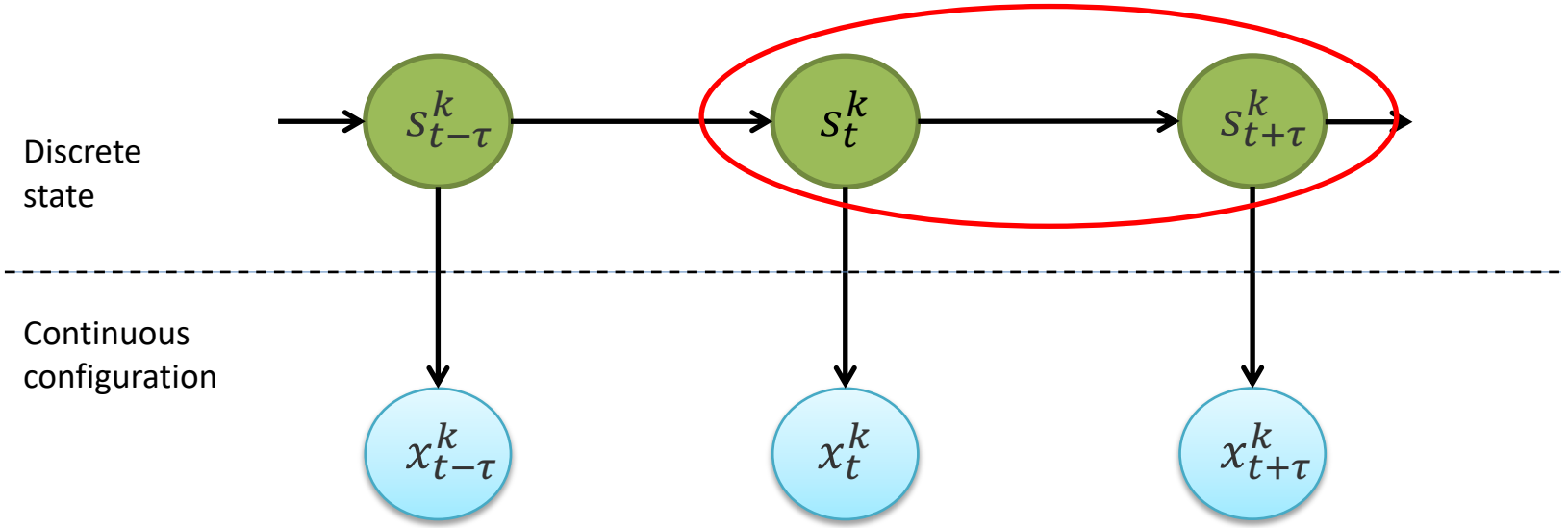


Formulation of the TRAM model



Simulation at ensemble k

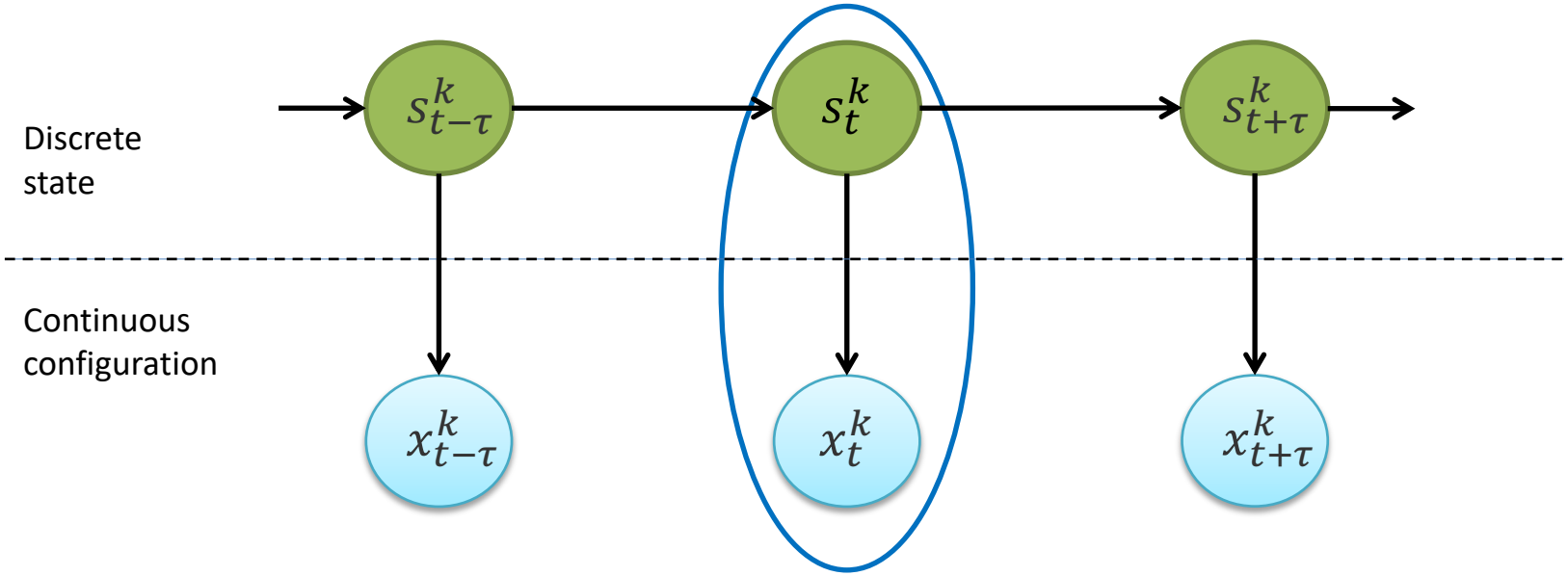
Formulation of the TRAM model



Simulation at ensemble k

$$\mathbb{P}\left(s_{t+\tau}^{(k)} = j \middle| s_t^{(k)} = i\right) = T_{ij}^{(k)}$$ (modeling by MSM)

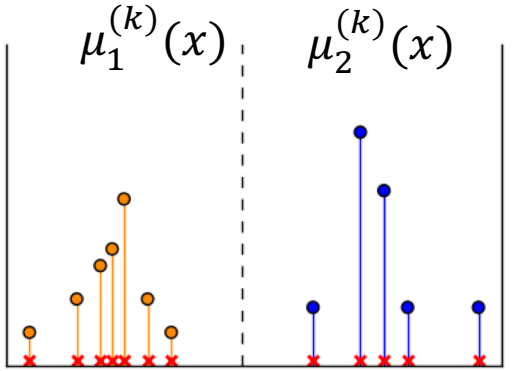
Formulation of the TRAM model



Simulation at ensemble k

$$\mathbb{P}\left(x_t^{(k)} \middle| s_t^{(k)} = i\right) = \mu_i^{(k)}(x_t)$$

(output according to local equilibrium)



Formulation of the TRAM model

Model for one (whole) trajectory:

$$L(\text{traj from ensemble } k) = \mu^{(k)}_{s(0)} \cdot T^{(k)}_{s(0)s(1)} \cdot \mu^{(k)}_{s(1)} \cdots T^{(k)}_{s(N-1)s(N)} \cdot \mu^{(k)}_{s(N)}$$

Rearranging:

$$L(k) = \prod_{i,j} \left(T^{(k)}_{ij}\right)^{c^{(k)}_{ij}} \cdot \prod_{x \in X^k} \mu^{(k)}_{s(x)}$$

Model for all trajectories from all ensembles:

$$L = \prod_{k} L(k)$$

TRAM: workflow

probabilistic model: $$L = \prod_{i,j,k} \left(T_{ij}^{(k)}\right)^{c_{ij}^{(k)}} \cdot \prod_{k} \prod_{x \in X^k} \frac{e^{-b^{(k)}(x)} \mu(x)}{z_{s(x)}^{(k)}}$$

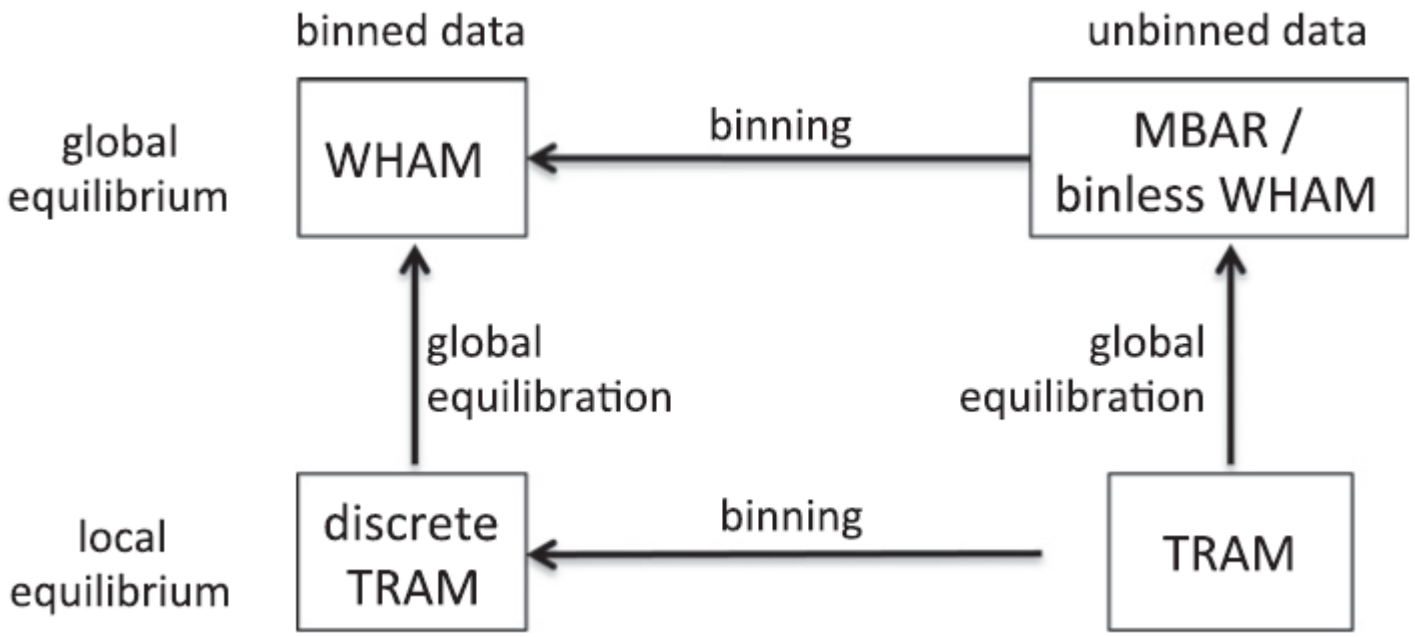
$$z_i^{(k)} T_{ij}^{(k)} = z_j^{(k)} T_{ji}^{(k)}$$

optimize model parameters T and μ (and z[μ])

stationary probabilities (thermodynamics)

kinetic probabilities (rates)

Relation between the methods



Real-world applications

PMI-Mdm2: medically relevant; complex mechanism

- $^{25\text{-}109}$Mdm2: amino acids 25-109 of Mdm2
  - Mdm2 is a natural protein.
  - Mdm2 is overexpressed (produced in increased quantity) in certain cancer types. This leads to pathogenic interaction of Mdm2 with other proteins
- PMI: peptide (12 amino acids) was developed to stop this pathogenic interaction by blocking Mdm2's binding site.
- PMI is unfolded when unbound [2] but folded when bound to Mdm2. [1]
  → We expect to see a process of coupled folding and binding.

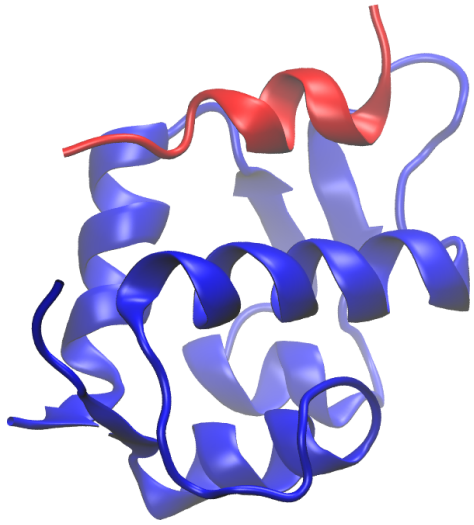
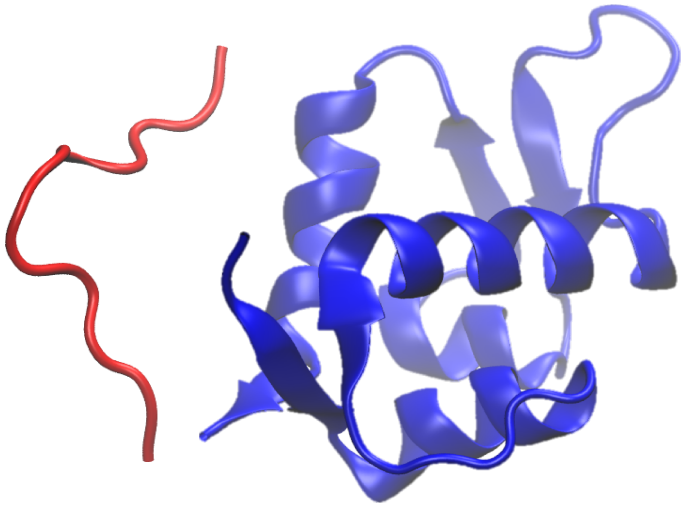


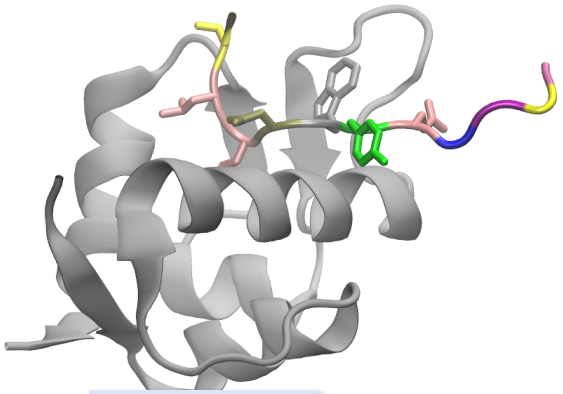
image: X-ray crystal structure from [1]



[1] Pazgier *et al., Proc. Natl. Acad. Sci.* **106**, 4665 (2009)
[2] **Paul** *et al., Nat. Commun.* **8**, 1095 (2017)

PMI-Mmd2: analysis of the **physical data only**

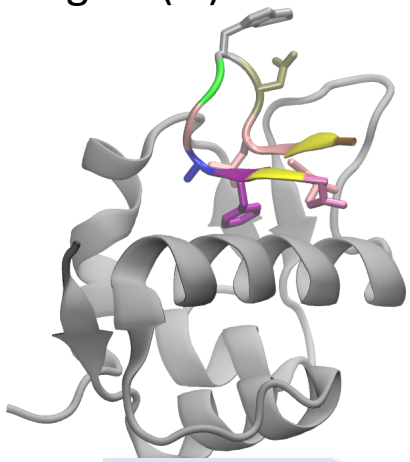
- Not a single full unbinding event is contained in the physical data.
- There are many long-lived states, that appear stable on time scales of 1 to 10 µs.
  - The short (1µs) simulations do not escape these long-lived states.
  - → No exit probabilities and not stationary weights ($\boldsymbol{\pi}$) can be determined for these states.



state 8

T_{8j}=?

π_8=?



state 6

T_{6j}=?

π_6=?

- → No useful MSM could be estimated.

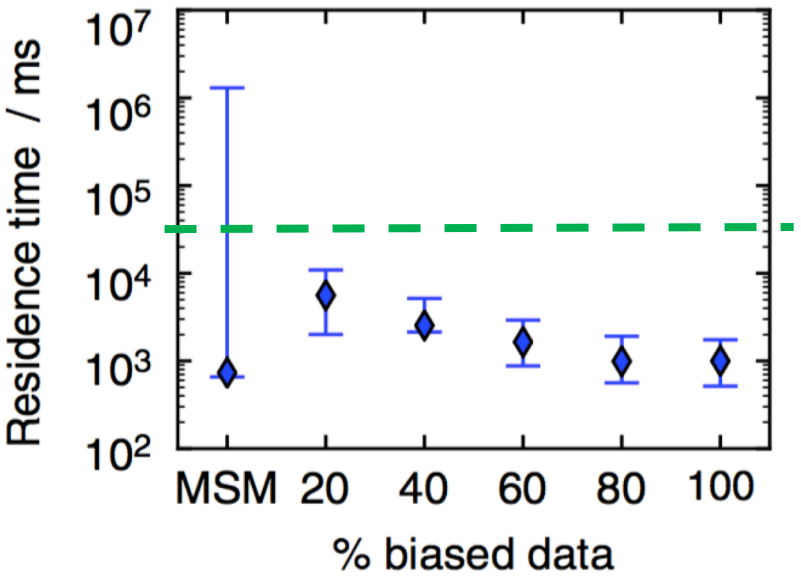
PMI-Mmd2: analysis of **all** simulation data with TRAM

We determine the dissociation constant $K_d = [\mathrm{P}]_{\mathrm{eq}}[\mathrm{L}]_{\mathrm{eq}}/[\mathrm{PL}]_{\mathrm{eq}}$ from

- our simulations using TRAM [3]: $0.34\ \mathrm{nM}$ [0.22 nM, 0.44 nM]
- experiment [3]: $3.02\ \mathrm{nM}$ [2.41 nM, 3.63 nM]

Agrees within the expected "force field" inaccuracies (factor of 10) [1,2].

We determine the residence time $\boldsymbol{k_{\mathrm{off}}^{-1}}$:



experimental value [3]
26.8 s [24.7 s, 34.1 s]

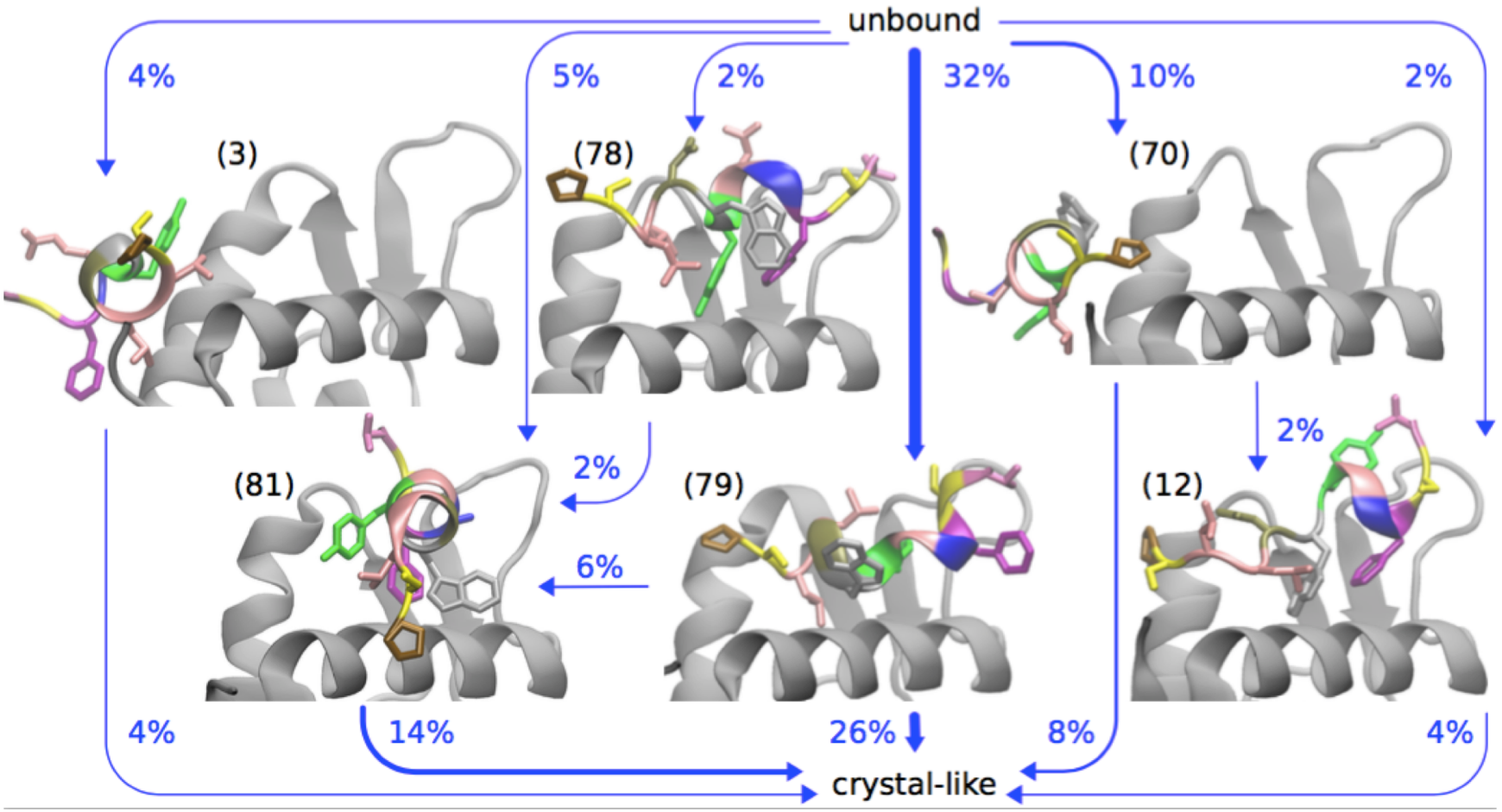
simulation result [3]:
0.88 s [0.48 s, 1.33 s]

Inclusion of biased data drastically **reduces the statistical errors**.

[1] Best *et al., J. Chem. Theory Comput.* **10**, 5113 (2014)
[2] Rauscher *et al., J. Chem. Theory Comput.* **11**, 5513 (2014)
[3] **Paul** ... Abualrous *et al., Nat. Commun,* **8**, 1095 (2017)

errors = 95% confidence intervals

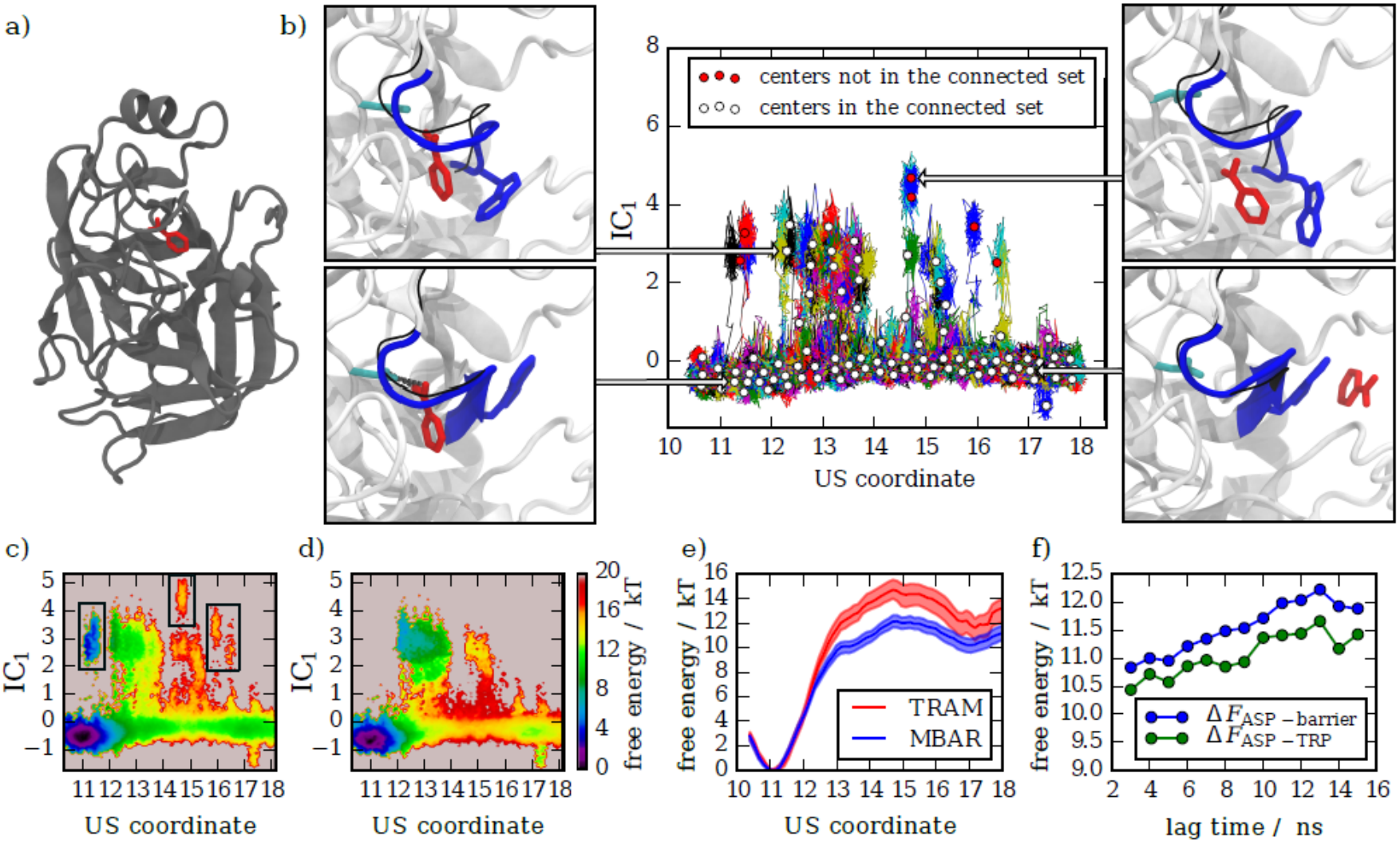
PMI-Mdm2: binding mechanism



Further reading

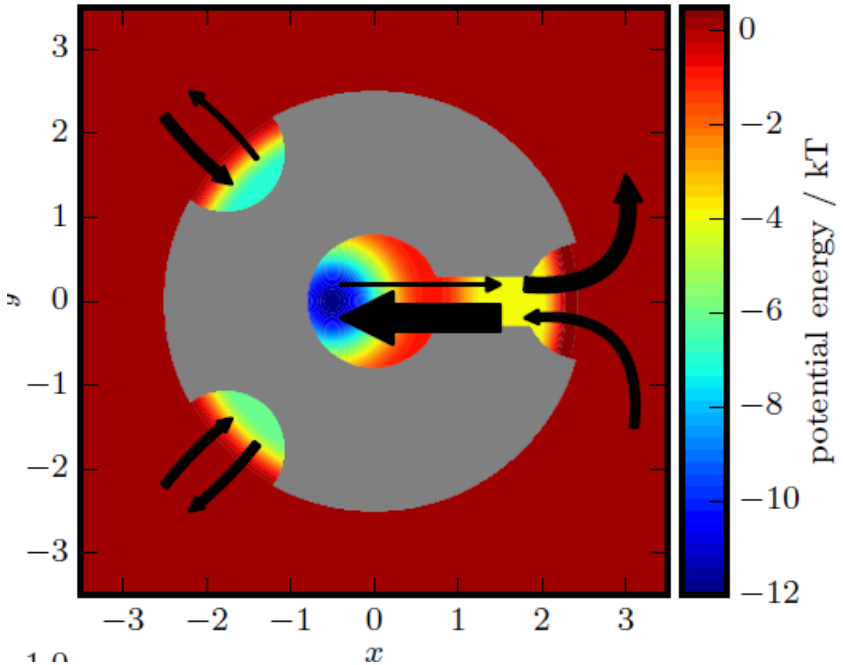
- Wu, Mey, Rosta, Noé: "Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states", *J. Chem. Phys.* **141**, 214106 (2014)
- Wu, Paul, Wehmeyer, Noé: "Multiensemble Markov models of molecular thermodynamics and kinetics", *PNAS* **113**, E3221–E3230 (2016)
- Paul *et al.* "Protein-peptide association kinetics beyond the seconds timescale from atomistic simulations" *Nat. Commun.,* **8**, 1095 (2017)

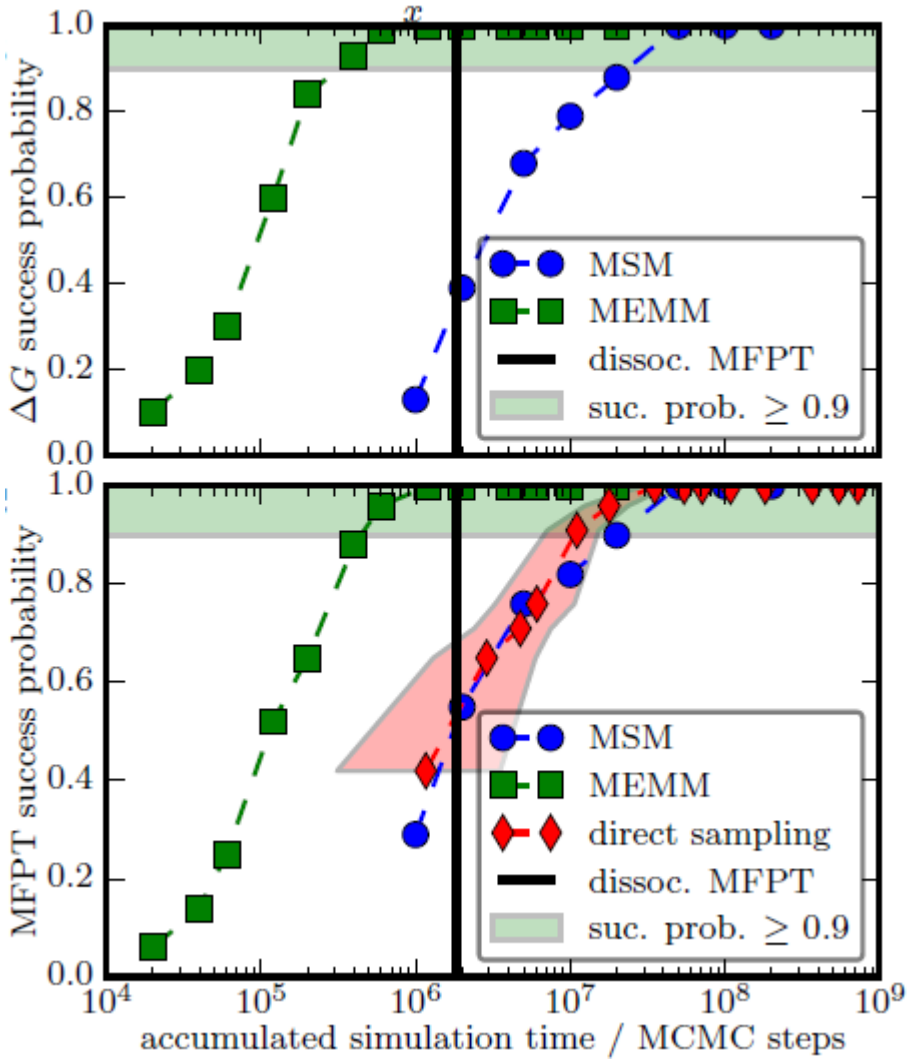
Application 1: Trypsin-Benzamidine



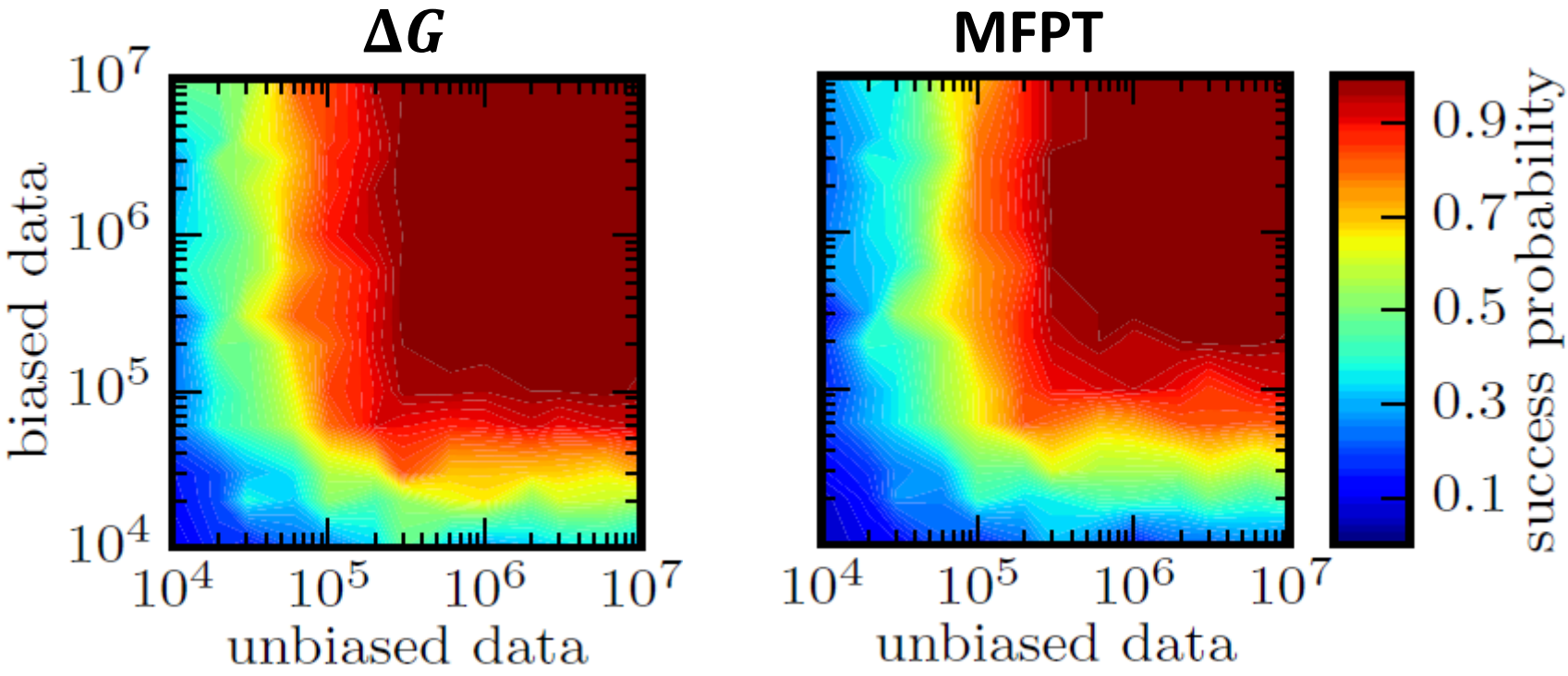
TRAM: strategies for enhanced sampling of kinetics

Model system:

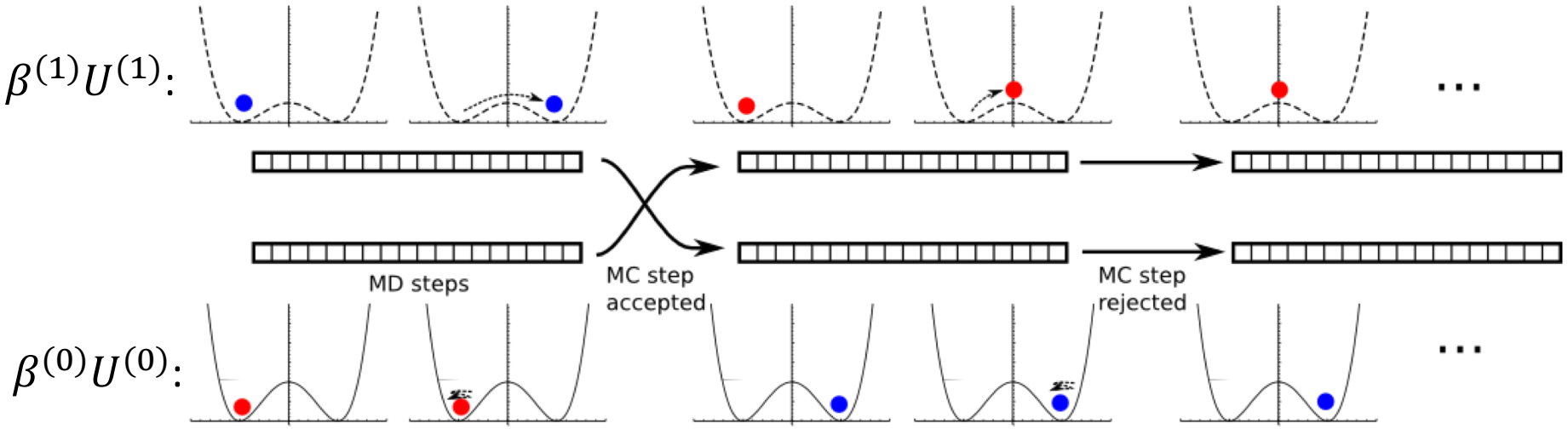




TRAM: strategies for enhanced sampling of kinetics



What is replica exchange?



- sample from generalized ensemble :

$$p(x_i, x_j, \ldots, x_k) = \frac{e^{-\beta^{(0)}U^{(0)}(x_i)}}{Z^{(0)}} \cdot \frac{e^{-\beta^{(1)}U^{(1)}(x_j)}}{Z^{(1)}} \cdot \ldots \cdot \frac{e^{-\beta^{(K)}U^{(K)}(x_K)}}{Z^{(K)}}$$

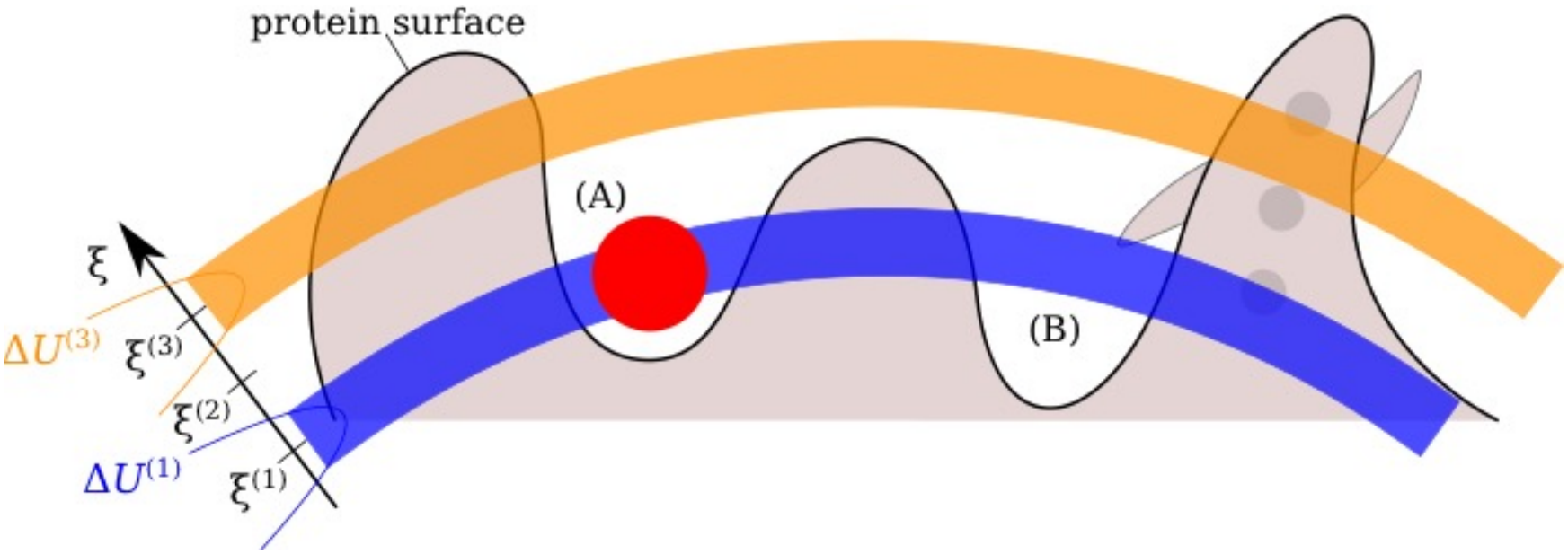
- accept exchanges with Metropolis criterion

$$p_{\text{accept}} = \min\left[1, \frac{e^{-\beta^{(1)}U^{(1)}(x_2)} e^{-\beta^{(2)}U^{(2)}(x_1)}}{e^{-\beta^{(1)}U^{(1)}(x_1)} e^{-\beta^{(2)}U^{(2)}(x_2)}}\right]$$

with labels updated after an accepted exchange.

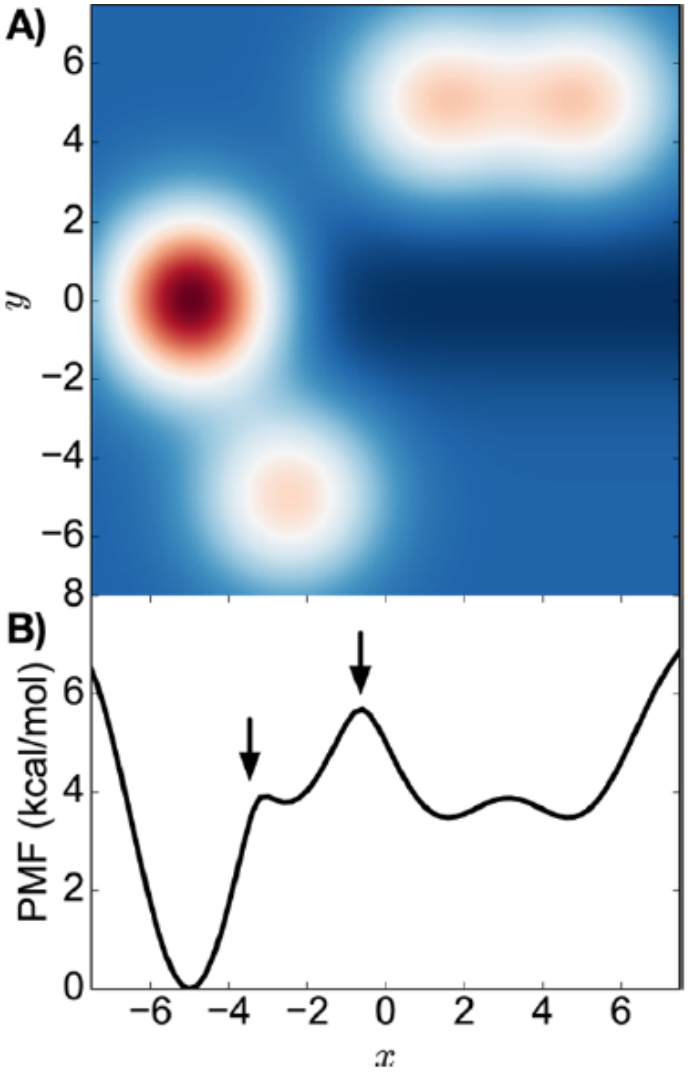
Fukunishi and Watanabe, *J. Chem. Phys*. **116,** 9058 (2002)

The role of HREMD

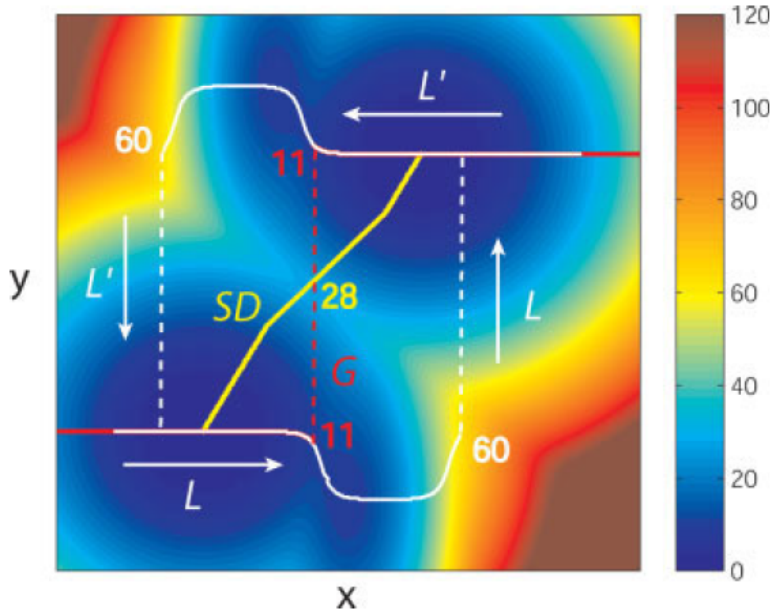


What is valid input data for TRAM?

Overlap in (d)TRAM

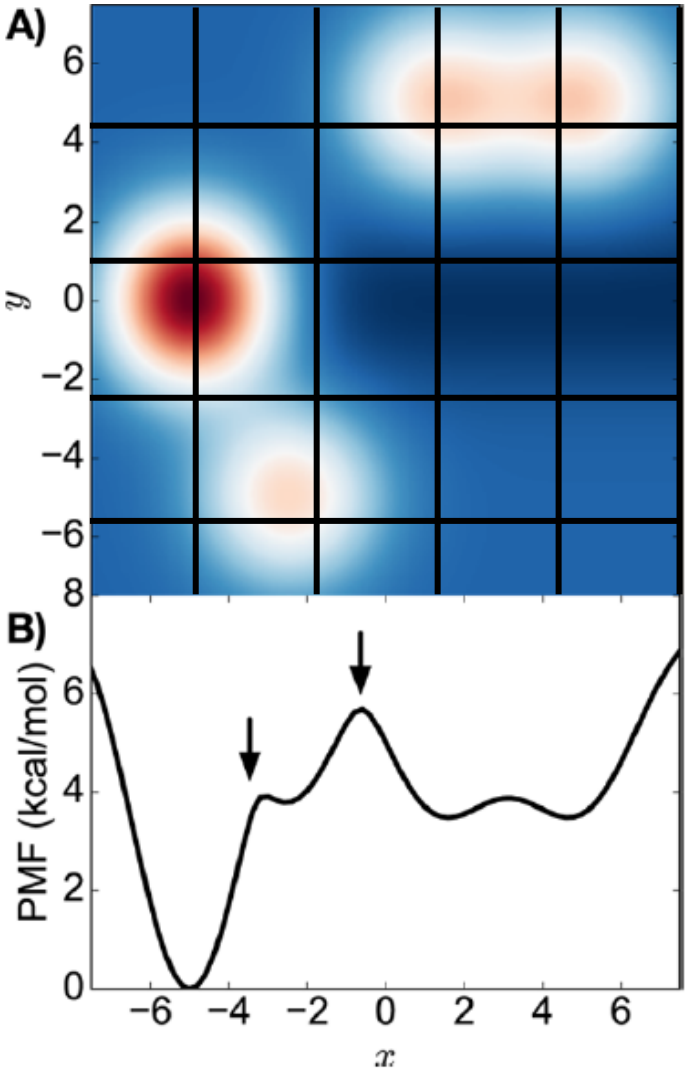


Jo *et al, J. Phys. Chem. B* **120** 8733 (2016)

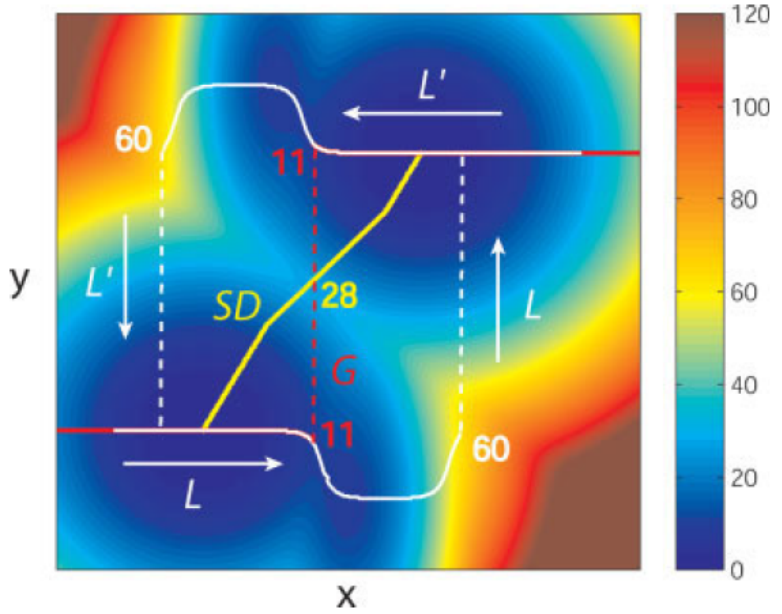


Rosta *et al, J. Comput. Chem.* **30**, 1634 (2009)

Overlap in (d)TRAM



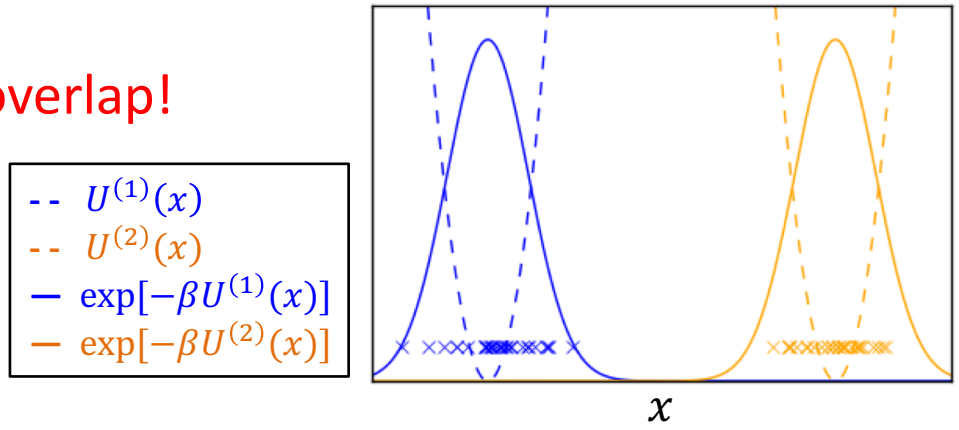
Jo *et al, J. Phys. Chem. B* **120** 8733 (2016)



Rosta *et al, J. Comput. Chem.* **30**, 1634 (2009)

Overlap of biased distributions

Biased distributions have to overlap!



Diagnostics:

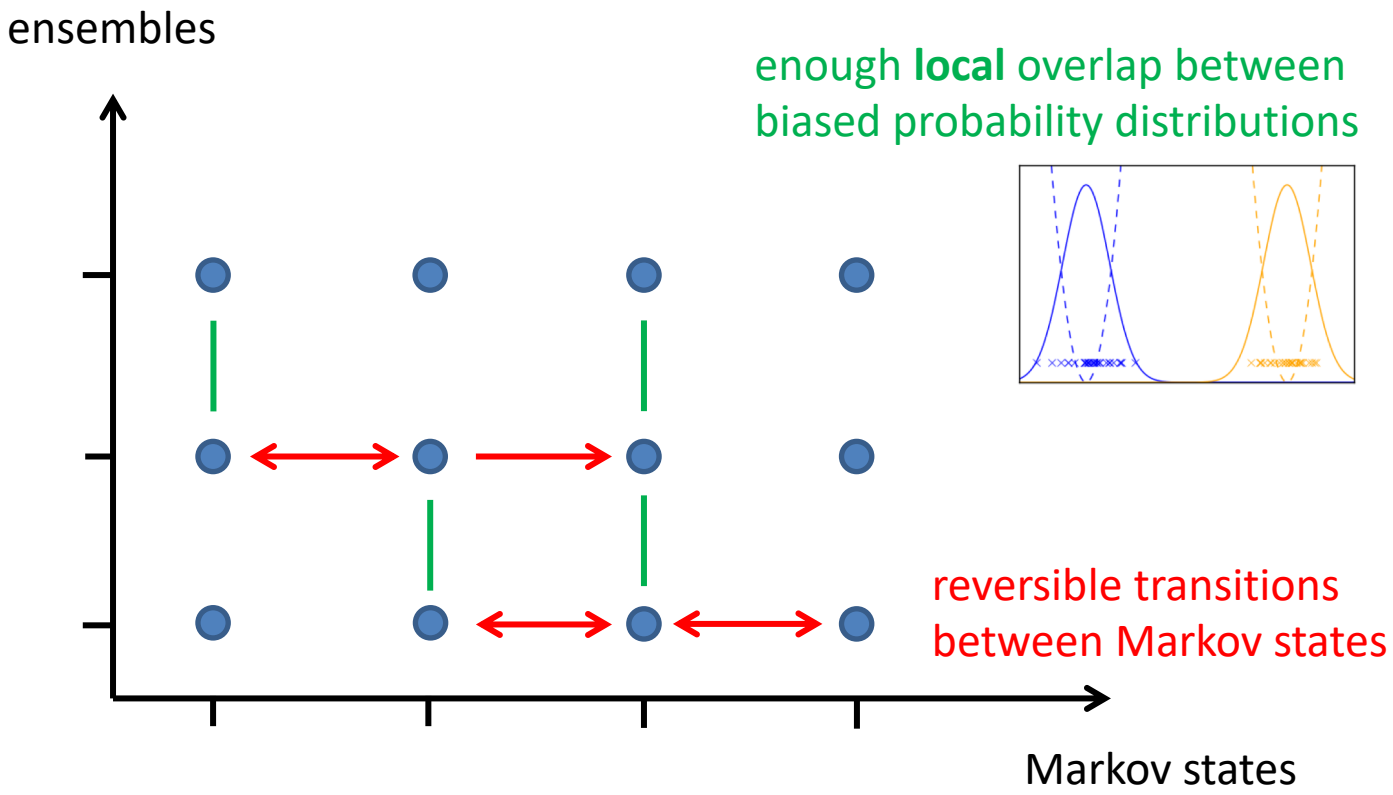
- Post-hoc replica exchange: How many exchanges would have been accepted if the simulation had been carried out with replica exchange between ensembles? How does this number compare to the number of simulated samples?

$$\text{score} = (N+M)\frac{1}{N\,M}\sum_{x\in X^{(k)}}\sum_{y\in X^{(l)}}\min\left[1,\frac{e^{-\beta U^{(k)}(x)}e^{-\beta U^{(l)}(y)}}{e^{-\beta U^{(k)}(y)}e^{-\beta U^{(l)}(x)}}\right] \leqslant 1$$

- error of the free energies estimated by (M)BAR (equation from [1]). Alternative way to relate the overlap of distributions to the number of samples.

[1] Shirts and Chodera, Statistically optimal analysis of samples from multiple equilibrium states, J. Chem. Phys. 129, 124105 (2008)

Overlap in (d)TRAM



Much of this is based on empirical evidence from numerical examples.

summary

- We have introduced the TRAM method which combines Boltzmann reweighting and Markov state models. It replace the histogram-based analysis methods with transition-based methods.
- TRAM allows to combine free-energy calculation (for which we have many tools) with direct MD simulation of the downhill processes (which are easy) to obtain an optimal estimate of the full unbiased kinetics.

Further reading

- Wu, Mey, Rosta, Noé: "Statistically optimal analysis of state-discretized trajectory data from multiple thermodynamic states", *J. Chem. Phys.* **141** 214106 (2014)
- Wu, Paul, Wehmeyer, Noé: "Multiensemble Markov models of molecular thermodynamics and kinetics", *PNAS* E3221–E3230 (2016)

TRAM: Boltzmann reweighting

$$\langle O \rangle^i = \int O(x) p^i(x) \mathrm{d}x \approx \frac{1}{N} \sum_n^N O(x_n)$$

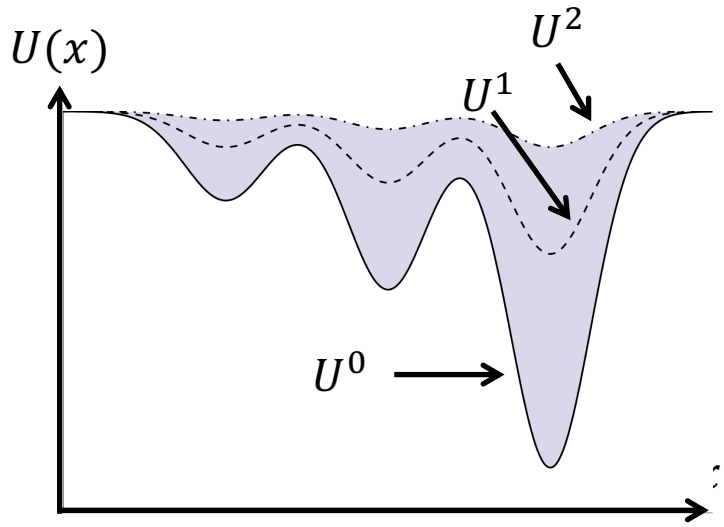
importance sampling

$$\langle O \rangle^i = \int O(x) p^j(x) \frac{p^i(x)}{p^j(x)} \mathrm{d}x$$

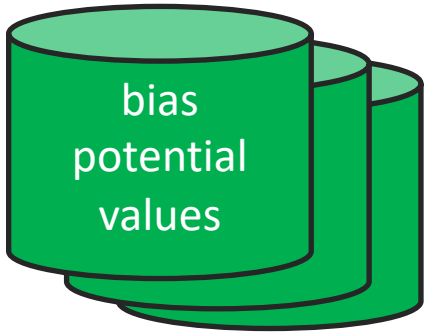
$$\approx \frac{1}{N} \sum_n^N O(x_n) \frac{p^i(x_n)}{p^j(x_n)}$$

in chemistry

$$p^i(x) = e^{\beta F^i - \beta U^i(x)}$$



TRAM: combining normal MD with biased MD



probabilistic model:

$$L = \prod_{i,j,k} \left(T_{ij}^k\right)^{c_{ij}^k} \cdot \prod_k \prod_{x \in X^k} \frac{e^{-b^k(x)}\mu(x)}{z_{s(x)}^k}$$

$$z_i^k T_{ij}^k = z_j^k T_{ji}^k$$

optimize model parameters T and μ (and z[μ])

stationary probabilities (thermodynamics)

kinetic probabilities (rates)

WHAM derivation

$$\log L = \sum_{i,k} N_i^{(k)} \log \pi_i^{(k)}$$

$$= \sum_{i,k} N_i^{(k)} \log \left(\frac{\pi_i \gamma_i^{(k)}}{\sum_j \pi_j \gamma_j^{(k)}} \right)$$

$$= \sum_{i,k} N_i^{(k)} \log \pi_i \gamma_i^{(k)} - \sum_{i,k} N_i^{(k)} \log \sum_j \pi_j \gamma_j^{(k)}$$

$$= \sum_{i,k} N_i^{(k)} \log \pi_i \gamma_i^{(k)} - \sum_k N^{(k)} \log \sum_j \pi_j \gamma_j^{(k)}$$

$$\frac{\partial L}{\partial \pi_n} = \sum_k \frac{N_n^{(k)}}{\pi_n \gamma_n^{(k)}} \gamma_n^{(k)} - \sum_k \frac{N^{(k)} \gamma_n^{(k)}}{\sum_j \pi_j \gamma_j^{(k)}} = 0$$

$$\frac{1}{\pi_n} \sum_k N_n^{(k)} = \sum_k \frac{N^{(k)} \gamma_n^{(k)}}{\sum_j \pi_j \gamma_j^{(k)}}$$

$$\pi_n = \frac{N_n}{\sum_k \frac{N^{(k)} \gamma_n^{(k)}}{\sum_j \pi_j \gamma_j^{(k)}}}$$

(d)TRAM: solution

update equations:

$$\pi_i^{\text{new}} = \frac{\sum_{j,k} c_{ji}^{(k)}}{\sum_{l,j} \frac{\left(c_{ij}^{(l)} + c_{ji}^{(l)}\right)\gamma_i^{(l)}\nu_j^{(l)}}{\gamma_i^{(l)}\pi_i\nu_j^{(l)} + \gamma_j^{(l)}\pi_j\nu_i^{(l)}}}$$

$$\nu_i^{(k),\text{new}} = \nu_i^{(k)} \sum_j \frac{\left(c_{ij}^{(k)} + c_{ji}^{(k)}\right)\gamma_j^{(k)}\pi_j}{\gamma_i^{(k)}\pi_i\nu_j^{(k)} + \gamma_j^{(k)}\pi_j\nu_i^{(k)}}$$

transition matrix:

$$T_{ij}^{(k)} = \frac{\left(c_{ij}^{(k)} + c_{ji}^{(k)}\right)\gamma_j^{(k)}\pi_j}{\gamma_i^{(k)}\pi_i\nu_j^{(k)} + \gamma_j^{(k)}\pi_j\nu_i^{(k)}}$$