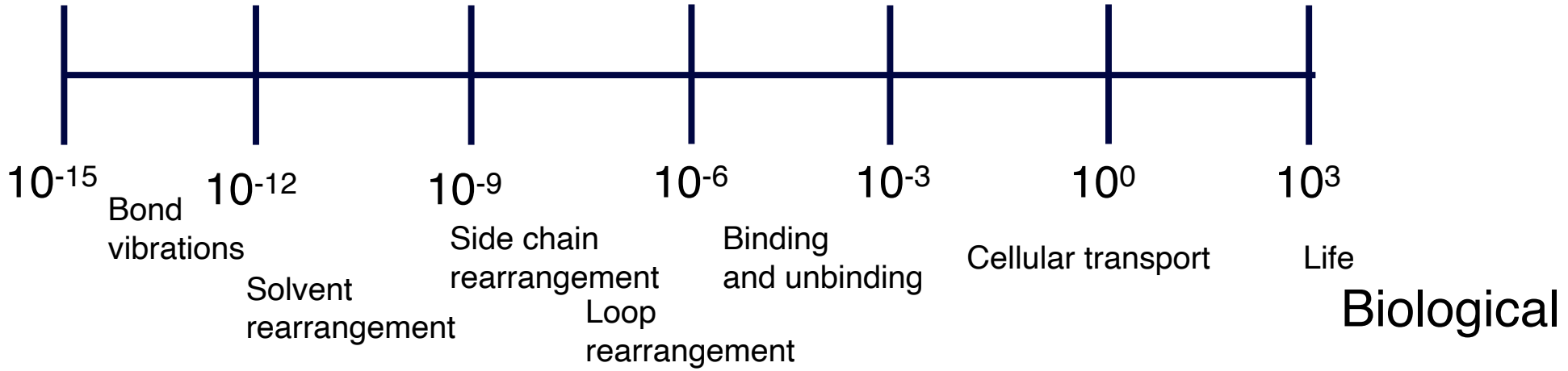
Enhanced Sampling Workshop: Day 1 Theory and Hands-On

Michael R. Shirts
Department of Chemical and Biological Engineering
University of Colorado Boulder

3rd i-CoMSE Workshop: Methods for Enhanced Sampling
March 20, 2023

Why enhanced sampling?

- The Multiple Time Scale Problem
  - Interesting molecular processes involve multiple time scales!
  - If it were just long time scale, we could just do that directly!



Enhanced Sampling for Thermodynamics

- Example: Calculating a binding affinity
- Break some physics laws, but not others
- Keep:
  - $p(x) \propto \exp(-\beta U(x))$
  - Thermodynamics
- Break:
  - Continuity of molecular trajectories
  - F=mA
  - Kinetics

Enhanced Sampling for Kinetics

- Trajectories as SERIES of configurations
- Break:
  - F=mA
  - $p(x) \propto \exp(-\beta U(x))$ (maybe)
  - Whether a trajectory is fully continuous or not
  - BUT break in ways that one can recover the results afterwards

A complete overview of using simulations to calculate thermodynamics (in 5 min)

All thermodynamic properties are ratios of highly multidimensional integrals over coordinates

$$e^{-f(\vec{a}_1)+f(\vec{a}_2)} = \frac{\int e^{-u(\vec{a}_1,\vec{x})}d\vec{x}}{\int e^{-u(\vec{a}_2,\vec{x})}d\vec{x}}$$

f = βA u(x) = βU(x)

x = all coordinates of the system

a = parameters of the Hamiltonian of the system (T,P, force field)

$$\langle O(\vec{a})\rangle = \frac{\int O(\vec{a},\vec{x})e^{-u(\vec{a},\vec{x})}d\vec{x}}{\int e^{-u(\vec{a},\vec{x})}d\vec{x}}$$

O is some observable of the system: volume, radius of gyration, etc.

A complete overview of using simulations to calculate thermodynamics (in 5 min)

Use molecular simulations to perform Monte Carlo integration of these ratios of integrals

$$\langle O \rangle_i = \int O(\vec{x}) p_i(\vec{x}) d\vec{x} \quad = \frac{1}{N} \sum_{n=1}^{N} O(\vec{x}_n)$$

x_n sampled from $p_i(x)$

- This is why when you calculate average energy from a simulation trajectory, you don't use Boltzmann factors - they were automatically included by running the simulation!

A complete overview of using simulations to calculate thermodynamics (in 5 min)

- Generate samples from $p_i(x)$ using:
  - Markov Chain Monte Carlo (MCMC)
  - Given the current state,
    - assuming it has the right distribution,
    - select a new state that preserves this given probability distribution.
  - Eventually*, will converge to the right distribution

  - Molecular dynamics
    - ACTUALLY a type of MCMC, in the direction of force, with 100%* acceptance
    - Apply Newton's equation of motion: $dx^2/dt = -m_i^{-1}\nabla U(x)$:
    - solve the coupled PDE by as an initial value problem.
- Whatever the method, it will have **long** correlation times

A complete overview of using simulations to calculate thermodynamics (in 5 min)

Identify improved ways to accelerate sampling through the multidimensional space of interest

- Umbrella sampling
- Replica exchange
- Hamiltonian replica exchange
- Accelerated MD
- Multicanonical simulation
- Adaptive biasing force
- Weighted ensemble
- Metadynamics . . . .

Identify improved ways to estimate multidimensional integrals given a set of molecular simulation data

- Free energy perturbation
- Bennett acceptance ratio
- WHAM
- MBAR . . .

"Taco Bell" formulation of enhanced sampling

Limited number of stat mech ingredients

"Taco Bell" formulation of enhanced sampling

Combinatorially large number of methods











"Taco Bell" formulation of enhanced sampling

Limited number of ingredients

Construct bias along a collective variable

Introduce dynamics along a auxiliary variable

Interpret a derivative as a expectation value

Assume things are Gaussian

Express as a maximum likelihood problem

Accumulate partition function in some DOF adaptively

Integrate out degrees of freedom

"Machine Learning"

Use importance sampling to calculate unsampled averages

Let's understand the Boltzmann distribution some more

- It's often much clearer to work with **probability** than with energies.

$$p_i(x) \propto e^{-\beta U_i(x)}$$

Means "proportional to"

$$p_i(x) = \frac{e^{-\beta U_i(x)}}{Q_i}$$

"Normalizing constant" or "partition function"

$$p_i(x) = e^{\beta A_i - \beta U_i(x)}$$

Where $A_i = - k_B T \ln Q_i$, the free energy

We just need to do some sampling

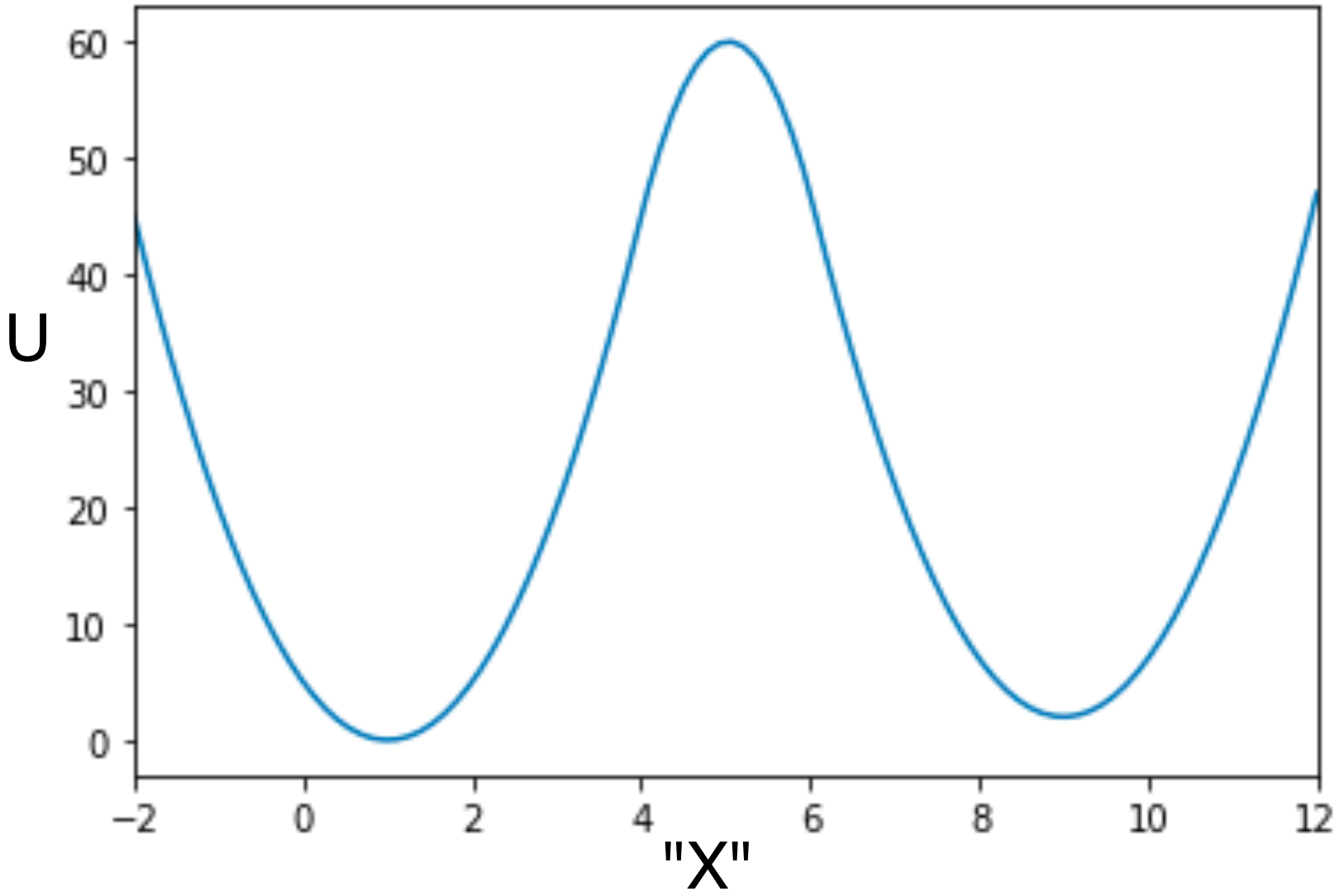
- Our goal: sample representative x_i from all of the distribution:

$$p_i(x) = e^{\beta A_i - \beta U_i(x)}$$

- Problem: the distribution looks like this:

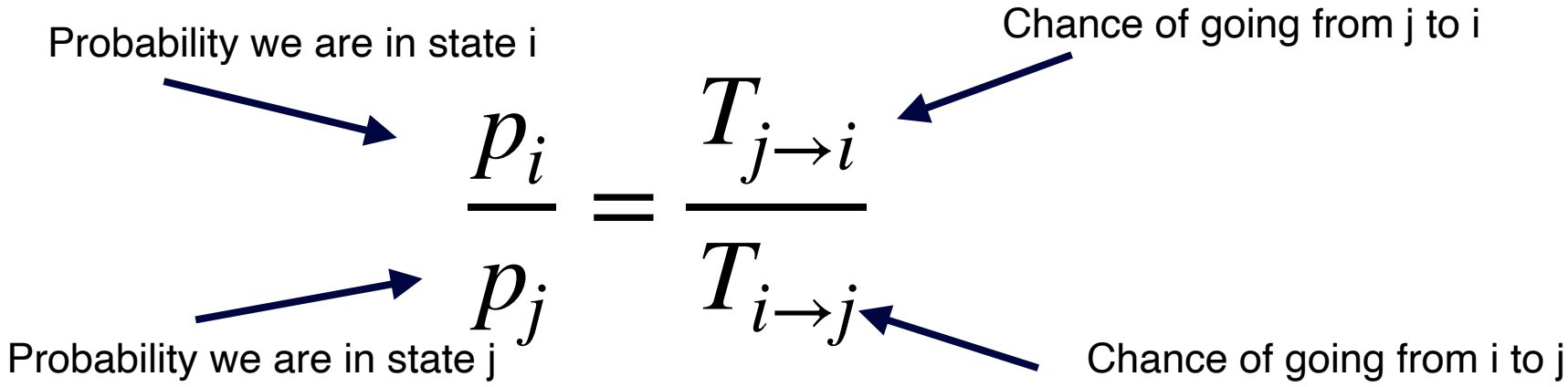
- 10,000 atoms means it's a 30,000 dimensional problem
- Only a vanishingly small fraction of the x_i have high probability (low energy)

A toy problem
(let's go to the notebook!)



Reminder: Markov Chain Monte Carlo

- We want a rule for flow between states, that preserves the probability AT the states



- We keep the ratio of "stuff" in state i and j equal by making sure the ratio of stuff going IN and OUT is consistent
- **Detailed balance**

Reminder: Markov Chain Monte Carlo

$$\frac{p(x_i)}{p(x_j)} = \frac{e^{\beta A - \beta U(x_i)}}{e^{\beta A - \beta U(x_i)}} = e^{\beta\left(U(x_j) - U(x_i)\right)}$$

Any rule that satisfies this works! $$\frac{T_{x_j \to x_i}}{T_{x_i \to x_j}} = e^{\beta\left(U(x_j) - U(x_i)\right)}$$

A rule: Step 1: Propose a move randomly and isotropically in any direction
Step 2: If $U(x_j) < U(x_i)$, then move there
If $U(x_j) > U(x_i)$, then move with probability: $e^{\beta\left(U(x_j) - U(x_i)\right)}$

Metropolis algorithm
(actually invented by Marshall and Arianna Rosenbluth, 1951)

Checking the rule

Step 1: Propose a move randomly and isotropically in any direction

This is really important, because otherwise, the change of proposing a move from i is not the same as the reverse.

$$T_{i\to j} = \alpha(i \to j)P(i \to j)$$

Chance of going from i to j

Chance of **proposing** a move from i to j

Chance of **accepting** a move from i to j

Step 2:

$$\frac{T_{j\to i} = \alpha(j \to i)P(j \to i)}{T_{i\to j} = \alpha(i \to j)P(i \to j)} = \frac{P(j \to i)}{P(i \to j)} =$$

if $U(x_i) < U(x_j)$

$$\frac{1}{e^{\beta\left(U(x_i) - U(x_j)\right)}} = \checkmark \; e^{\beta\left(U(x_j) - U(x_i)\right)}$$

if $U(x_i) > U(x_j)$

$$\frac{e^{\beta\left(U(x_j) - U(x_i)\right)}}{1} = e^{\beta\left(U(x_j) - U(x_i)\right)}$$

Back to the notebook

Importance sampling can allow us to remove a bias

$$\langle O \rangle_i = \int O(\vec{x}) p_i(\vec{x}) d\vec{x} = \int O(\vec{x}) p_i(\vec{x}) \frac{p_j(\vec{x})}{p_j(\vec{x})} d\vec{x}$$

$$= \int O(\vec{x}) p_j(\vec{x}) \frac{p_i(\vec{x})}{p_j(\vec{x})} d\vec{x} \qquad \langle O \rangle_i = \frac{1}{N} \sum_{n=1}^{N} O(\vec{x_n}) \frac{p_i(\vec{x_n})}{p_j(\vec{x_n})}$$

x_n sampled from $p_j(x)$

Sampling from $p_j(x)$, observables from $p_i(x)$

If $p_i(x)/p_j(x)$ is similar in magnitude as x_n varies, then we can avoid generating the entire Monte Carlo chain

If $p_i(x)/p_j(x)$ varies too much, then it becomes very noisy!

How can we remove a bias?

$$\frac{p_i(x)}{p_j(x)} = \frac{e^{\beta A_i - \beta U_i(x)}}{e^{\beta A_j - \beta U_j(x)}} = e^{\beta \Delta A - \beta \Delta U(x)}$$

So we we can calculate this!

How do we figure out the free energy difference ΔA?

$$\frac{p_i(x)}{p_j(x)} = \frac{e^{\beta A_i - \beta U_i(x)}}{e^{\beta A_j - \beta U_j(x)}} = e^{\beta \Delta A - \beta \Delta U(x)}$$

$$O(\vec{x}) = 1$$

$$\langle O \rangle_i = \frac{1}{N} \sum_{n=1}^{N} O(\vec{x}_n) e^{\beta \Delta A - \beta \Delta U(\vec{x}_n)}$$

$$e^{-\beta \Delta A} = \frac{1}{N} \sum_{n=1}^{N} e^{-\beta \Delta U_i(\vec{x})}$$

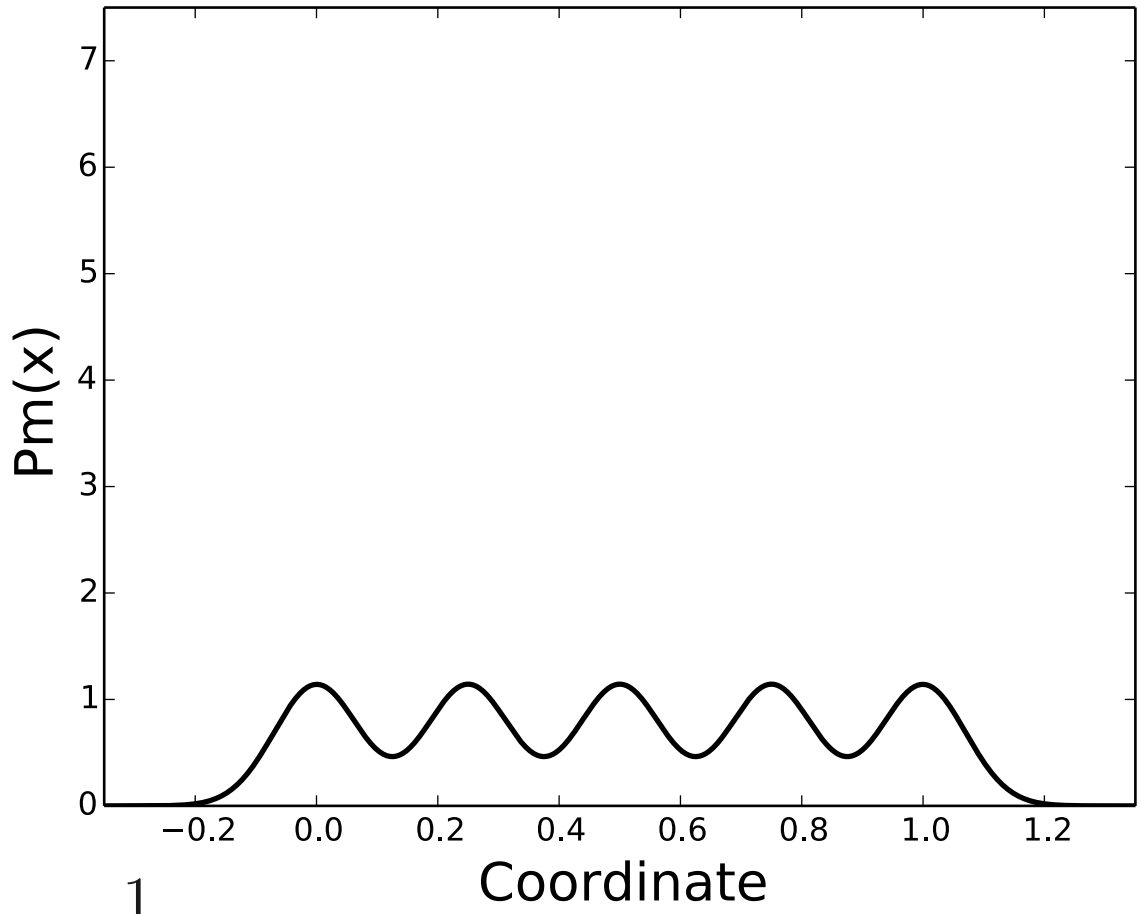
Zwanzig /
Exponential averaging

Finally, we get:

$$\langle O \rangle_i = \frac{\frac{1}{N} \sum_{n=1}^{N} O(\vec{x}_n) e^{-\beta U(\vec{x}_n)}}{\frac{1}{N} \sum_{n=1}^{N} e^{-\beta U(\vec{x}_n)}}$$

Back to the notebook!

How do we remove multiple biases?

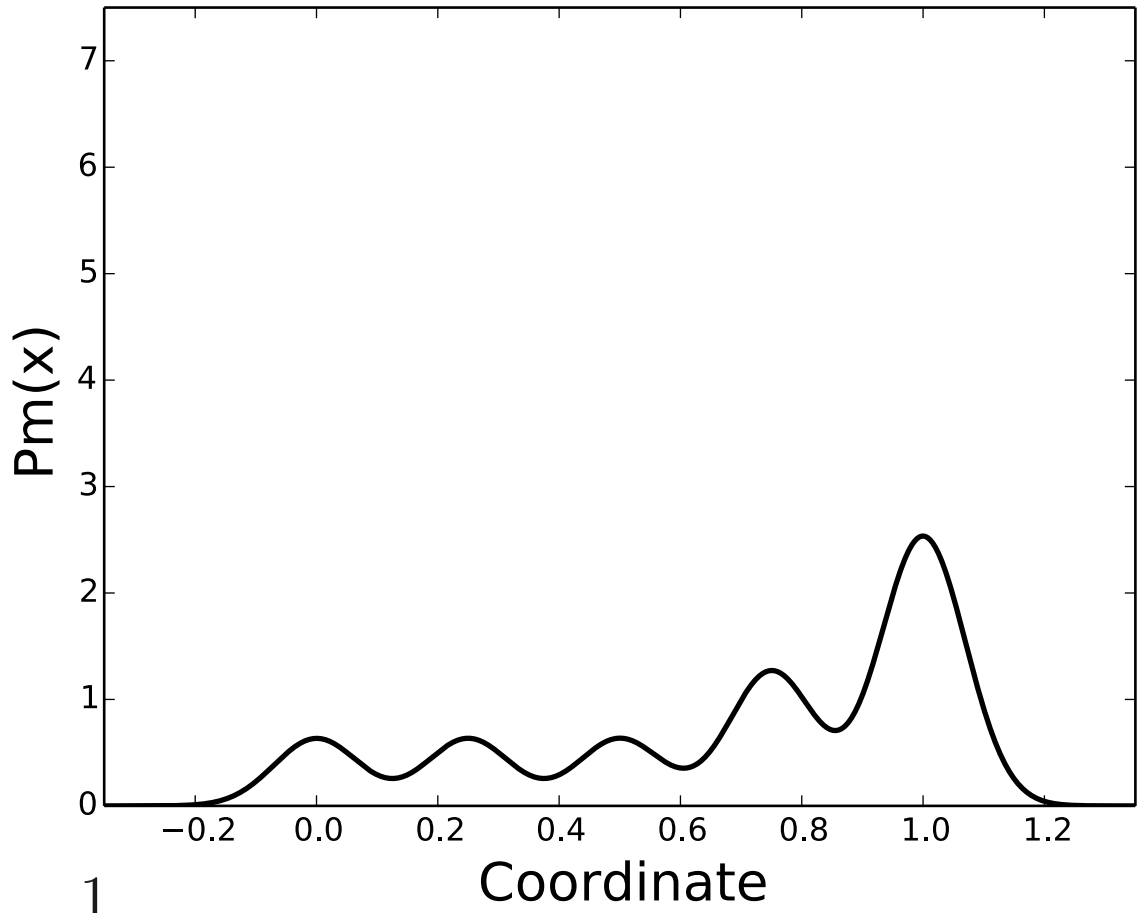


Different numbers of samples?
N = [10,10,10,20,40]

$$p_m(\vec{x}) = \frac{1}{5}\left[p_1(\vec{x}) + p_2(\vec{x}) + p_3(\vec{x}) + p_4(\vec{x}) + p_5(\vec{x})\right]$$

$p_m(x)$ is called a ‘mixture distribution’

How do we remove multiple biases?

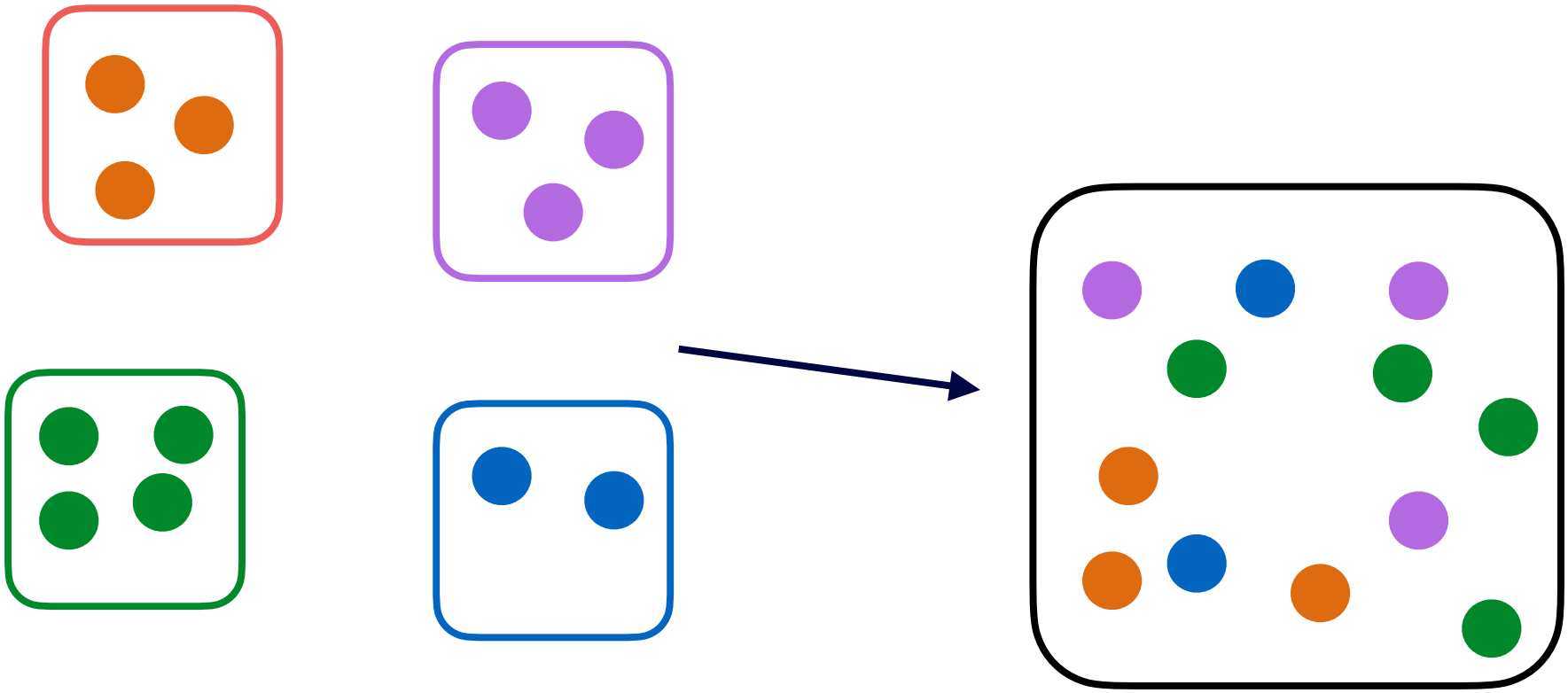


Different numbers of samples?
N = [10,10,10,20,40]

$$p_m(\vec{x}) = \frac{1}{\sum_{k=1}^{5} N_k} \left[N_1 p_1(\vec{x}) + N_2 p_2(\vec{x}) + N_3 p_3(\vec{x}) + N_4 p_4(\vec{x}) + N_5 p_5(\vec{x}) \right]$$

$p_m(x)$ is called a ‘mixture distribution’

We create a MIXTURE distribution to reweight from!



$$p_m(\vec{x}) = \frac{1}{\sum_{k=1}^{5} N_k} \left[N_1 p_1(\vec{x}) + N_2 p_2(\vec{x}) + N_3 p_3(\vec{x}) + N_4 p_4(\vec{x}) + N_5 p_5(\vec{x}) \right]$$

$p_m(x)$ is the ‘mixture distribution’

Importance sampling from the mixture distribution gives lowest variance estimate

$$e^{-\beta A_i} = \frac{1}{N}\sum_{n=1}^{N}\frac{e^{-\beta U_i(x_n)}}{\sum_k \frac{N_k}{N} e^{\beta A_k - \beta U_k(x_n)}}$$

$$\langle O \rangle_i = \frac{1}{N}\sum_{n=1}^{N} O(x_n)\frac{e^{\beta A_i - \beta U_i(x_n)}}{\sum_k \frac{N_k}{N} e^{\beta A_k - \beta U_k(x_n)}}$$

$$p_m(x) = \sum_k \frac{N_k}{N} p_k(x) = \sum_k \frac{N_k}{N} e^{\beta A_k - \beta U_k(x)}$$

$$e^{-\beta A_i} = \frac{1}{N}\sum_{n=1}^{N}\frac{e^{-\beta U_i(x_n)}}{p_m(x)}$$

$$\langle O \rangle_i = \frac{1}{N}\sum_{n=1}^{N} O(x_n)\frac{e^{\beta A_i - \beta U_i(x_n)}}{p_m(x)}$$

- How do we find the A_i?????
- We have one equation for each A_i, so N equations and N unknowns
- (Technically, N-1 equations and N-1 unknowns)
- It's a set of nonlinear equations, and somebody has already written code for it.

pymbar: A Python implementation of reweighting from the mixture distribution

- Fast, robust, and efficient solution of the multistate reweighting equations
- Includes many examples
  - Potentials of mean force
  - Solvation free energies
  - Force bias single molecule experiments
- Automated setup.py and installation through conda
- https://github.org/choderalab/pymbar

This generally approach has a bunch of different variants and MANY names

- Sometimes called the Multistate Bennett Acceptance Ratio (MBAR)
  - Shirts and Chodera, *J. Chem. Phys.*, 129:124105 (2008)
- It reduces to:
  - Weighted histogram analysis method (WHAM) in the limit of zero-width histogram bins
  - Bennett acceptance ratio for two states
- Is the maximum likelihood estimate of all data collected
- Provably the lowest variance reweighting estimator

WARNING!! WARNING!!!

- How do I know if I've sampled enough?
- What if I choose bad directions to bias along?
- What are the uncertainties in my estimates?

How do I know if I've sampled enough?

- Enhanced sampling will give you better sampling along the degrees of freedom you FORCED it to sample better.
- It will NOT necessarily help you sample degrees of freedom that you did not force
- "orthogonal" degrees of freedom.
- You need to monitor those;
- You want to speed things up along the slowest DOF. How do you know them???
- You can't see . . . what you don't see . . .

Example of orthogonal degrees of freedom you may miss

- Pulling two proteins apart
  - Use some enhanced sampling in the direction of the COM of the protein.
- However, perhaps the proteins need to rearrange in order to come apart.
- Pulling them apart does not necessarily speed up the rearrangement
  - If you pull hard enough, you may be able to open a door without turning the handle. . . .