# Multi-Ensemble Markov Models and TRAM

Maaike Galama

22-02-2022

## MD/MC – challenge

Why MD/MC? → sample from the Boltzmann distribution

$$P(x) = \frac{e^{-\beta U(x)}}{\int e^{-\beta U(x)}}, \qquad \beta = k_B T^{-1}$$

Goal → estimate macroscopic properties of system (not individual trajectories)

## MD/MC – challenge

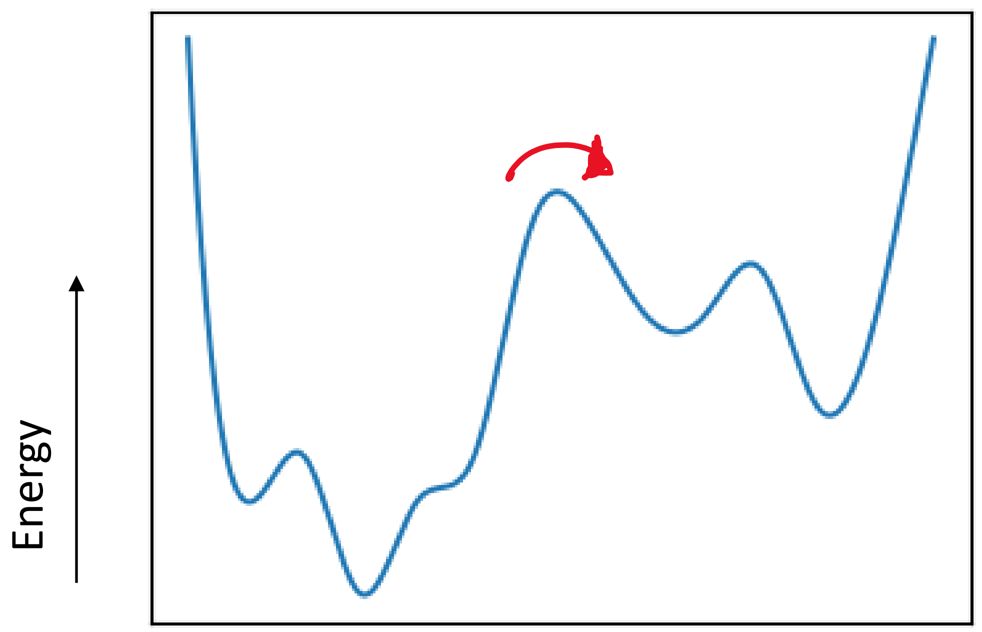
Why MD/MC? → sample from the Boltzmann distribution

$$P(x) = \frac{e^{-\beta U(x)}}{\int e^{-\beta U(x)}}, \qquad \beta = k_B T^{-1}$$

Problems:

1. Systems are high-dimensional
2. Rare events might not occur during simulation timescale

## Rare events



# Contents

- Importance sampling

- Enhanced sampling methods
  - Umbrella sampling
  - Multi-temperature simulations

- Analysis methods
  - Reweighting methods (WHAM, MBAR)
  - Multi-ensemble Markov Models (d-TRAM, TRAM)

## Importance sampling

Compute observable w.r.t. $F(x)$

$$\mathbb{E}_F[O(x)] = \int O(x)\, F(x)\, dx \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)$$

Law of large numbers

→ Only holds when we can sample from $F(x)$!

## Importance sampling

Problem: we have distribution $F(x)$ that is “hard” to sample from

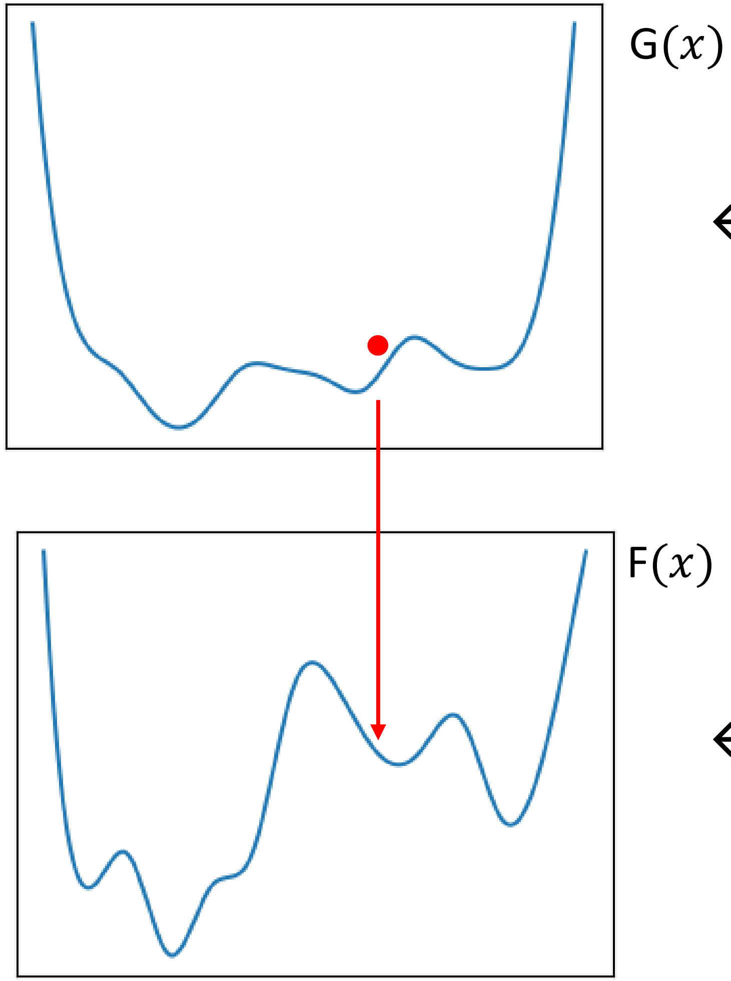
Idea:

- sample from an easier to sample distribution $G(x)$
- compute observables properties that belong to $F(x)$ with the samples drawn from $G(x)$.

*(Not really a sampling algorithm, more of a general technique)*

Kahn, Herman, and Andy W. Marshall. "Methods of reducing sample size in Monte Carlo computations." *Journal of the Operations Research Society of America* 1.5 (1953): 263-278.

# Importance sampling



← Sample by simulating this system…

← … and compute observables for this system

## Importance sampling

$$\mathbb{E}_F[O(x)] = \int O(x)F(x)dx$$

$$= \int O(x)\frac{G(x)}{G(x)}F(x)\,dx$$

## Importance sampling

$$\mathbb{E}_F[O(x)] = \int O(x)F(x)dx$$

$$= \int O(x)\frac{F(x)}{G(x)}G(x)\, dx$$

## Importance sampling

$$\mathbb{E}_F[O(x)] = \int O(x)F(x)dx$$

$$= \int \boxed{O(x)\frac{F(x)}{G(x)}} G(x)\, dx$$

New observable: $O'(x) = O(x)\frac{F(x)}{G(x)}$

## Importance sampling

$$\mathbb{E}_F[O(x)] = \int O(x)F(x)dx$$

New sampling distribution: $G(x)$

$$= \int \boxed{O(x)\frac{F(x)}{G(x)}}\boxed{G(x)}dx$$

New observable: $O'(x) = O(x)\frac{F(x)}{G(x)}$

## Importance sampling

$$\mathbb{E}_F[O(x)] = \int O(x)F(x)dx$$

$$= \int \boxed{O(x)\frac{F(x)}{G(x)}}\,\boxed{G(x)}\,dx$$

New sampling distribution: $G(x)$

$$= \mathbb{E}_G[O'(x)]$$

New observable: $O'(x) = O(x)\frac{F(x)}{G(x)}$

This is now an expectation value over $G(x)$!

# Importance sampling

$$\mathbb{E}_G[O'(x)] = \mathbb{E}_F[O(x)]$$

$$= \int O(x)\,\frac{F(x)}{G(x)}\,G(x)\;dx$$

## Importance sampling

$$\mathbb{E}_G[O'(x)] = \mathbb{E}_F[O(x)]$$

$$= \int O(x)\,\frac{F(x)}{G(x)}\,G(x)\,dx$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\,\frac{F(x_i)}{G(x_i)}$$

## Importance sampling

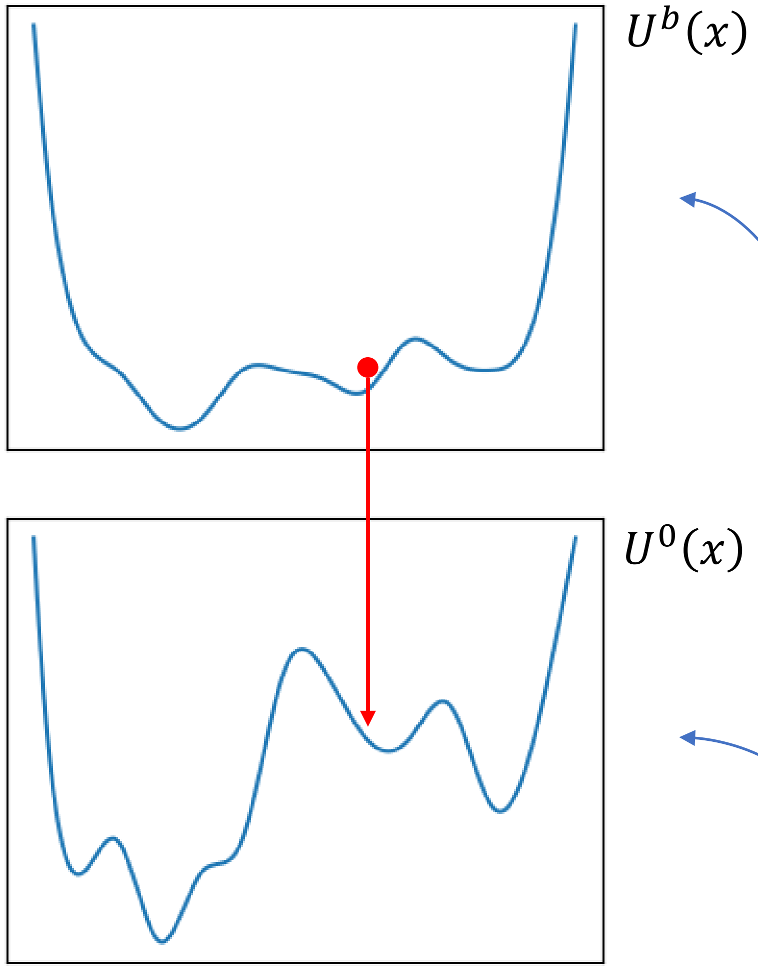
$$\mathbb{E}_G[O'(x)] = \mathbb{E}_F[O(x)]$$

$$= \int O(x)\frac{F(x)}{G(x)}G(x)\,dx$$

$$\approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\frac{F(x_i)}{G(x_i)}$$

$x_i \sim G(x)$

# Importance sampling in MD



Our distributions are now Boltzmann distributions

$$\varphi^0(x) = \frac{1}{Z^0} e^{-\beta U^0(x)} = e^{-\beta(U^0(x) - F^0)}$$

$(F = -\beta^{-1} \log Z)$

Easier to sample distribution: biased potential $U^b(x)$

Target (physical) distribution: reference potential $U^0(x)$

# Importance sampling in MD

Our distributions are now Boltzmann distributions

Target distribution: determined by reference potential $U^0(x)$

Easier to sample distribution: biased potential $U^b(x)$

$$\mathbb{E}_0[O(x)] = \int O(x) e^{-\beta(U^0(x) - F^0)} dx$$

$$= \int O(x) \frac{e^{-\beta(U^b(x) - F^b)}}{e^{-\beta(U^b(x) - F^b)}} e^{-\beta(U^0(x) - F^0)} dx$$

## Importance sampling in MD

Our distributions are now Boltzmann distributions

Target distribution: determined by reference potential $U^0(x)$

Easier to sample distribution: biased potential $U^b(x)$

$$\mathbb{E}_0[O(x)] = \int O(x) e^{-\beta(U^0(x) - F^0)} dx$$

$$= \int O(x) \frac{e^{-\beta(U^0(x) - F^0)}}{e^{-\beta(U^b(x) - F^b)}} e^{-\beta(U^b(x) - F^b)} dx$$

# Importance sampling in MD

Our distributions are now Boltzmann distributions

Target distribution: determined by reference potential $U^0(x)$

Easier to sample distribution: biased potential $U^b(x)$

$$\mathbb{E}_0[O(x)] = \int O(x) e^{-\beta(U^0(x)-F^0)} dx$$

$$= \int O(x) \frac{e^{-\beta(U^0(x) - F^0)}}{e^{-\beta(U^b(x) - F^b)}} e^{-\beta(U^b(x) - F^b)} dx$$

$$\approx \frac{1}{N} \sum_{i=1}^{N} O(x_i)\, e^{-\beta(U^0(x) - U^b(x) - F^0 + F^b)}$$

Where $x_i \sim \varphi^b(x)$

## Importance sampling in MD: Boltzmann reweighting

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{-\beta(U^0(x) - U^b(x) - F^0 + F^b)}$$

- $U^0(x)$: the unbiased or physical energy
- $U^b(x)$: the biased energy
- $b(x) = U^b(x) - U^0(x)$: the bias energy

$\rightarrow$ Biased state defined by bias energy

$$U^b(x) = U^0(x) + b(x)$$

# How to use importance sampling?

- Choose a nice bias to more efficiently sample the space

$\rightarrow$Enhanced sampling methods

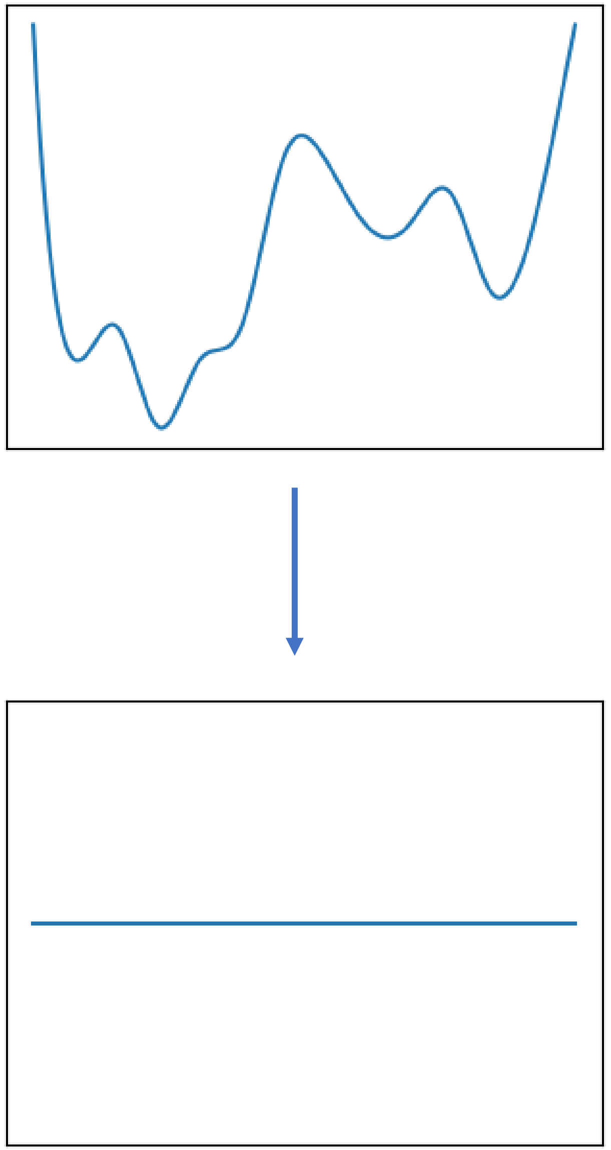
## The perfect bias?

A uniform distribution is very easy to sample

$\rightarrow U^b(x) = 0$

Bias energy:

$$b(x) = U^b(x) - U^0(x) = -U^0(x)$$



## The perfect bias?

Now in 3N dimensions…

- Sample a uniform distribution 3N-d space?
  - Many samples will be from high-energy states
  - Samples with $p \approx 0$ do not contribute to expectation values
- Generally, we are interested in some transition
  - Passing through a membrane
  - Protein-ligand binding
  - Protein (un)folding

→ reaction coordinate

## The perfect bias?

Now in 3N dimensions…

- How to sample a uniform distribution 3N-d space?
- Many samples will be from high-energy states
- Generally, we are interested in some transition
  - Passing through a membrane → coordinate on axis perpendicular to membrane
  - Protein-ligand binding → distance between protein and ligand, MSEs
  - Protein (un)folding → it's complicated

→ reaction coordinate

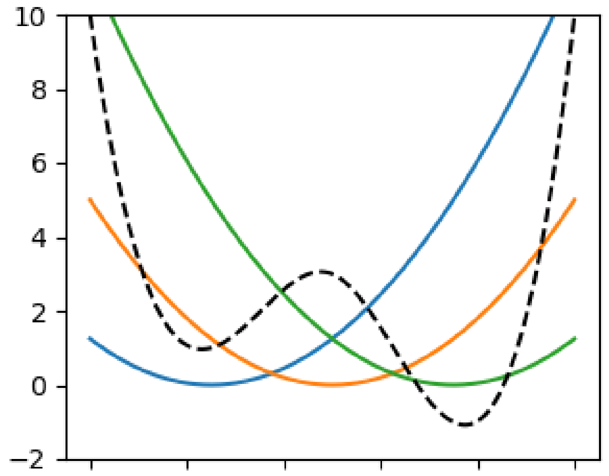
## Umbrella sampling

- Enforce uniform(-ish) sampling along the reaction coordinate
- Define K biased states: $U^k(x) \ = U^0(x) + b^k(x)$
- Usually, $b^k(x) = \frac{1}{2}(x \ - x_0^k)^2$ → ‘umbrellas’
- Bias potentials enforce sampling around their bias center $x_0^k$

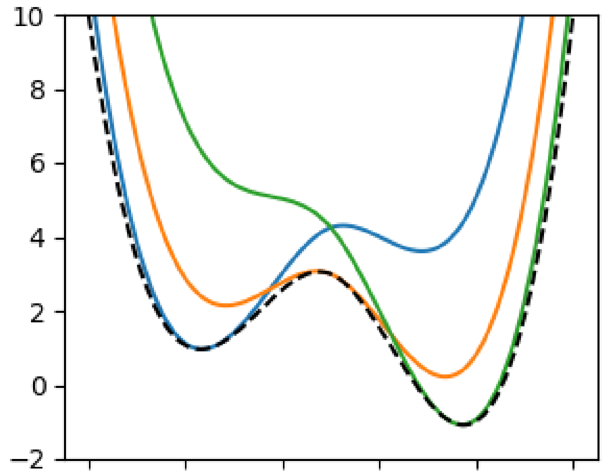
Torrie, Glenn M., and John P. Valleau. "Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling." *Journal of Computational Physics* 23.2 (1977): 187-199.

# Umbrella sampling

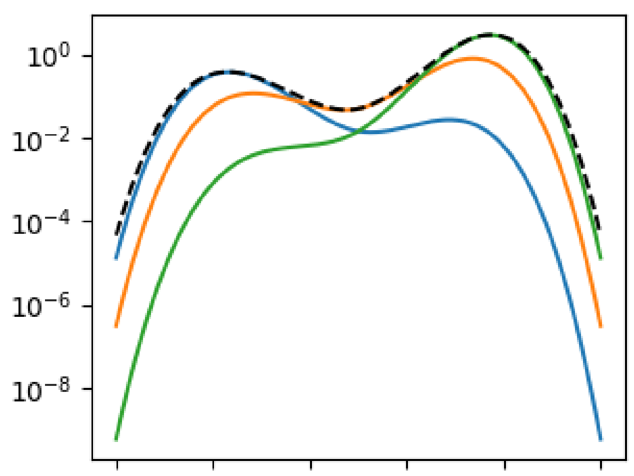
- Enforce uniform(-ish) sampling along the reaction coordinate
- Define K biased states: $U^k(x) \ = U^0(x) + b^k(x)$
- Usually, $b^k(x) = \frac{1}{2}(x \ - x_0^k)^2$ → 'umbrellas'
- Bias potentials enforce sampling around their bias center $x_0^k$



Bias potentials $b^k(x)$



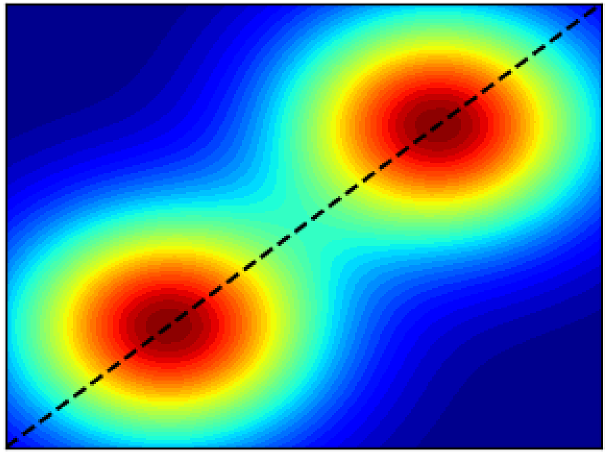
$U^k(x) \ = U^0(x) + b^k(x)$

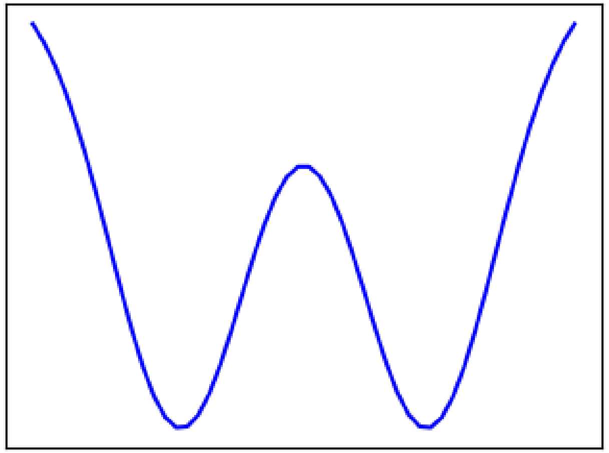


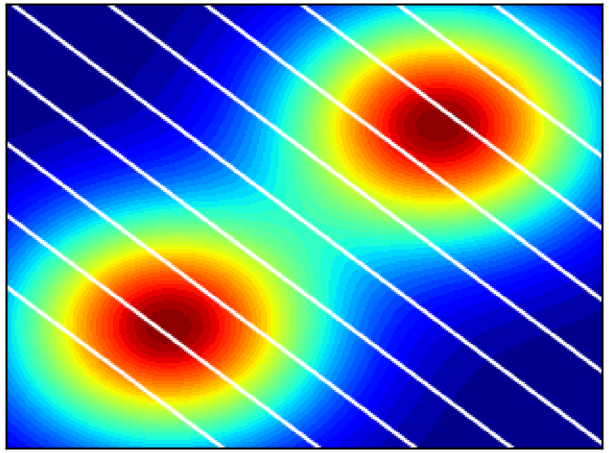
$\varphi^k(x) \propto \ e^{-\beta(U^0(x) - b^k(x))}$

# Umbrella sampling

In higher dimensions: umbrella’s are spaced along the reaction coordinate


Choose reaction coordinate along transition region


Free energy along reaction coordinate


Umbrella’s force system in orthorgonal regions along reaction coordinate

## Parallel tempering

- Idea: high temperatures ‘flatten’ the Boltzmann distribution
- Higher-energy states become more accessible at higher temperatures

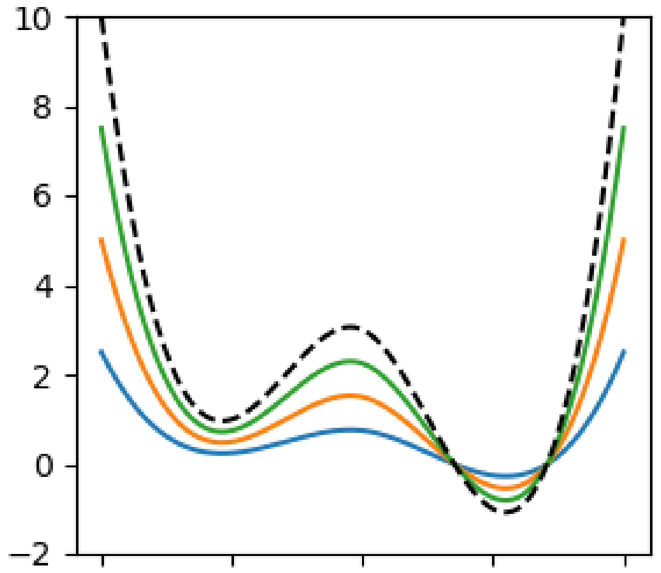
$$\varphi^k(x) \propto e^{-\beta^k U^0(x)}$$

- Bias potentials determined by temperatures

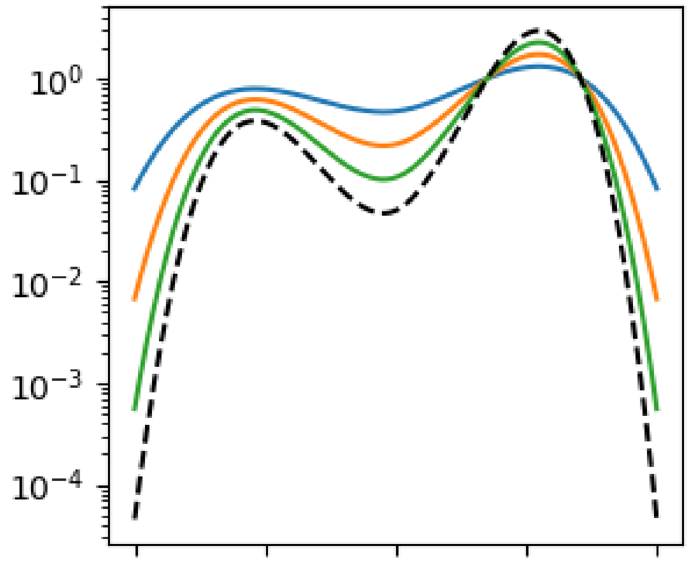
$$b^k = (\beta^k - \beta^0)U^0(x)$$

# Parallel tempering

- Idea: high temperatures ‘flatten’ the Boltzmann distribution
- Higher-energy states become more accessible at higher temperatures



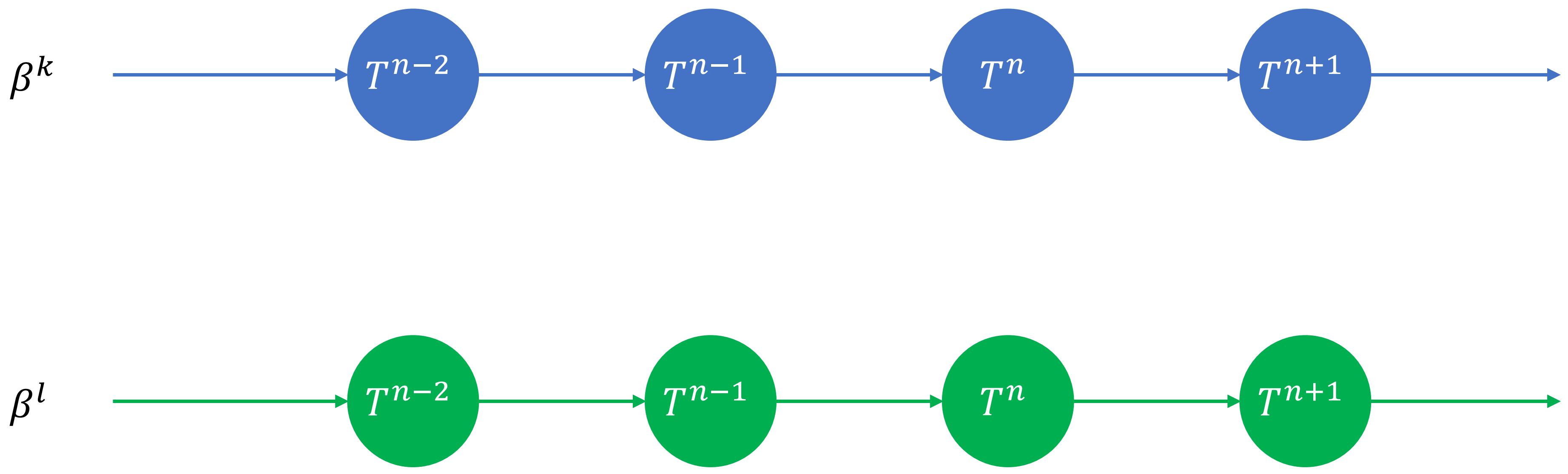
$u^k(x) = \beta^k U(x)$



$\varphi^k(x) \propto e^{-\beta^k U\,(x)}$

## Replica exchange

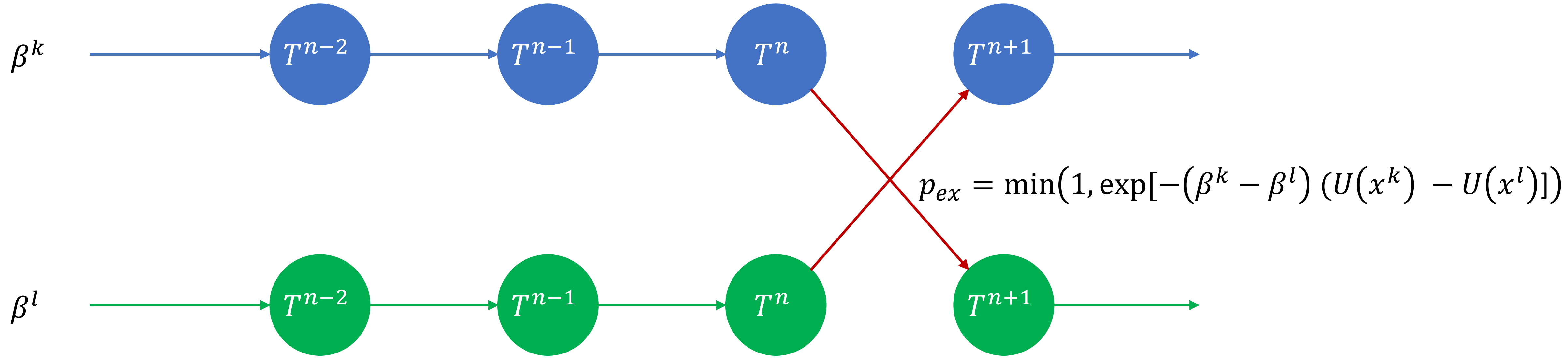
- Swap conformations between states with different temperatures



Swendsen, Robert H., and Jian-Sheng Wang. "Replica Monte Carlo simulation of spin-glasses." *Physical review letters* 57.21 (1986): 2607.

# Replica exchange

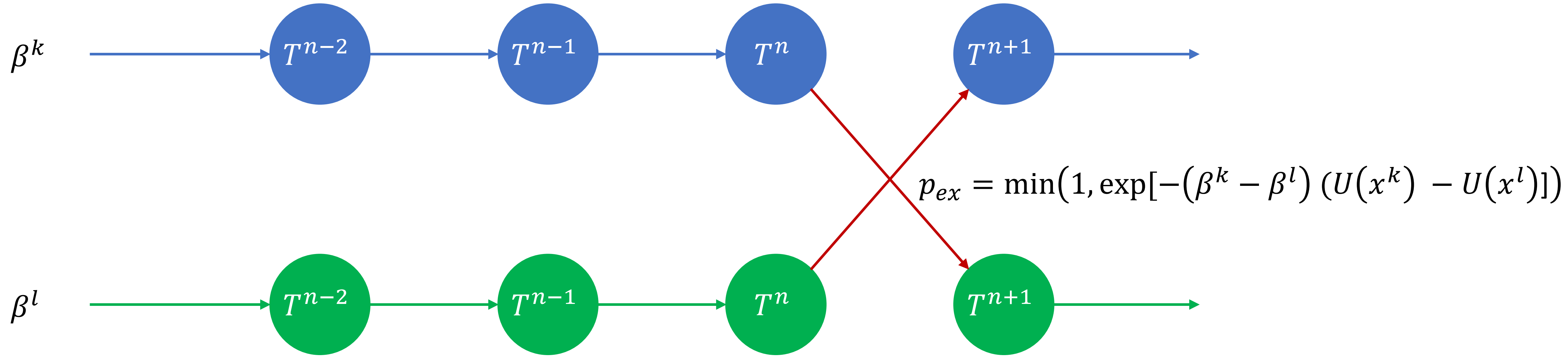
- Swap conformations between states with different temperatures



$$p_{ex} = \min(1, \exp[-(\beta^k - \beta^l)\,(U(x^k) - U(x^l)])$$

# Replica exchange

- Swap conformations between states with different temperatures



$$p_{ex} = \min\left(1, \exp[-\left(\beta^k - \beta^l\right)\left(U\left(x^k\right) - U\left(x^l\right)\right]\right)$$

- High-energy states become accessible at lower temperatures

## Analyzing data

Enhanced sampling:

→Data from different thermodynamic states

→How to recombine?

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\ \frac{e^{-\beta^0 U^0(x) + \beta^0 F^0}}{e^{-\beta^k U^k(x) + \beta^k F^k}}$$

# Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{-\beta^0 U^0(x) + \beta^k U^k(x) + \beta^0 F^0 - \beta^k F^k}$$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{-\beta^0 U^0(x) + \beta^k U^k(x) + \beta^0 F^0 - \beta^k F^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^k U^k(x) - \beta^0 U^0(x)$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) + \beta^0 F^0 - \beta^k F^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^k U^k(x) - \beta^0 U^0(x)$

# Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) + \beta^0 F^0 - \beta^k F^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^k U^k(x) - \beta^0 U^0(x)$
→ **Dimension-less** Free energy $f^k = \beta^k F^k$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) + f^0 - f^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^b U^b(x) - \beta^0 U^0(x)$

→ **Dimension-less** Free energy $f^k = \beta^k F^k$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) + f^0 - f^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^b U^b(x) - \beta^0 U^0(x)$

→ **Dimension-less** Free energy $f^k = \beta^k F^k$

→ Free energy **differences**: $f^0 \coloneqq 0,\ f^k \coloneqq f^k - f^0$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) + \cancel{f^0} - f^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^b U^b(x) - \beta^0 U^0(x)$

→ **Dimension-less** Free energy $f^k = \beta^k F^k$

→ Free energy **differences**: $f^0 \coloneqq 0,\ f^k \coloneqq f^k - f^0$

## Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) - f^k}$$

→ **Dimension-less** bias energy: $b^k(x) = \beta^b U^b(x) - \beta^0 U^0(x)$
→ **Dimension-less** Free energy $f^k = \beta^k F^k$
→ Free energy **differences**: $f^0 \coloneqq 0,\ f^k \coloneqq f^k - f^0$

☺

# Analyzing data

Boltzmann reweighting:

$$\mathbb{E}_0[O(x)] \approx \frac{1}{N}\sum_{i=1}^{N} O(x_i)\, e^{b^k(x) - \boxed{f^k}}$$

But I don’t know these ☹

→Need: method to estimate free energies from data from multiple thermodynamic states

# MBAR
## Multistate Bennett Acceptance Ratio

- Method to combine data from multiple thermodynamic states to estimate probability distribution at a reference state

Shirts, Michael R., and John D. Chodera. "Statistically optimal analysis of samples from multiple equilibrium states." *The Journal of chemical physics* 129.12 (2008): 124105.

# MBAR

Have:

- $S$ simulations performed at a biased state

$$U^k(x) = U^0(x) + b^k(x), \qquad k \epsilon 1, \dots, S$$

- $N^k$ i.i.d. samples per state, $\sum_k N^k = N$

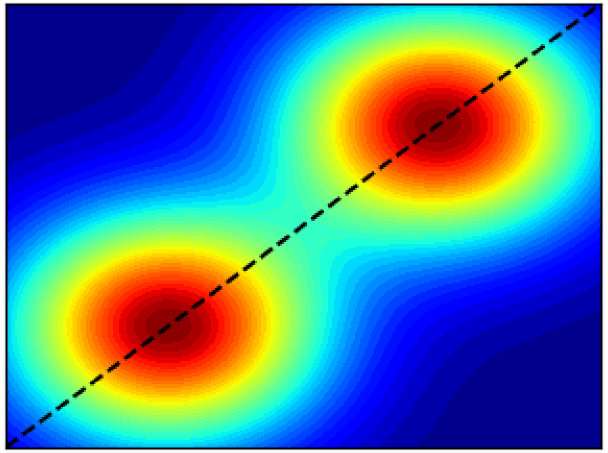
## MBAR

Have:

- $S$ simulations performed at a biased state

$$U^k(x) = U^0(x) + b^k(x), \qquad k \in 1, \dots, S$$

- $N^k$ i.i.d. samples per state, $\sum_k N^k = N$



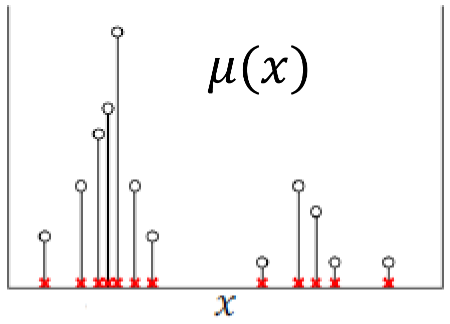
$k$ corresponds to e.g.

- one umbrella (umbrella sampling)
- one temperature (parallel tempering)

## MBAR

- Distribution over samples (point-wise)

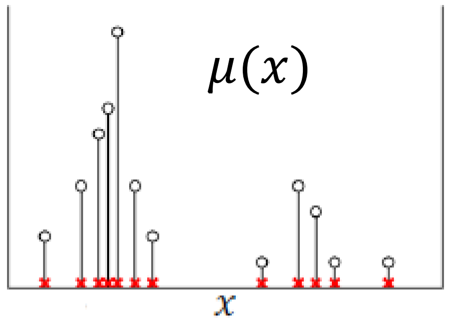
$$\mu(x)$$



# MBAR

- Distribution over samples (point-wise)

$$\mu(x)$$



$\mu(x)$: the contribution of sample $x$ to the Boltzmann distribution

$$\sum_x \mu(x) = 1$$

## MBAR

- Biased distributions

The bias energy

$$\mu^k(x) = \exp[-b^k(x) + f^k]\, \mu(x)$$

- $k \in \{1, \ldots S\}$ thermodynamic state/simulation index

## MBAR

- Biased distributions

The bias energy

$$\mu^k(x) = \exp[-b^k(x) + f^k]\,\mu(x)$$

Unbiased sample weight

- $k \in \{1, \dots S\}$ thermodynamic state/simulation index

# MBAR

- Biased distributions

$$\mu^k(x) = \exp[-b^k(x) + f^k]\,\mu(x)$$

- $k \in \{1, \dots S\}$ thermodynamic state/simulation index

The bias energy

Unbiased sample weight

$f^k = -\beta^k \log Z^k$

The dimension-less free energy of state $k$
→ Ensures normalization of $\mu^k(x)$
→ Unknown!

## MBAR

- Biased distributions

$$\mu^k(x) = \exp[-b^k(x) + f^k]\,\mu(x)$$

The bias energy

Unbiased sample weight

$f^k = -\beta^k \log Z^k$

The dimension-less free energy of state $k$
→ Ensures normalization of $\mu^k(x)$
→ Unknown!

- $k \in \{1, \dots S\}$ thermodynamic state/simulation index

- $\mu^k(x)$ are distributions over samples!

$$\sum_x \mu^k(x) = 1$$

- (or: $\mu^k(x)$ is the statistical sample weight in state $k$)

## MBAR

- Biased distributions

$$\mu^k(x) = \exp[-b^k(x) + f^k]\,\mu(x)$$

The bias energy

Unbiased sample weight

$f^k = -\beta^k \log Z^k$

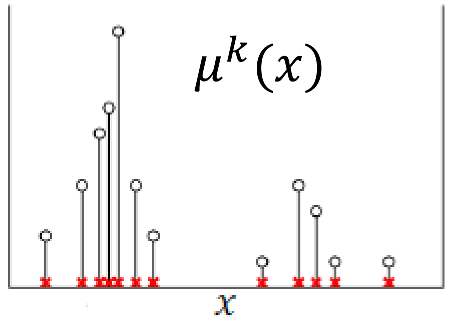
The dimension-less free energy of state $k$
→ Ensures normalization of $\mu^k(x)$
→ Unknown!

- $k \in \{1, \dots S\}$ thermodynamic state/simulation index

- $\mu^k(x)$ are distributions over samples!

$$\sum_x \mu^k(x) = 1$$

- (or: $\mu^k(x)$ is the statistical sample weight in state $k$)



## MBAR

- Biased distributions

$$\mu^k(x) = \exp[-b^k(x) + f^k]\,\mu(x)$$

$$\mu(x) = 1/\sum_{l=1}^{S} N^l \exp[-b^l(x) + f^l]$$

## MBAR

- Biased distributions

$$\mu^k(x) = \frac{\exp[-b^k(x) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x) + f^l]}$$

$$\mu(x) = 1/\sum_{l=1}^{S} N^l \exp[-b^l(x) + f^l]$$

## MBAR

How to find the $f^k$?

- Likelihood of observing all samples:

$$L(x) = \prod_{k=1}^{S} \prod_{x \epsilon X^k} \mu^k(x)$$

## MBAR

How to find the $f^k$?

- Likelihood of observing all samples:

$$L(x \mid f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{x \epsilon X^k} \mu^k(x) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

## MBAR

How to find the $f^k$?

- Likelihood of observing all samples:

$$L(x \mid f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{x \epsilon X^k} \mu^k(x) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

Find $f^k$ that maximize the likelihood of observing all data!

→ = convex optimization problem ☺

## MBAR

How to find the $f^k$?

- Likelihood of observing all samples:

$$L(x \mid f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{x \epsilon X^k} \mu^k(x) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

$f^k$ → Optimization parameters.

## MBAR

How to find the $f^k$?

- Likelihood of observing all samples:

$$L(x \mid f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{x \epsilon X^k} \mu^k(x) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

$f^k$ → Optimization parameters.

Input:

$\exp[-b^l(x_n^k)]$ → Bias coefficients

$N^k$ → Number of samples in simulation k

## MBAR - input

$$L_{MBAR}(f^1, \ldots, f^S) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

$$\exp[-b^l(x_n^k)]$$

The $n$-th coordinate in the trajectory
Of samples taken at state $k$
Evaluated at the state $l$ Hamiltonian

## MBAR - input

$$L_{MBAR}(f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \frac{\exp[-b^k(x_n^k) + f^k]}{\sum_{l=1}^{S} N^l \exp[-b^l(x_n^k) + f^l]}$$

$$\exp[-b^l(x_n^k)]$$

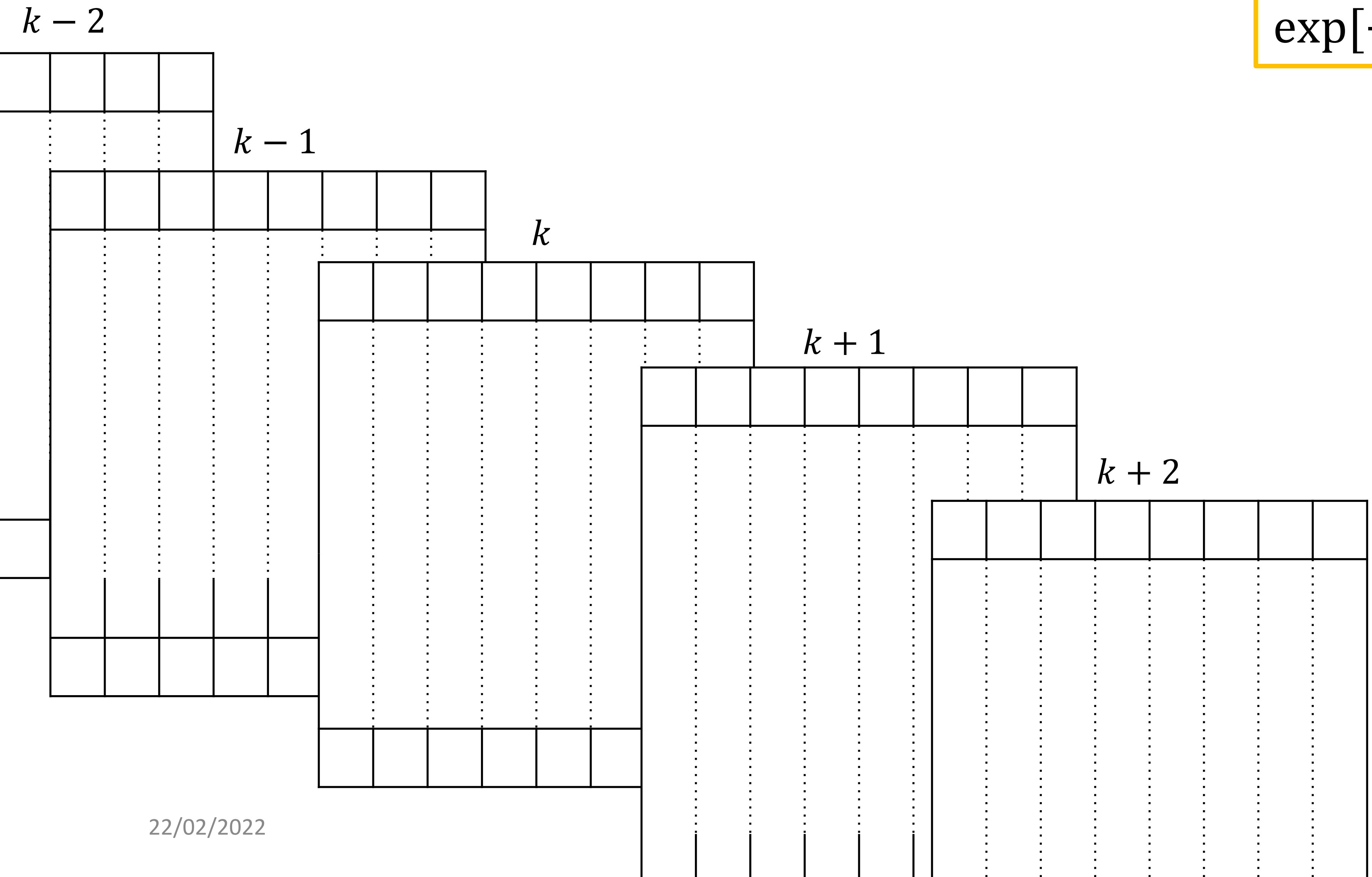
The $n$-th coordinate in the trajectory
Of samples taken during simulation $k$
Evaluated at the bias potential of simulation $l$

```
bias_matrices =  [np.ndarray([[0.0, 0.43, 0.28, ...], [0.0, 1.28, 0.32, ...], ...]),
                  np.ndarray([[0.0, 0.23, 0.86, ...], [0.5, 0.50, 1.02, ...], ...]), ...]
```

3D data structure
→ List of $S$ nd-arrays
→ $k$-th array has shape $(N^k \times S)$

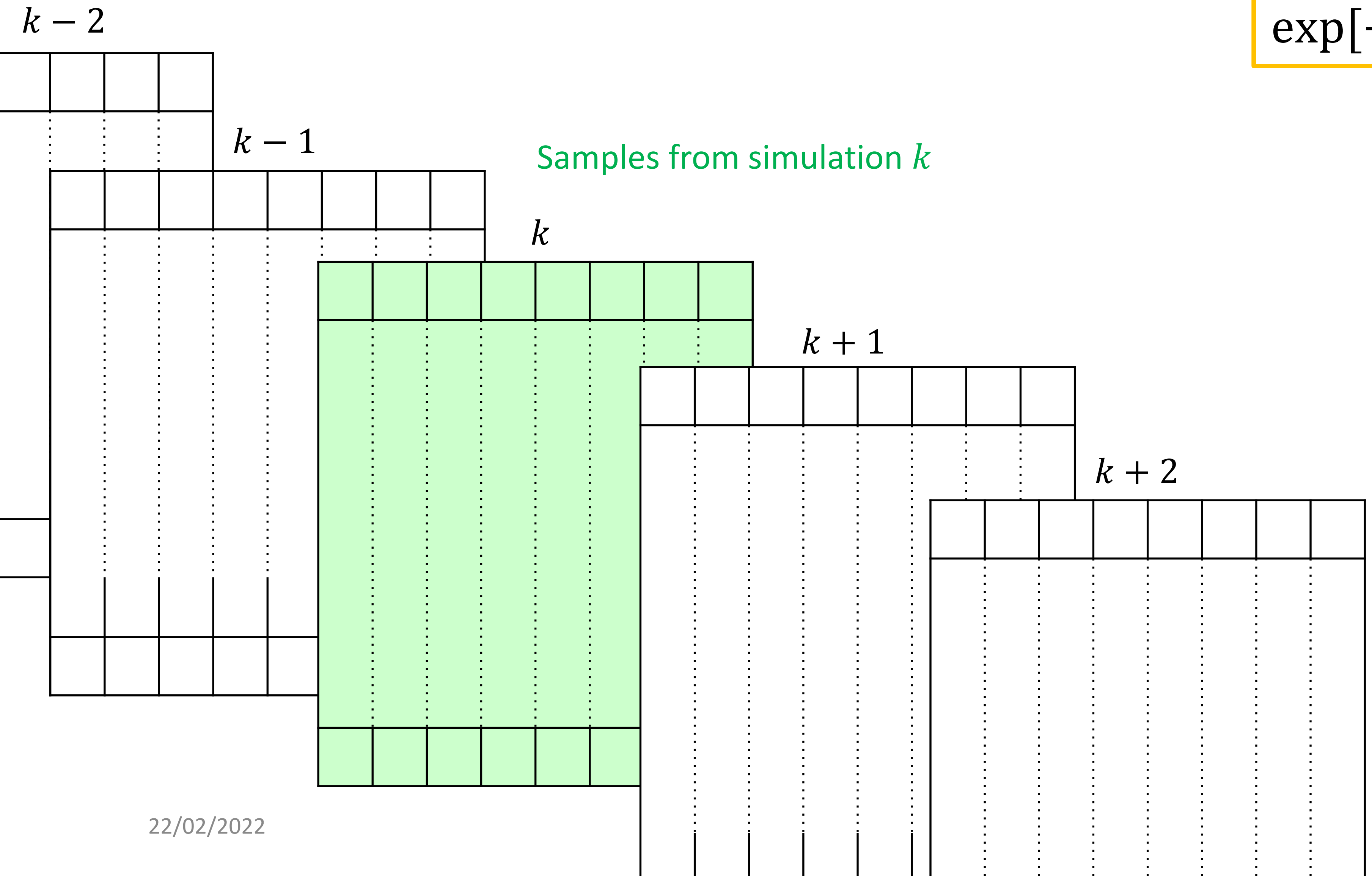
## The bias matrices

$\boxed{\exp[-b^{l}(x_{n}^{k})]}$



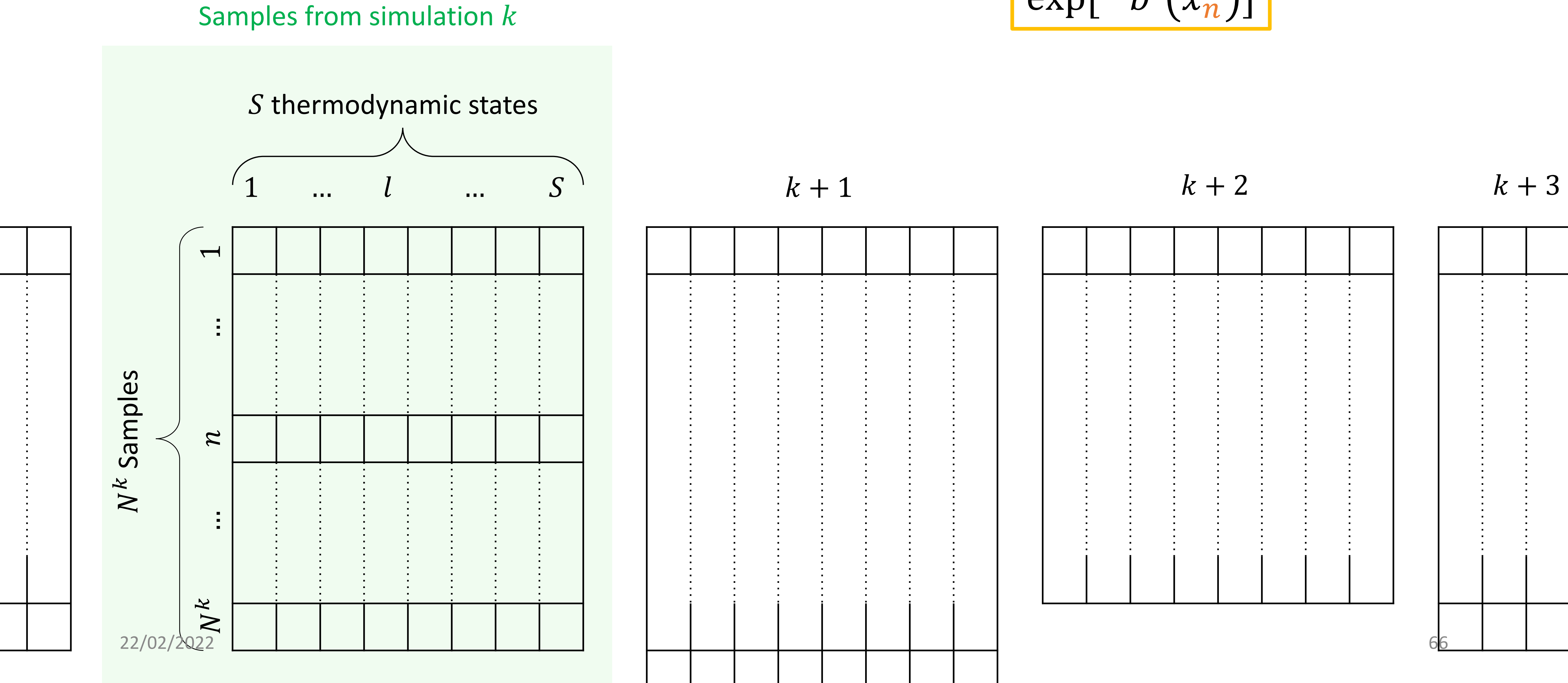
# The bias matrices

$$\boxed{\exp[-b^{l}(x_{n}^{k})]}$$



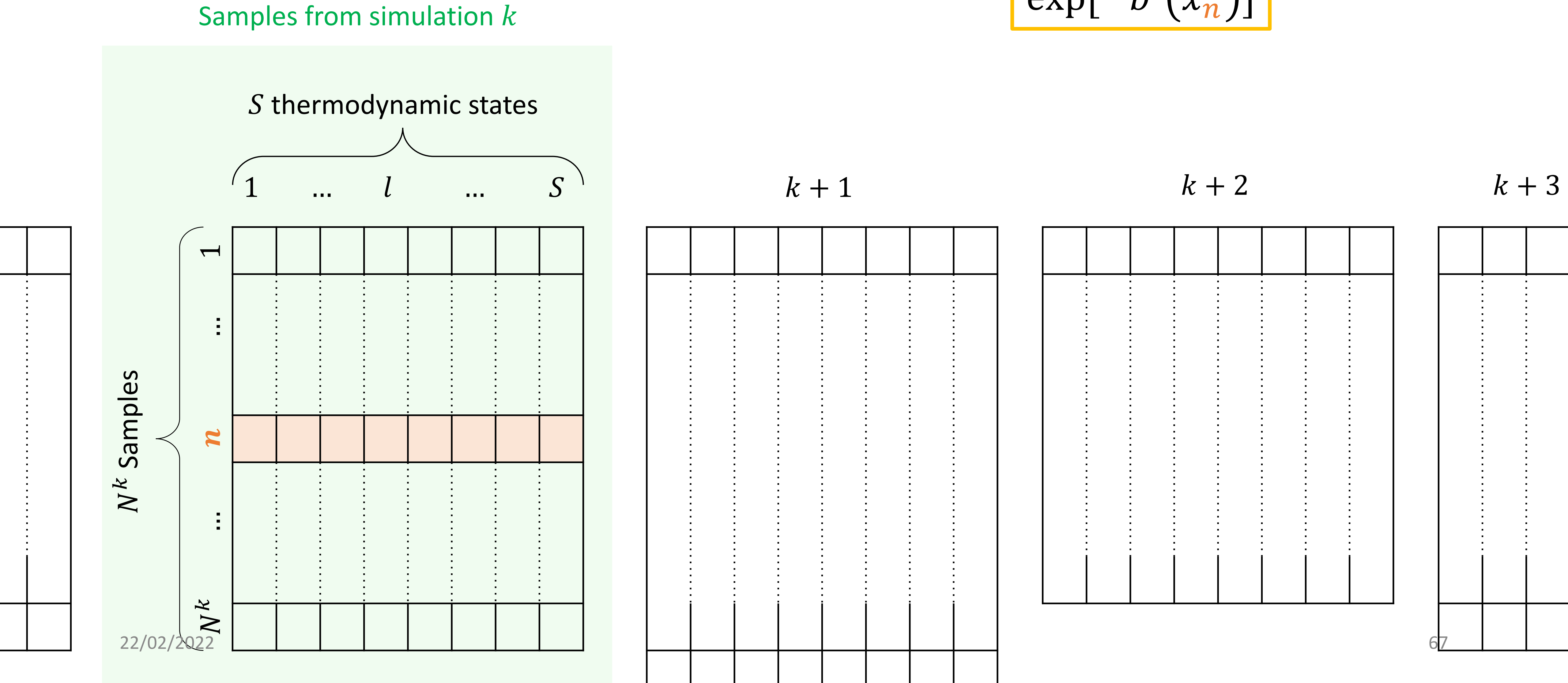
# The bias matrices

$\boxed{\exp[-b^{l}(x_{n}^{k})]}$

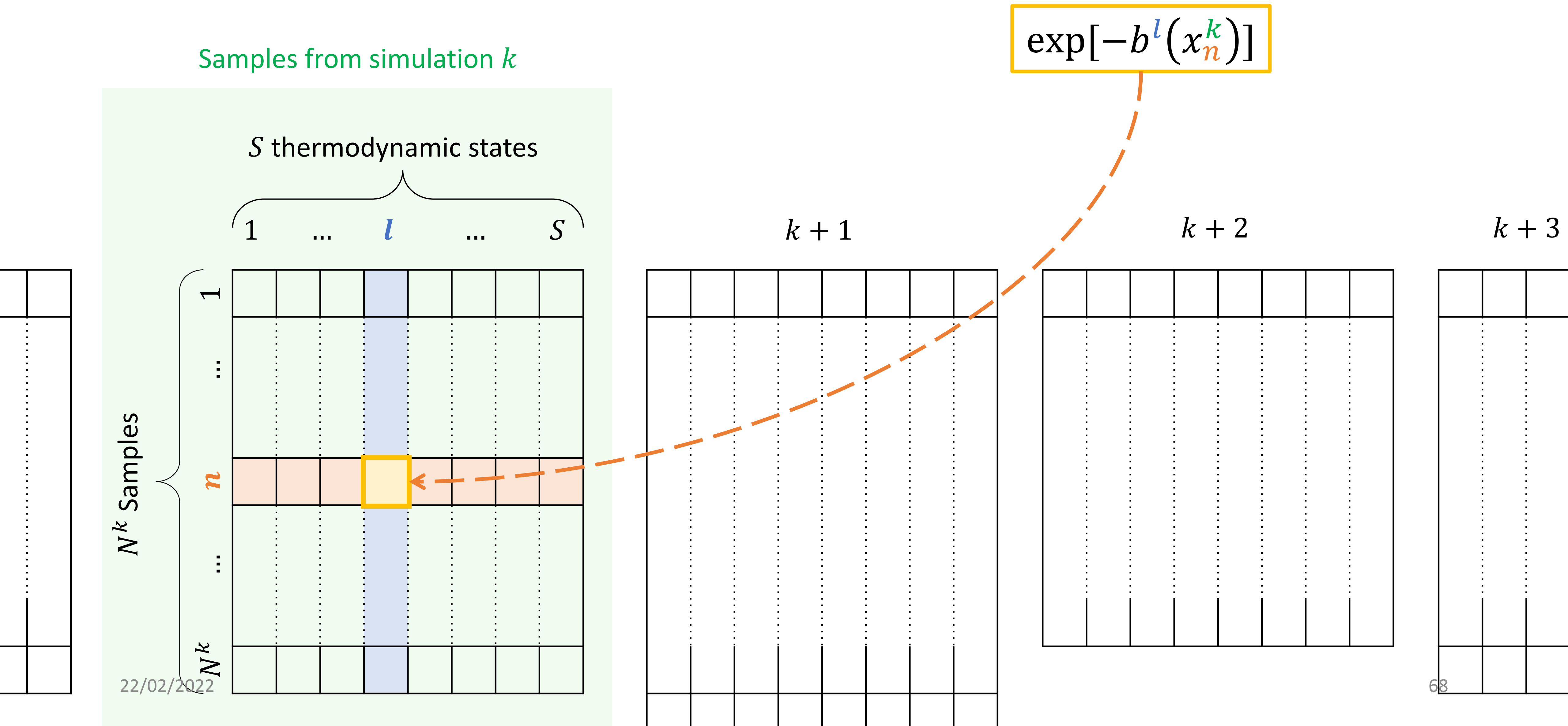


# The bias matrices

$$\boxed{\exp[-b^{l}(x_{n}^{k})]}$$



# The bias matrices

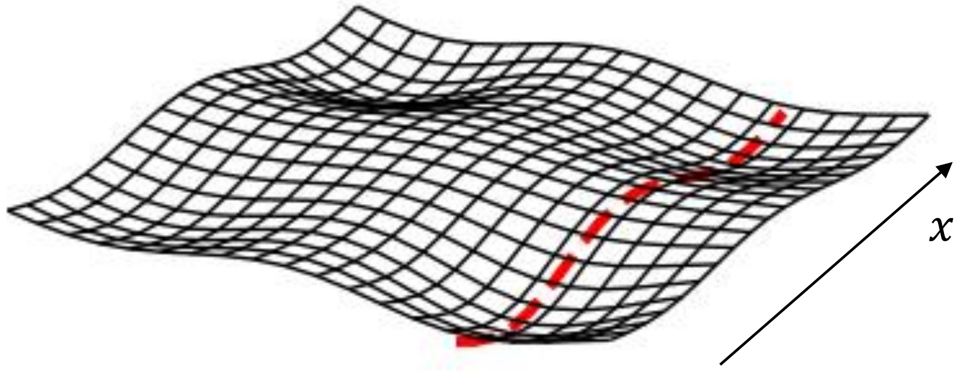


## MBAR in PyEMMA

- In .thermo package: http://www.emma-project.org/latest/api/generated/thermo-api/pyemma.thermo.mbar.html

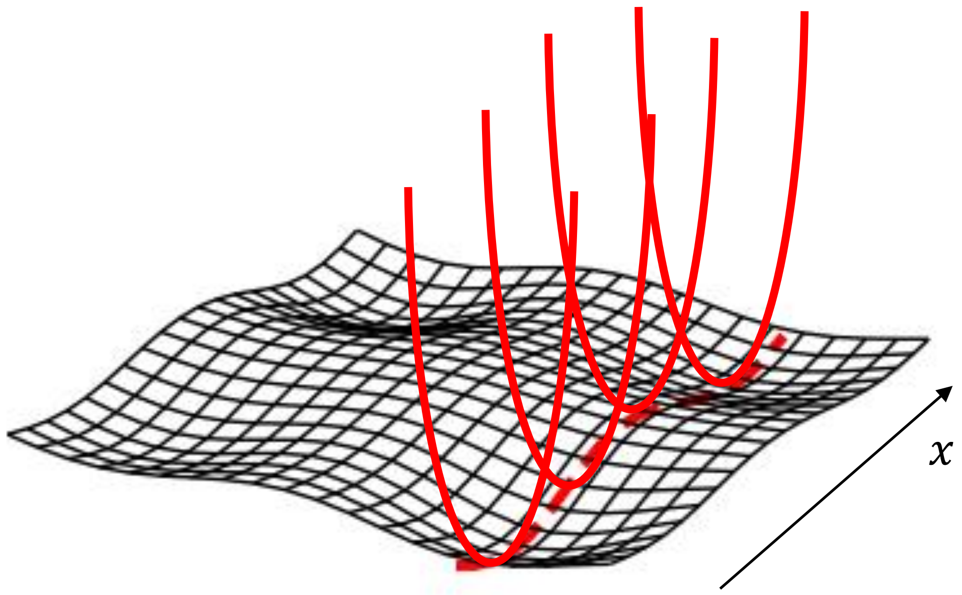
## What does this have to do with MSMs?

- MBAR assumes samples are drawn from a global equilibrium
  - →inefficient, samples need to be spaced far apart in time
  - →slow degrees of freedom can introduce a systematic error



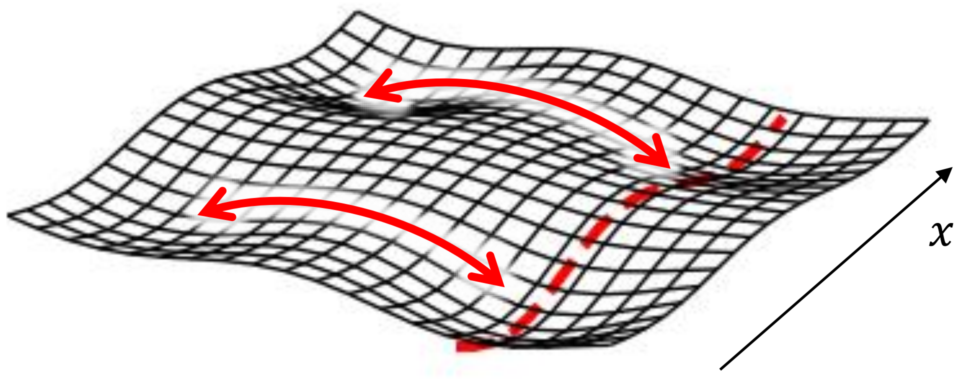
# What does this have to do with MSMs?

- MBAR assumes samples are drawn from a global equilibrium
  - →inefficient, samples need to be spaced far apart in time
  - →slow degrees of freedom can introduce a systematic error



# What does this have to do with MSMs?

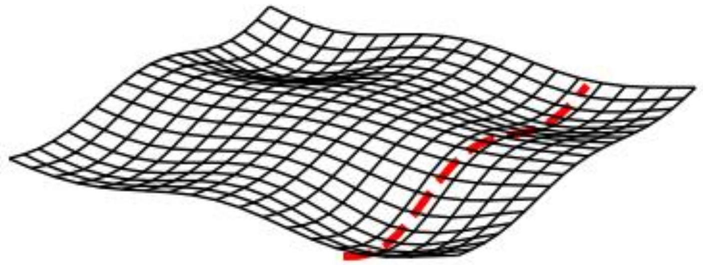
- MBAR assumes samples are drawn from a global equilibrium
  - →inefficient, samples need to be spaced far apart in time
  - →slow degrees of freedom can introduce a systematic error



# What does this have to do with MSMs?

- MBAR assumes samples are drawn from a global equilibrium
  - →inefficient, samples need to be spaced far apart in time
  - →slow degrees of freedom can introduce a systematic error

Idea: combine MBAR with Markov state Models



# MEMM

- Each thermodynamic state is governed by a Markov state model
- All MSMs together form a Multi-Ensemble Markov Model (MEMM)

$$\left.\begin{array}{l}
P_{ij}^{1} = \begin{bmatrix} p_{1m}^{1} & \cdots & p_{1m}^{1} \\ \vdots & \ddots & \vdots \\ p_{m1}^{1} & \cdots & p_{mm}^{1} \end{bmatrix} \\
P_{ij}^{2} = \begin{bmatrix} p_{1m}^{2} & \cdots & p_{1m}^{2} \\ \vdots & \ddots & \vdots \\ p_{m1}^{2} & \cdots & p_{mm}^{2} \end{bmatrix} \\
P_{ij}^{3} = \begin{bmatrix} p_{1m}^{3} & \cdots & p_{1m}^{3} \\ \vdots & \ddots & \vdots \\ p_{m1}^{3} & \cdots & p_{mm}^{3} \end{bmatrix} \\
P_{ij}^{4} = \begin{bmatrix} p_{1m}^{4} & \cdots & p_{1m}^{4} \\ \vdots & \ddots & \vdots \end{bmatrix}
\end{array}\right\} \boldsymbol{P}_{ij}$$

# MEMM

- Each thermodynamic state is governed by a Markov state model
- All MSMs together form a Multi-Ensemble Markov Model (MEMM)

$$P_{ij}^{1} = \begin{bmatrix} p_{1m}^{1} & \cdots & p_{1m}^{1} \\ \vdots & \ddots & \vdots \\ p_{m1}^{1} & \cdots & p_{mm}^{1} \end{bmatrix}$$

$$P_{ij}^{2} = \begin{bmatrix} p_{1m}^{2} & \cdots & p_{1m}^{2} \\ \vdots & \ddots & \vdots \\ p_{m1}^{2} & \cdots & p_{mm}^{2} \end{bmatrix}$$

$$P_{ij}^{3} = \begin{bmatrix} p_{1m}^{3} & \cdots & p_{1m}^{3} \\ \vdots & \ddots & \vdots \\ p_{m1}^{3} & \cdots & p_{mm}^{3} \end{bmatrix}$$

$$P_{ij}^{4} = \begin{bmatrix} p_{1m}^{4} & \cdots & p_{1m}^{4} \\ \vdots & \ddots & \vdots \end{bmatrix}$$

$$\boldsymbol{P}_{ij}$$

$p_{ij}^{k}$:
in thermodynamic state (simulation) $k$
transition probability from Markov state $i$ to Markov state $j$

## MEMM

- Each thermodynamic state is governed by a Markov state model
- All MSMs together form a Multi-Ensemble Markov Model (MEMM)

$$P^1_{ij} = \begin{bmatrix} p^1_{1m} & \cdots & p^1_{1m} \\ \vdots & \ddots & \vdots \\ p^1_{m1} & \cdots & p^1_{mm} \end{bmatrix}$$

$$P^2_{ij} = \begin{bmatrix} p^2_{1m} & \cdots & p^2_{1m} \\ \vdots & \ddots & \vdots \\ p^2_{m1} & \cdots & p^2_{mm} \end{bmatrix}$$

$$P^3_{ij} = \begin{bmatrix} p^3_{1m} & \cdots & p^3_{1m} \\ \vdots & \ddots & \vdots \\ p^3_{m1} & \cdots & p^3_{mm} \end{bmatrix}$$

$$P^4_{ij} = \begin{bmatrix} p^4_{1m} & \cdots & p^4_{1m} \\ \vdots & \ddots & \vdots \end{bmatrix}$$

$\boldsymbol{P}_{ij}$

Likelihood of MSM:

$$L^k_{MSM} = \prod_{i,j=1}^{B} (p^k_{ij})^{c^k_{ij}}$$

$p^k_{ij}$:
in thermodynamic state (simulation) $k$
transition probability from Markov state $i$ to Markov state $j$

## MEMM

How are the individual MSMs in the MEMM coupled?

## MEMM

How are the individual MSMs in the MEMM coupled?

- $\pi_i$ are the stationary distribution of $P_{ij}^k$

## MEMM

How are the individual MSMs in the MEMM coupled?

- $\pi_i$ are the stationary distribution of $P_{ij}^k$
- Distributions are all related to a reference distribution through Boltzmann reweighting:

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

# MEMM

How are the individual MSMs in the MEMM coupled?

- $\pi_i$ are the stationary distribution of $P_{ij}^k$
- Distributions are all related to a reference distribution through Boltzmann reweighting:

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

- Detailed balance: $\pi_i^k p_{ij}^k = \pi_j^k p_{ji}^k$

## TRAM
## Transition-based Reweighting Analysis Method

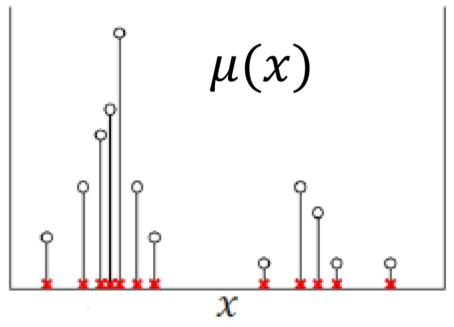
Combines discrete MEMM with continuous MBAR

- MEMM: transition probabilities (discrete)
- MBAR: sample weights $\mu(x)$ (continuous)

## TRAM

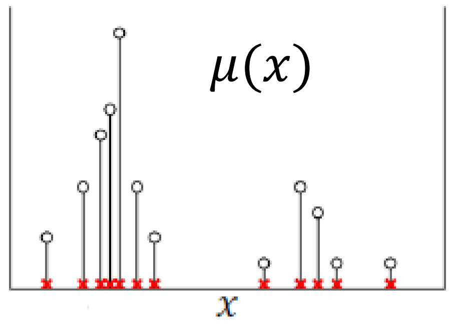
Combines discrete MEMM with continuous MBAR

- Continuous global reference distribution: $\mu(x)$



# TRAM

Combines discrete MEMM with continuous MBAR

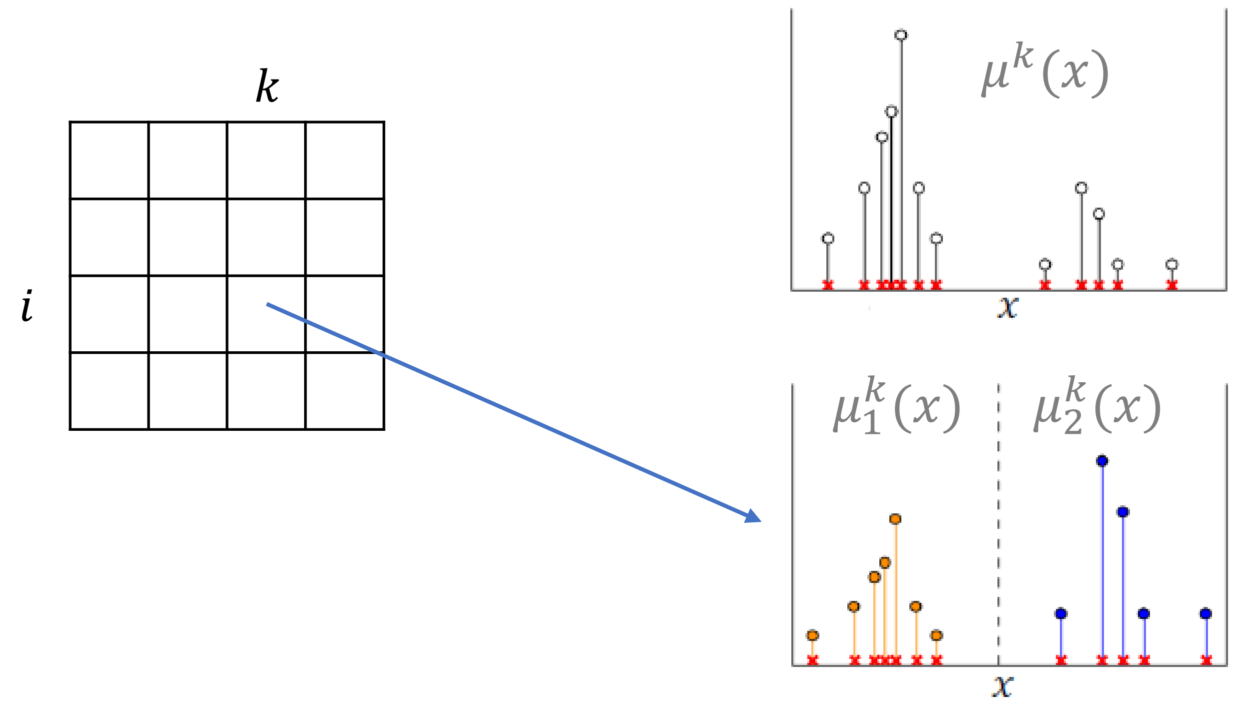


- Continuous global reference distribution: $\mu(x)$

- Markov states are governed by a local equilibrium (LEQ) distribution

→ $\mu_i^k(x)$ is the local equilibrium (LEQ) distribution of Markov state $i$ in thermodynamic state $k$

# TRAM



→ $\mu_i^k(x)$ is the local equilibrium (LEQ) distribution of Markov state $i$ in thermodynamic state $k$

## TRAM

- LEQ distributions are all related to $\mu(x)$ through Boltzmann reweighting:

$$\mu_i^k(x) = \begin{cases} \exp[f_i^k - b^k(x)]\,\mu(x) & x \in i \\ 0 & otherwise \end{cases}$$

## TRAM

- LEQ distributions are all related to $\mu(x)$ through Boltzmann reweighting:

$$\mu_i^k(x) = \begin{cases} \exp[f_i^k - b^k(x)]\,\mu(x) & x \in i \\ 0 & otherwise \end{cases}$$

- Detailed balance

$$\pi_i^k p_{ij}^k = \pi_j^k p_{ji}^k$$

## TRAM

- LEQ distributions are all related to $\mu(x)$ through Boltzmann reweighting:

$$\mu_i^k(x) = \begin{cases} \exp[f_i^k - b^k(x)]\,\mu(x) & x \in i \\ 0 & otherwise \end{cases}$$

- Detailed balance

$$Z_i^k p_{ij}^k = Z_j^k p_{ji}^k$$

## TRAM

- LEQ distributions are all related to $\mu(x)$ through Boltzmann reweighting:

$$\mu_i^k(x) = \begin{cases} \exp[f_i^k - b^k(x)]\,\mu(x) & x \in i \\ 0 & otherwise \end{cases}$$

- Detailed balance

$$e^{-f_i^k} p_{ij}^k = e^{-f_j^k} p_{ji}^k$$

# TRAM

- LEQ distributions are all related to $\mu(x)$ through Boltzmann reweighting:

$$\mu_i^k(x) = \begin{cases} \exp[f_i^k - b^k(x)]\,\mu(x) & x \in i \\ 0 & otherwise \end{cases}$$

- Detailed balance

$$e^{-f_i^k} p_{ij}^k = e^{-f_j^k} p_{ji}^k$$

TRAM: estimate $\boldsymbol{f_i^k}$ **(and** $\boldsymbol{p_{ij}^k}$ **and** $\boldsymbol{\mu(x)}$**)**

# TRAM - likelihood

- Discrete transition probabilities (MEMM) × continuous sample weights (LEQ)

Discrete state sequence

Continuous configurations within state

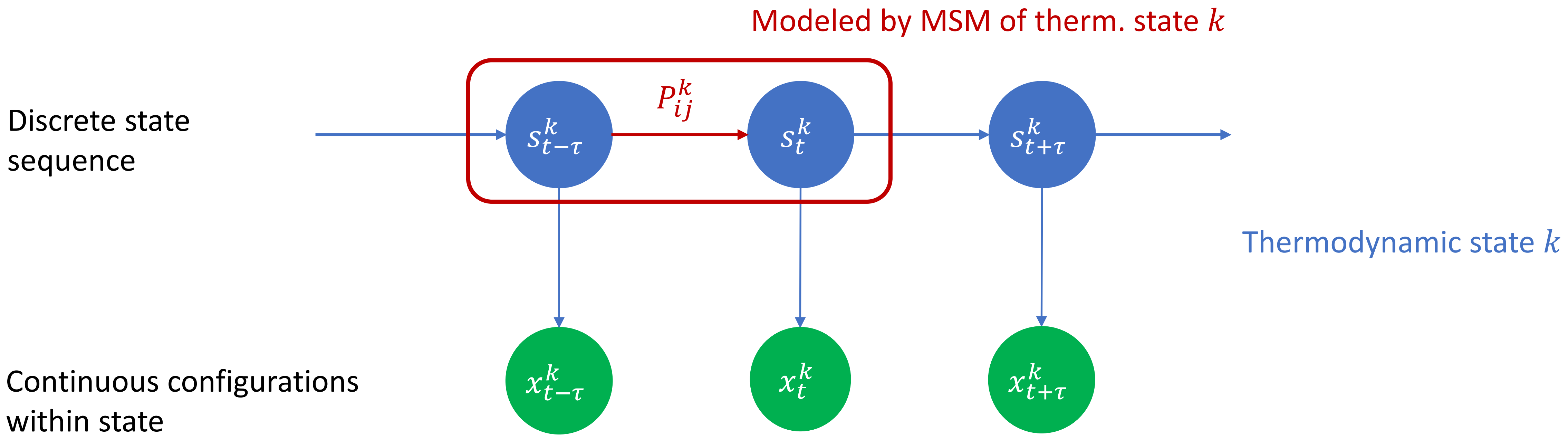
$$\begin{tikzcd}
{} \arrow[r] & s_{t-\tau}^{k} \arrow[r] \arrow[d] & s_{t}^{k} \arrow[r] \arrow[d] & s_{t+\tau}^{k} \arrow[r] \arrow[d] & {} \\
& x_{t-\tau}^{k} & x_{t}^{k} & x_{t+\tau}^{k} &
\end{tikzcd}$$

Thermodynamic state $k$

Wu, Hao, et al. "Multiensemble Markov models of molecular thermodynamics and kinetics." *Proceedings of the National Academy of Sciences* 113.23 (2016): E3221-E3230.

# TRAM - likelihood

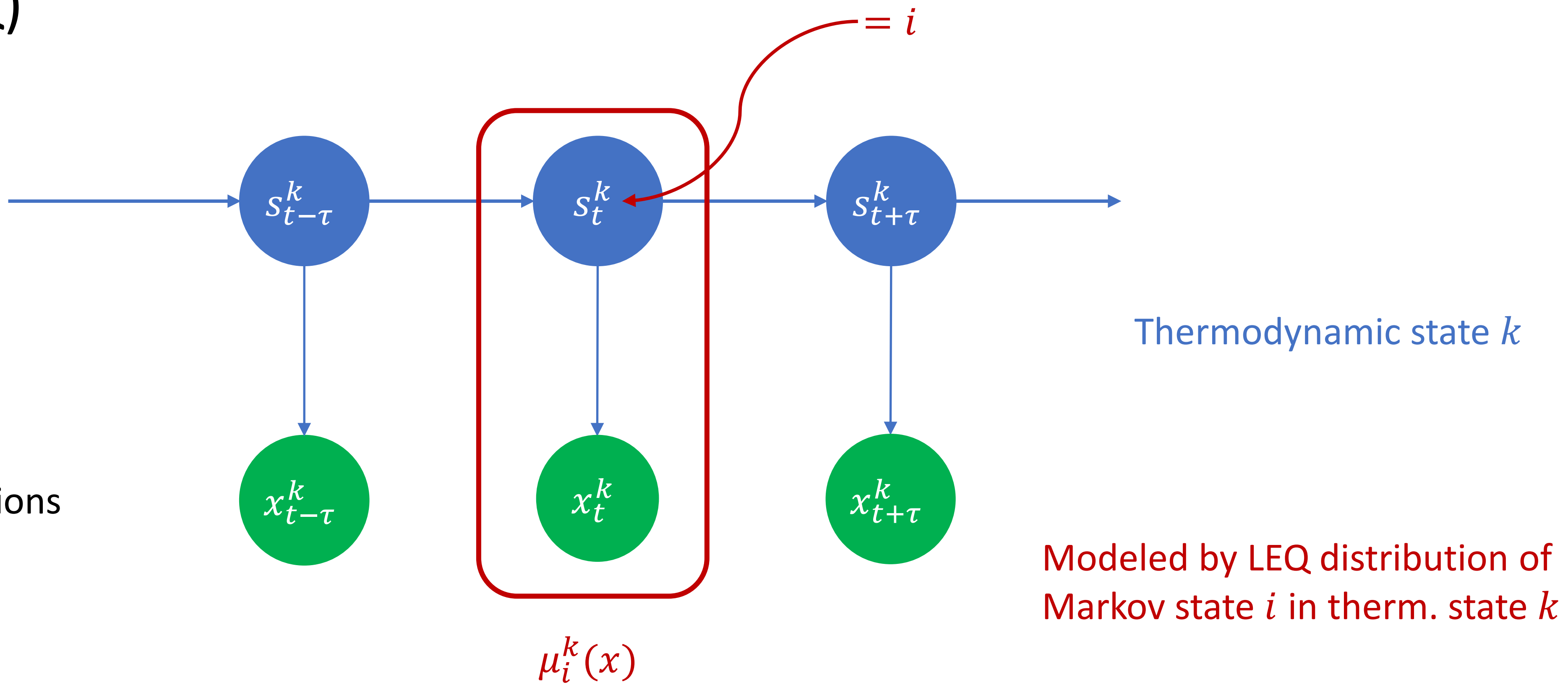
- Discrete transition probabilities (MEMM) × continuous sample weights (LEQ)



Wu, Hao, et al. "Multiensemble Markov models of molecular thermodynamics and kinetics." *Proceedings of the National Academy of Sciences* 113.23 (2016): E3221-E3230.

# TRAM - likelihood

- Discrete transition probabilities (MEMM) × continuous sample weights (LEQ)



Wu, Hao, et al. "Multiensemble Markov models of molecular thermodynamics and kinetics." *Proceedings of the National Academy of Sciences* 113.23 (2016): E3221-E3230.

# TRAM Likelihood

- Combines discrete transition probabilities (MEMM) with continuous sample weights (LEQ)

Likelihood of observing transitions:

$$L_{MSM}^{k} = \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}}$$

$\times$

Likelihood of observing all samples within the states:

$$L_{LEQ}^{k} = \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \mu_i^k(x)$$

$$L_{TRAM} = \prod_{k=1}^{S} \left( \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}} \right) \left( \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \mu_i^k(x) \right)$$

# TRAM Likelihood

Maximize likelihood

$$L_{TRAM} = \prod_{k=1}^{S} \left( \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}} \right) \left( \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \exp[f_i^k - b^k(x)]\, \mu(x) \right)$$

# TRAM Likelihood

Maximize likelihood

$$L_{TRAM} = \prod_{k=1}^{S} \left( \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}} \right) \left( \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \exp[f_i^k - b^k(x)]\, \mu(x) \right)$$

Under constraints:

- Detailed balance

$$e^{-f_i^k} p_{ij}^{k} = e^{-f_j^k} p_{ji}^{k}$$

- Normalized probabilities

$$\sum\nolimits_{x} \mu(x) = 1$$

$$\sum\nolimits_{j} p_{ij}^{k} = 1$$

## TRAM Likelihood

$$L_{TRAM} = \prod_{k=1}^{S} \left( \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}} \right) \left( \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \exp[f_i^k - b^k(x)] \, \mu(x) \right)$$

Input:

$c_{ij}^{k}$ → transition counts

$b^k(x)$ → reduced bias energies

## TRAM Likelihood

$$L_{TRAM} = \prod_{k=1}^{S} \left( \prod_{i,j=1}^{M} (p_{ij}^{k})^{c_{ij}^{k}} \right) \left( \prod_{i=1}^{M} \prod_{x \epsilon X_i^k} \exp[f_i^k - b^k(x)]\, \mu(x) \right)$$

Input:

$c_{ij}^{k}$ → transition counts

$b^k(x)$ → reduced bias energies

Unbiased distribution

$$\mu(x) = 1 / \sum_{l=1}^{S} R_{i(x)}^{l} \exp[\boxed{-b^l(x)} + f_{i(x)}^{l}]$$

Need full 3D bias matrix

# TRAM Likelihood

$$L_{TRAM} = \prod_{k=1}^{S}\left(\prod_{i,j=1}^{M}(p_{ij}^{k})^{c_{ij}^{k}}\right)\left(\prod_{i=1}^{M}\prod_{x \epsilon X_i^k} \exp[f_i^k - b^k(x)]\,\mu(x)\right)$$

Samples from simulation $k$

Input:

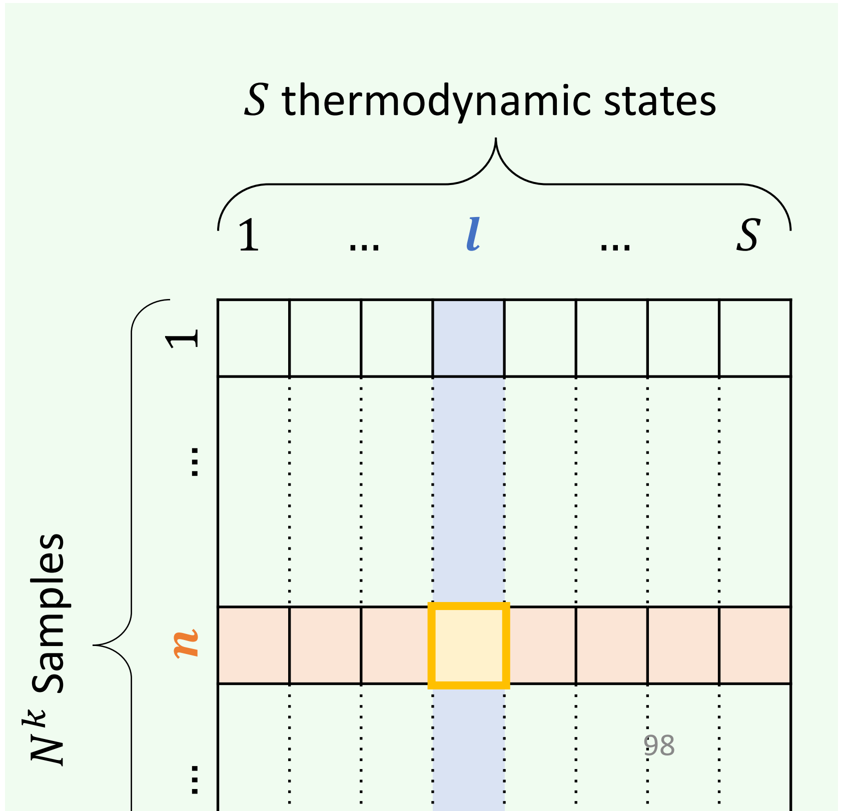
$c_{ij}^{k}$ → transition counts

$b^k(x)$ → reduced bias energies

Unbiased distribution

$$\mu(x) = 1/\sum_{l=1}^{S} R_{i(x)}^{l} \exp[\boxed{-b^l(x)} + f_{i(x)}^{l}]$$

Need full 3D bias matrix



# TRAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]
```

List of $S$ arrays, Array $k$ of length $N^k$

```
bias_matrices =  [np.ndarray([[0.0, 0.43, 0.28, ...], [0.0, 1.28, 0.32, ...], ...]),
                  np.ndarray([[0.0, 0.23, 0.86, ...], [0.5, 0.50, 1.02, ... ],...]), ...]
```

List of $S$ ndarrays
$k$-th ndarray has shape $(N^k \times S)$

$$\exp[-b^l(x_n^k)]$$

The $n$-th coordinate in the trajectory
Of samples taken during simulation $k$
Evaluated at the bias potential of simulation $l$

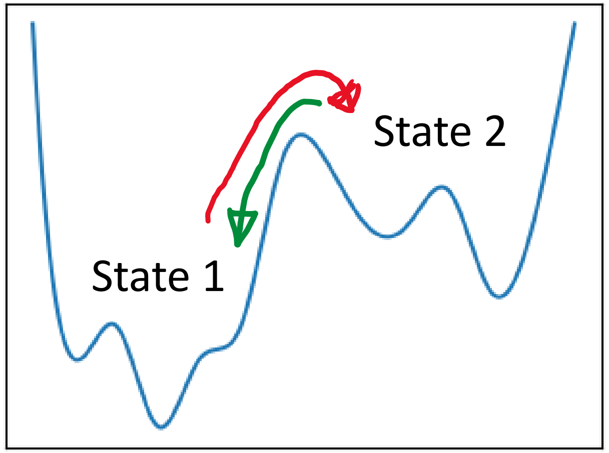
## Advantages of using TRAM

- Better estimation of free energies along the unbiased (orthogonal) degrees of freedom.
- System does not need to be equilibrated to global equilibrium
- Smaller de-correlation time (simulation time until one gets a new uncorrelated frame) $\rightarrow$ more efficient usage of the data.

# Advantages of using TRAM

Detailed balance → enhanced sampling of kinetics

$$e^{-f_i^k} p_{ij}^k = e^{-f_j^k} p_{ji}^k$$



$T_{12}$ is a rare event

$T_{21}$ can be simulated

Detailed balance:

$$e^{-f_2^k} P_{21}^k = e^{-f_2^k} P_{12}^k$$

If we know $e^{-f_2^k}$, $P_{21}^k$, and $e^{-f_2^k}$, we don't have to simulate $P_{12}^k$

## TRAM

- In PyEMMA:

http://www.emma-project.org/latest/api/generated/thermo-api/pyemma.thermo.tram.html

- (New) in Deeptime:

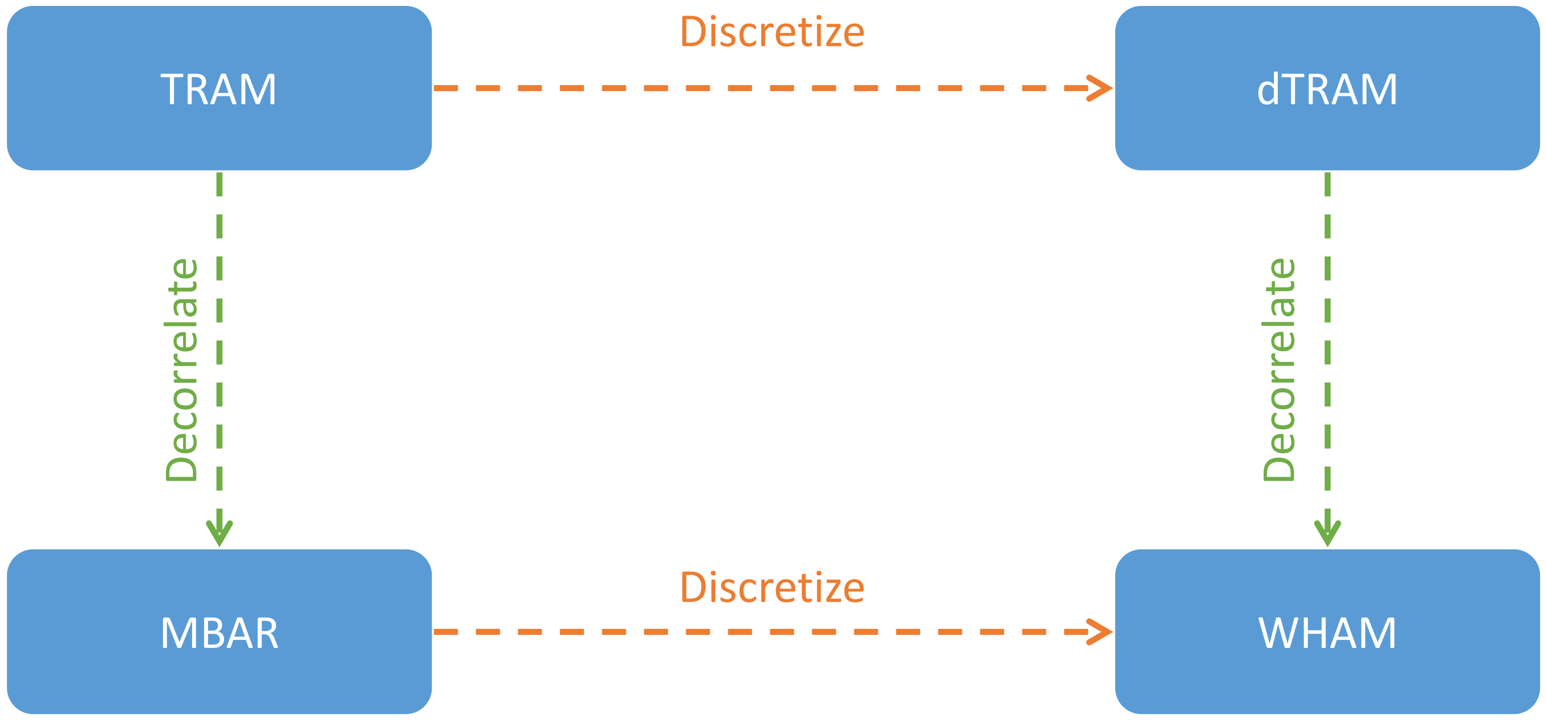
https://deeptime-ml.github.io/latest/api/generated/deeptime.markov.msm.TRAM.html

# TRAM – a simple notebook example

## Further reading

- Shirts, Michael R., and John D. Chodera. "Statistically optimal analysis of samples from multiple equilibrium states." *The Journal of chemical physics* 129.12 (2008): 124105.
- Wu, Hao, et al. "Multiensemble Markov models of molecular thermodynamics and kinetics." *Proceedings of the National Academy of Sciences* 113.23 (2016): E3221-E3230.

# WHAM/MBAR/dTRAM/TRAM



*Arrows represent operations on the data*

# WHAM
# Weighted Histogram Analysis Method

- Method to combine data from multiple thermodynamic states to estimate probability distribution at a reference state

- Discrete (hence: histogram)

Ferrenberg, Alan M., and Robert H. Swendsen. "Optimized monte carlo data analysis." *Computers in Physics* 3.5 (1989): 101-104

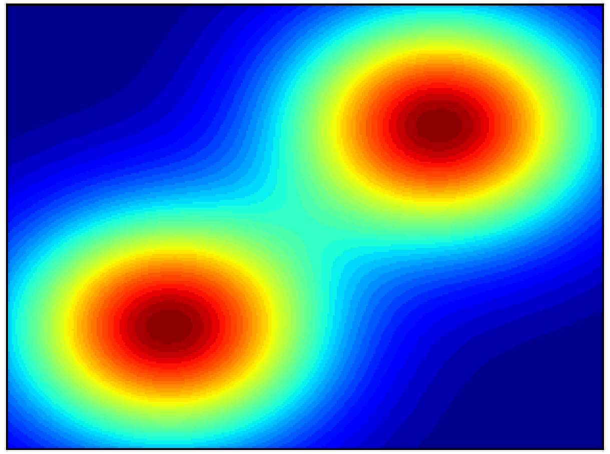
# Weighted Histogram Analysis Method

Have:

- $S$ simulations performed at a biased state

$$U^k(x) = U^0(x) + b^k(x), k \epsilon 1, \ldots, S$$

- $N^k$ i.i.d. samples per state, $\sum_k N^k = N$



## Weighted Histogram Analysis Method

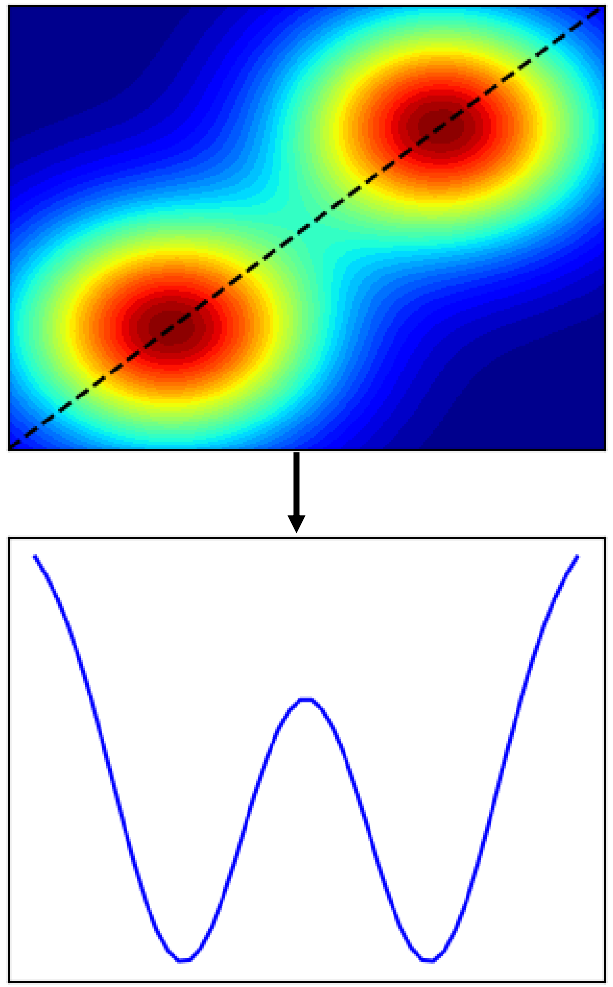
Have:

- $S$ simulations performed at a biased state

$$U^k(x) = U^0(x) + b^k(x), k \epsilon 1, \dots, S$$

- $N^k$ i.i.d. samples per state, $\sum_k N^k = N$
- Samples are discretized w.r.t. the reaction coordinate

Umbrella sampling: umbrella's spaced along the reaction coordinate

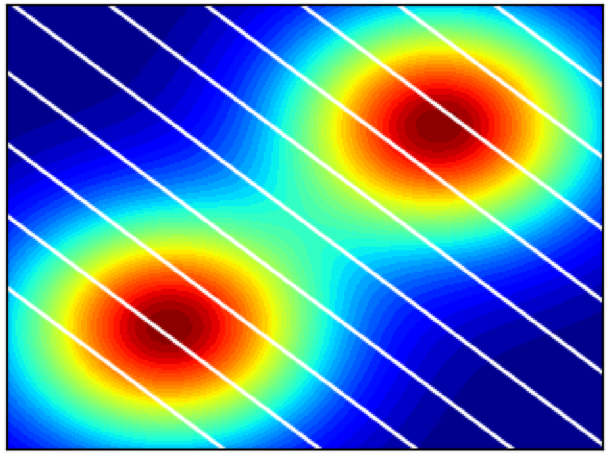


# Weighted Histogram Analysis Method

- $S$ simulations
- Samples are discretized into $B$ bins

- Each bin defined by bias coefficient

$$c_i^k = \exp[-b^k(x_i)]$$



Notation:

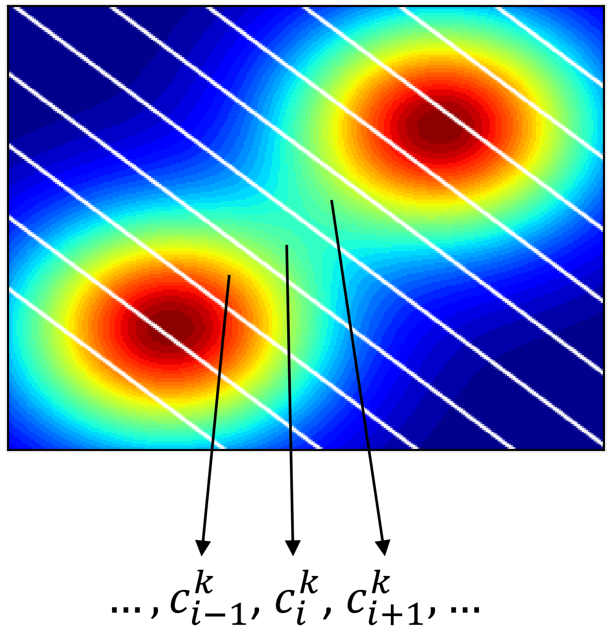
- $k$: therm state (superscript)
- $i$: conformational state/bin (subscript

# Weighted Histogram Analysis Method

- $S$ simulations
- Samples are discretized into $B$ bins

- Each bin defined by bias coefficient

$$c_i^k = \exp[-b^k(x_i)]$$



Notation:

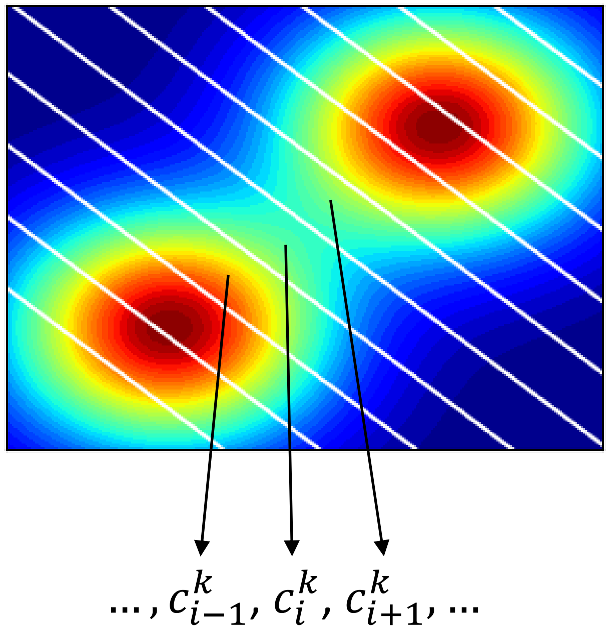
- $k$: therm state (superscript)
- $i$: conformational state/bin (subscript)

# Weighted Histogram Analysis Method

- $S$ simulations
- Samples are discretized into $B$ bins

- Each bin defined by bias coefficient

$$c_i^k = \exp[-b^k(x_i)]$$



Store counts for bin $i$ in simulation $k$:

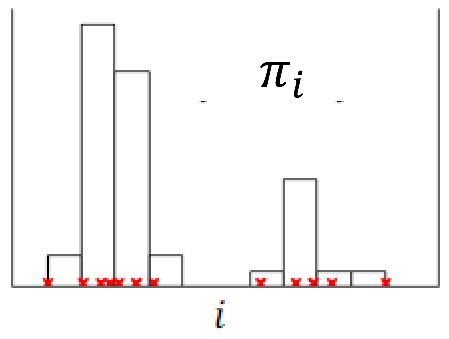
$$N^k = \sum_i n_i^k$$

# Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution



# Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

## Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

Normalizing constant of ensemble $k$

$$(\hat{Z}^k)^{-1} = \sum_{i=1}^{B} c_i^k \pi_i$$

## Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

Normalizing constant of ensemble $k$

$$(\hat{Z}^k)^{-1} = \sum_{i=1}^{B} c_i^k \pi_i$$

= Boltzmann reweighting!

# Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

Normalizing constant of ensemble $k$

$$(\hat{Z}^k)^{-1} = \sum_{i=1}^{B} c_i^k \pi_i$$

Likelihood of observing bin counts:

$$L(n_i^k | \pi_i^k) = \prod_{k=1}^{S} \prod_{i=1}^{B} (\pi_i^k)^{n_i^k}$$

# Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

Normalizing constant of ensemble $k$

$$(\hat{Z}^k)^{-1} = \sum_{i=1}^{B} c_i^k \pi_i$$

Likelihood of observing bin counts:

$$L(n_i^k) = \prod_{k=1}^{S} \prod_{i=1}^{B} (\pi_i^k)^{n_i^k}$$

$$= \prod_{k=1}^{S} \prod_{i=1}^{B} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_j} \right)^{n_i^k}$$

# Weighted Histogram Analysis Method

Probability of bin $i$ in simulation $k$

$$\pi_i^k = \hat{Z}^k c_i^k \pi_i$$

Unbiased distribution

Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

Normalizing constant of ensemble $k$

$$(\hat{Z}^k)^{-1} = \sum_{i=1}^{B} c_i^k \pi_i$$

Likelihood of observing bin counts:

$$L(n_i^k) = \prod_{k=1}^{S} \prod_{i=1}^{B} (\pi_i^k)^{n_i^k}$$

$$= \prod_{k=1}^{S} \prod_{i=1}^{B} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i} \right)^{n_i^k}$$

Solve for $\pi_i$ !

## Weighted Histogram Analysis Method

$$L(n_i^k|\pi_i) = \prod_{k=1}^{S}\prod_{i=1}^{B}\left(\frac{c_i^k\pi_i}{\sum_{j=1}^{B} c_j^k\pi_j}\right)^{n_i^k}$$

$\pi_i \rightarrow$ Optimization parameters.

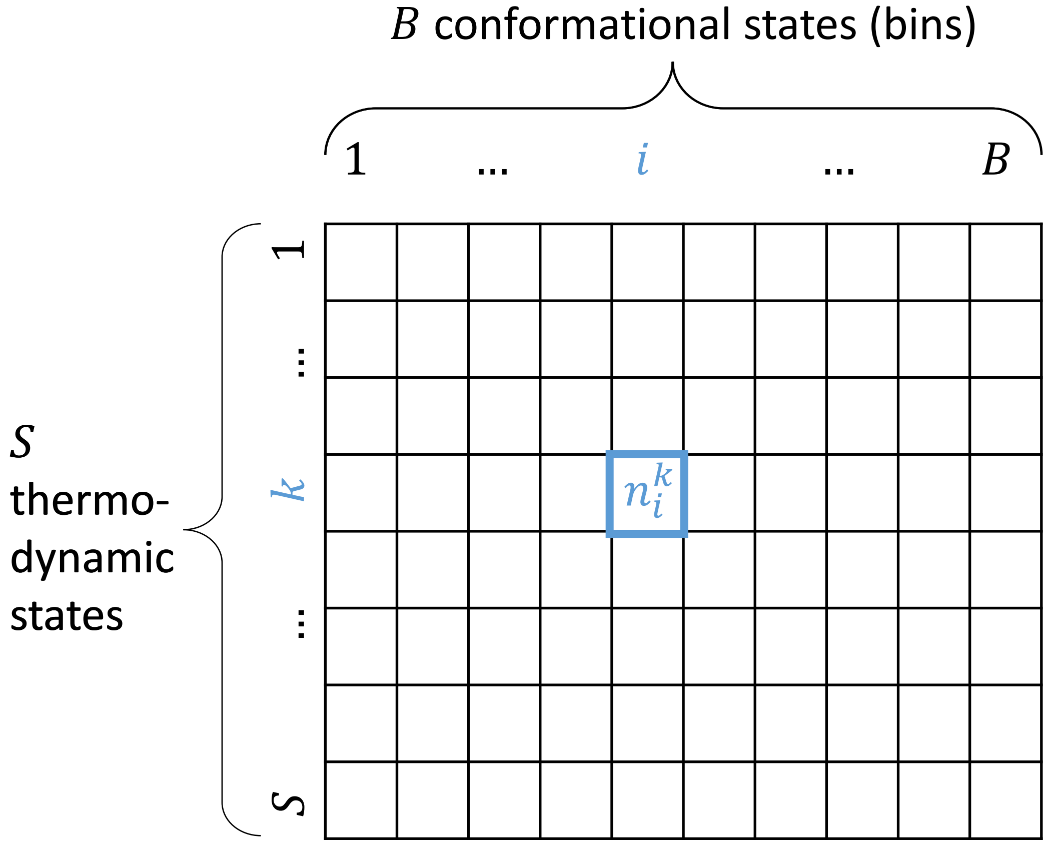
Input:

$c_i^k$ $\rightarrow$ Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

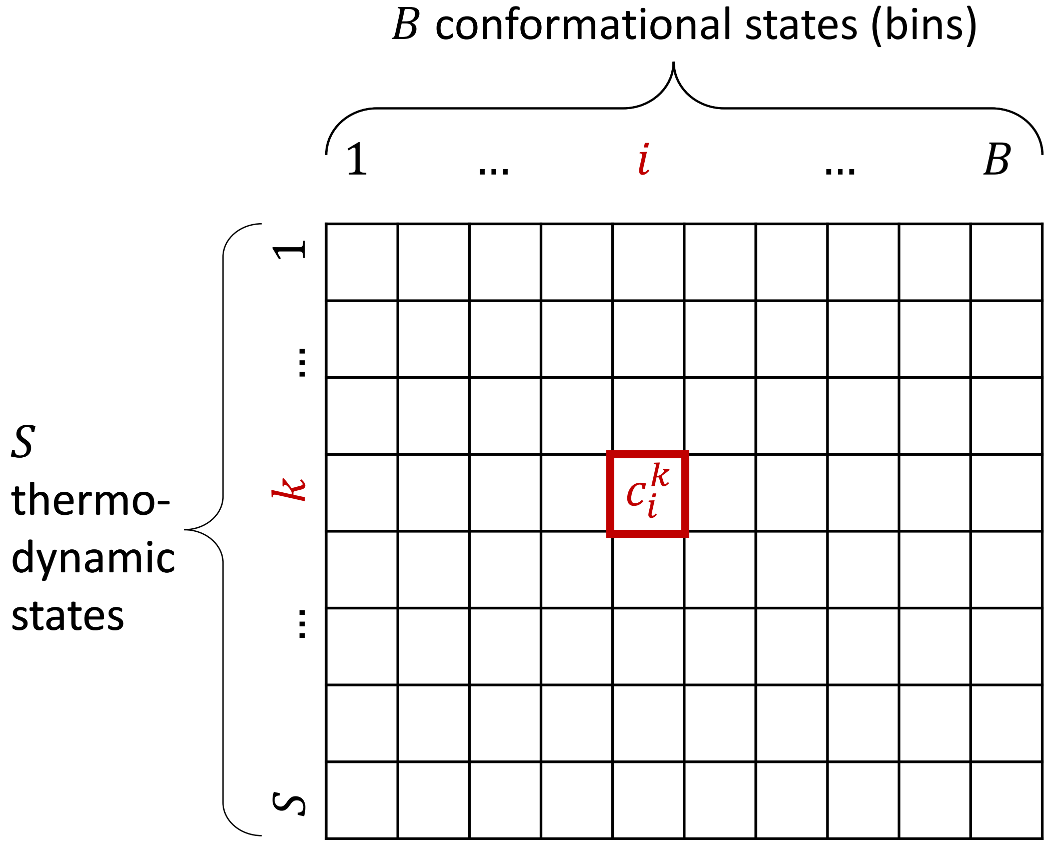
$n_i^k$ $\rightarrow$ State counts

# WHAM - input

1: Histogram containing all counts



2: Histogram containing bias energies

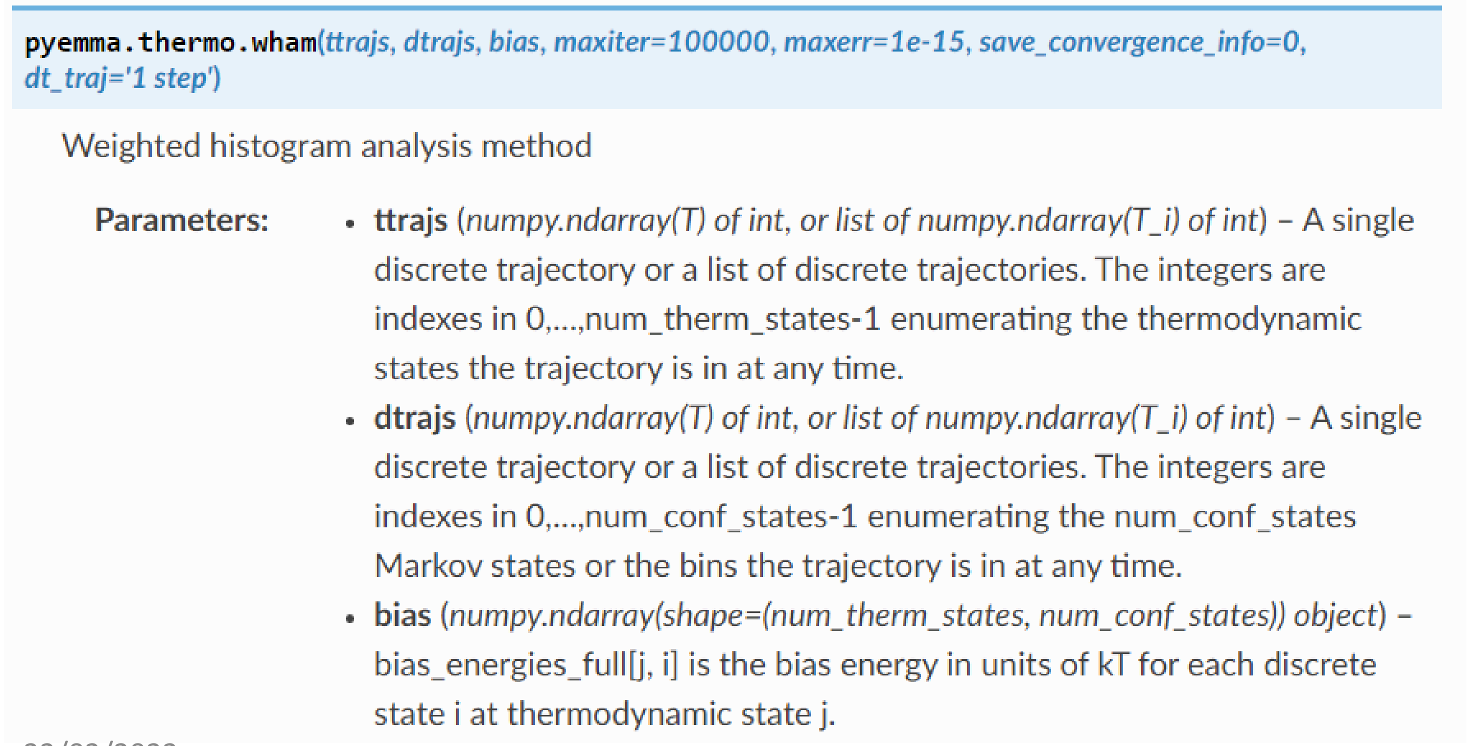


$c_i^k = \exp -b^k(x_i) \rightarrow$ Bias coeficcient in state $k$ at the center of bin $i$

# WHAM: do it yourself

- pyEMMA.thermo package contains a WHAM solver

http://emma-project.org/latest/api/generated/thermo-api/pyemma.thermo.wham.html



`pyemma.thermo.wham`(*ttrajs, dtrajs, bias, maxiter=100000, maxerr=1e-15, save_convergence_info=0, dt_traj='1 step'*)

Weighted histogram analysis method

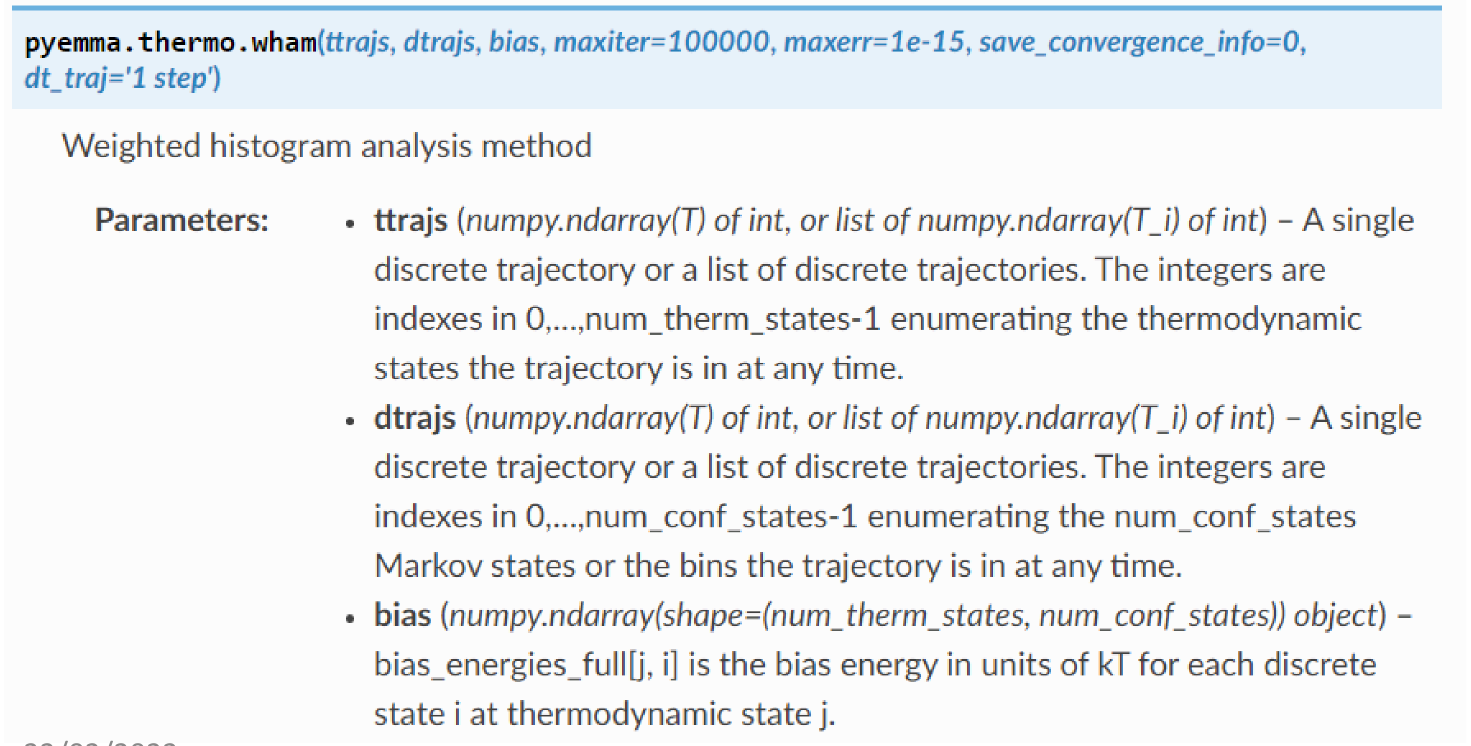
**Parameters:**

- **ttrajs** (*numpy.ndarray(T) of int, or list of numpy.ndarray(T_i) of int*) – A single discrete trajectory or a list of discrete trajectories. The integers are indexes in 0,...,num_therm_states-1 enumerating the thermodynamic states the trajectory is in at any time.
- **dtrajs** (*numpy.ndarray(T) of int, or list of numpy.ndarray(T_i) of int*) – A single discrete trajectory or a list of discrete trajectories. The integers are indexes in 0,...,num_conf_states-1 enumerating the num_conf_states Markov states or the bins the trajectory is in at any time.
- **bias** (*numpy.ndarray(shape=(num_therm_states, num_conf_states)) object*) – bias_energies_full[j, i] is the bias energy in units of kT for each discrete state i at thermodynamic state j.

## WHAM: do it yourself

- pyEMMA.thermo package contains a WHAM solver

http://emma-project.org/latest/api/generated/thermo-api/pyemma.thermo.wham.html



`pyemma.thermo.wham`*(ttrajs, dtrajs, bias, maxiter=100000, maxerr=1e-15, save_convergence_info=0, dt_traj='1 step')*

Weighted histogram analysis method

**Parameters:**

- **ttrajs** (*numpy.ndarray(T) of int, or list of numpy.ndarray(T_i) of int*) – A single discrete trajectory or a list of discrete trajectories. The integers are indexes in 0,...,num_therm_states-1 enumerating the thermodynamic states the trajectory is in at any time.
- **dtrajs** (*numpy.ndarray(T) of int, or list of numpy.ndarray(T_i) of int*) – A single discrete trajectory or a list of discrete trajectories. The integers are indexes in 0,...,num_conf_states-1 enumerating the num_conf_states Markov states or the bins the trajectory is in at any time.
- **bias** (*numpy.ndarray(shape=(num_therm_states, num_conf_states)) object*) – bias_energies_full[j, i] is the bias energy in units of kT for each discrete state i at thermodynamic state j.

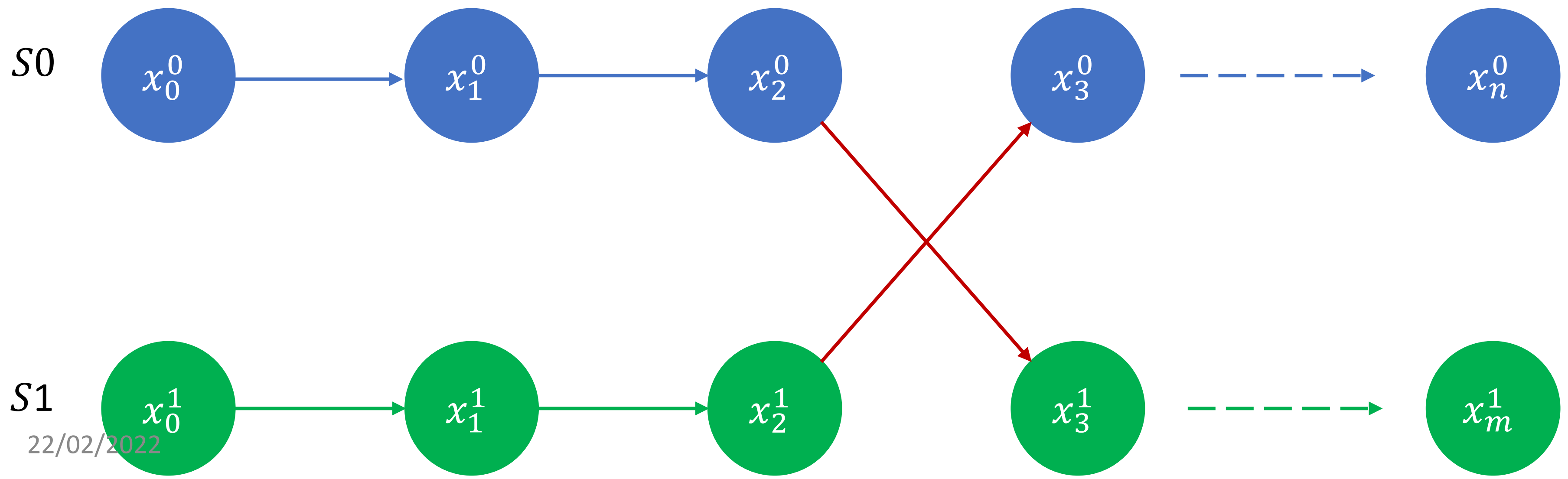
Thermodynamic state indices

Bin indices

Bias energies

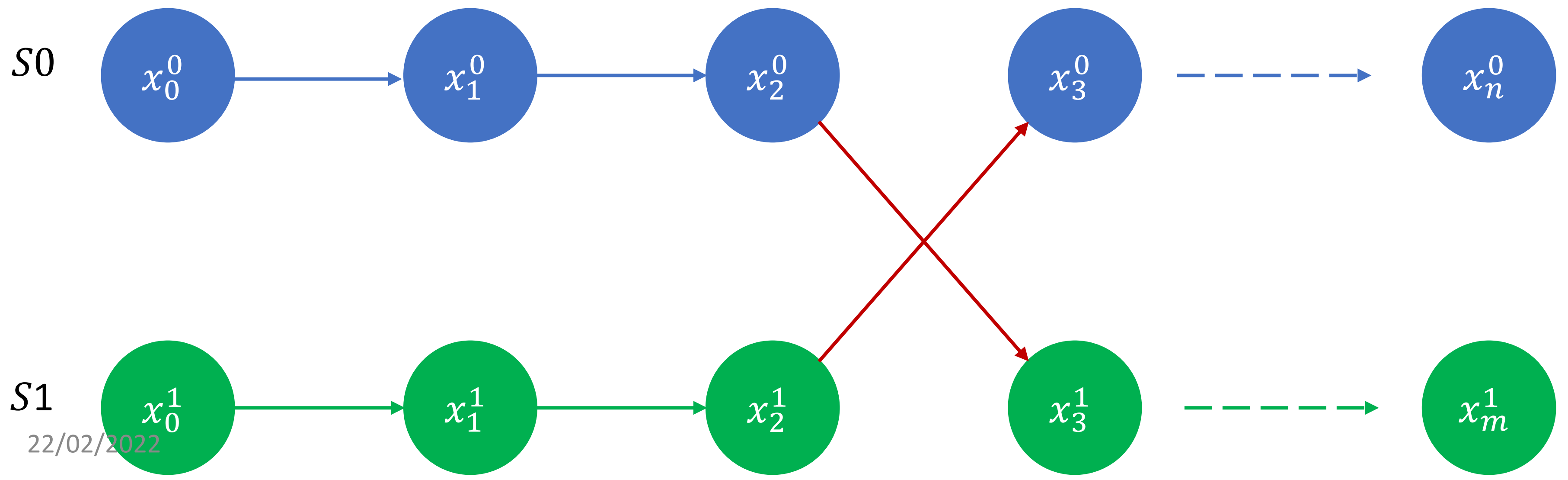
# WHAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]

ttrajs= [np.ndarray([0, 0, 0, 1, 0, 0, ...]),
         np.ndarray([1, 1, 1, 0, 1, 1, ...], ...]
```



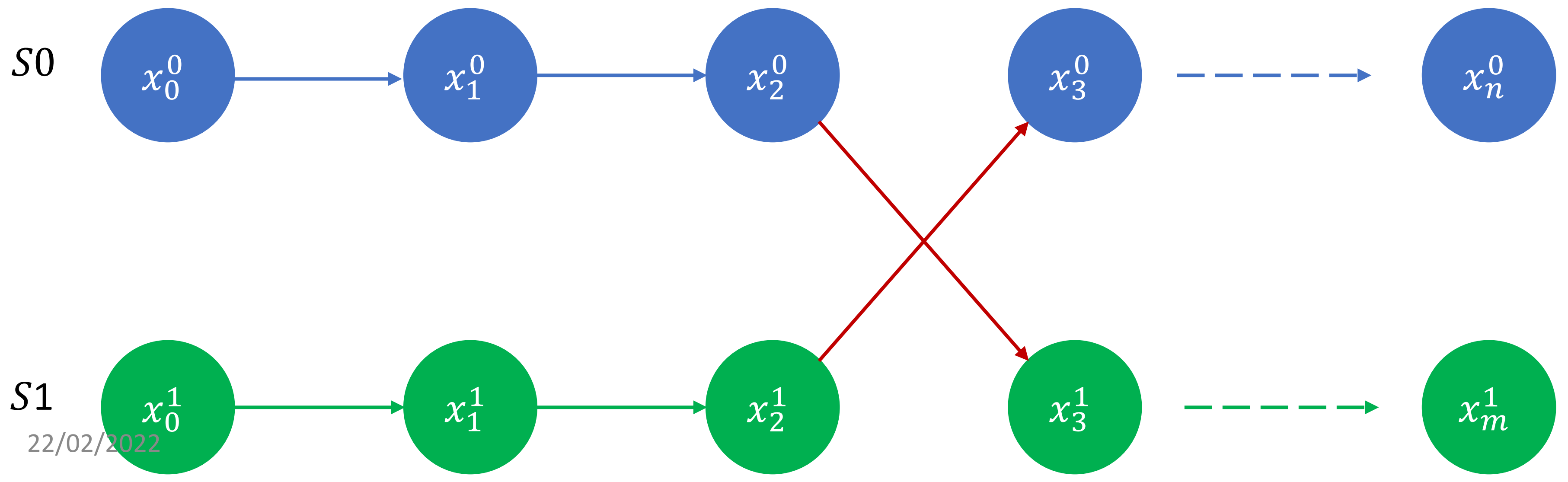
# WHAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]

ttrajs= [np.ndarray([0, 0, 0, 1, 0, 0, ...]),
         np.ndarray([1, 1, 1, 0, 1, 1, ...], ...]
```

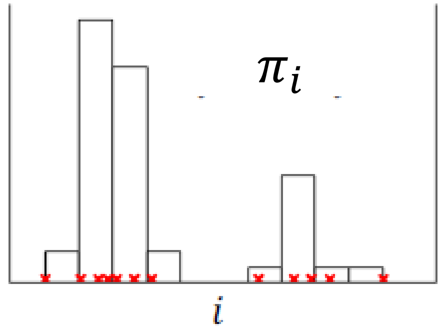


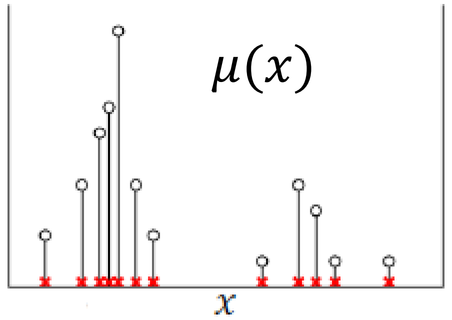
# WHAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]

ttrajs= [np.ndarray([0, 0, 0, 1, 0, 0, ...]),
         np.ndarray([1, 1, 1, 0, 1, 1, ...], ...]
```



# MBAR





- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0

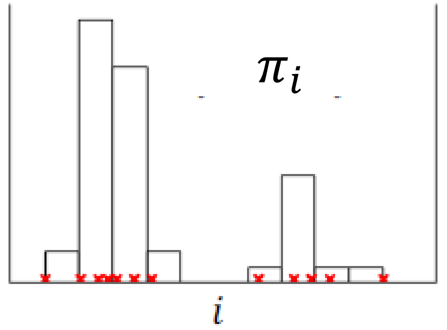
$c_i^k$ → Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

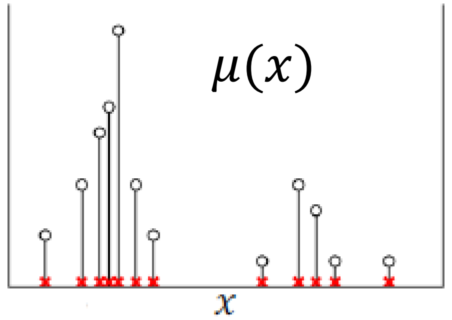
$$L_{WHAM}\left(n_i^k \middle| \pi_i\right) = \prod_{k=1}^{S} \prod_{i=1}^{B} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i} \right)^{n_i^k}$$

# MBAR

- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0

$c_i^k$ → Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

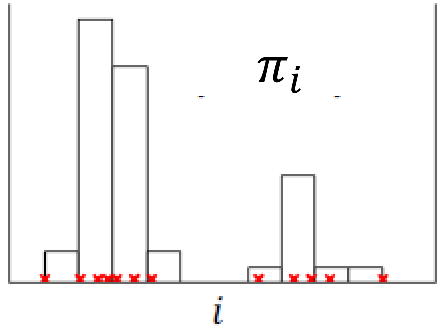


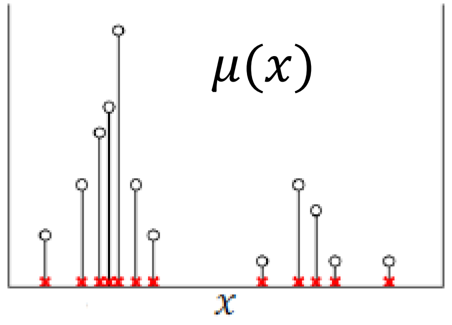


$$L_{WHAM}\left(n_i^k \middle| \pi_i\right) = \prod_{k=1}^{S} \prod_{i=1}^{B} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i} \right)^{n_i^k}$$

These become the bias energy for the sample coordinate
(= new bin center!): $\exp[-b^k(x_n^k)]$

# MBAR





- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0

$c_i^k$ → Bias coefficients $c_i^k = \exp[-b^k(x_i)]$

$$L_{WHAM}(n_i^k|\pi_i) = \prod_{k=1}^{S} \boxed{\prod_{i=1}^{B}} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i} \right)^{n_i^k}$$

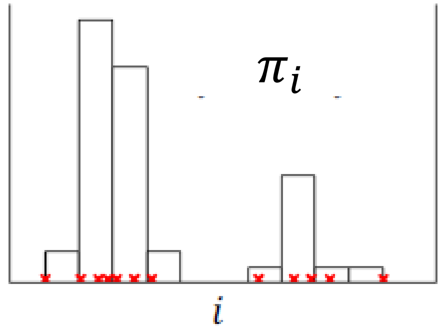
These become the bias energy for the sample coordinate
(= new bin center!): $\exp[-b^k(x_n^k)]$

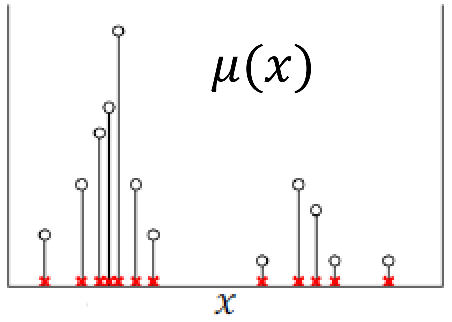
This becomes a product over all samples

# MBAR

- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0

$c_i^k$ → Bias coefficients $c_i^k = \exp[-b^k(x_i)]$





$$L_{WHAM}(n_i^k|\pi_i) = \prod_{k=1}^{S} \boxed{\prod_{i=1}^{B}} \left(\frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i}\right)^{n_i^k}$$

Bin counts become…?

These become the bias energy for the sample coordinate

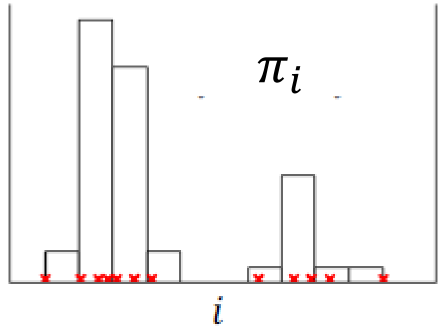
(= new bin center!): $\exp[-b^k(x_n^k)]$

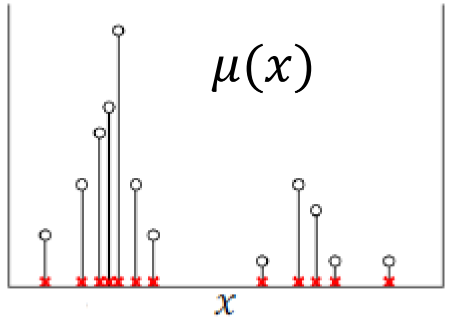
This becomes a product over all samples

# MBAR

- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0

$c_i^k$ → Bias coefficients $c_i^k = \exp[-b^k(x_i)]$





$$L_{WHAM}(n_i^k|\pi_i) = \prod_{k=1}^{S} \boxed{\prod_{i=1}^{B}} \left( \frac{c_i^k \pi_i}{\sum_{j=1}^{B} c_j^k \pi_i} \right)^{n_i^k}$$

Bin counts become…?

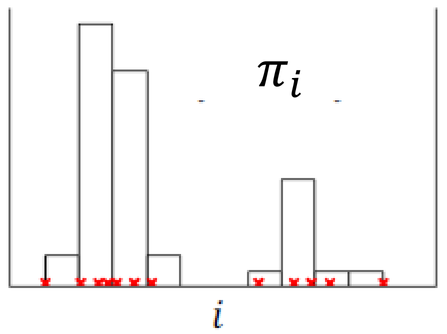
These become the bias energy for the sample coordinate
(= new bin center!): $\exp[-b^k(x_n^k)]$

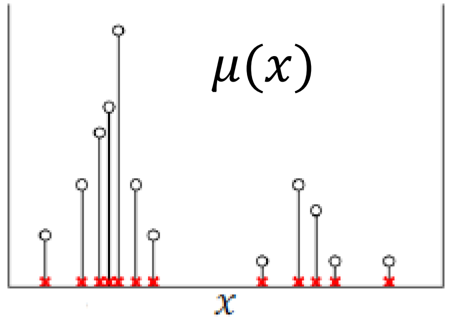
This becomes a product over all samples

We reweight not $\pi_i$ , but with respect to a distribution $\mu(x)$ over all samples

# MBAR

- A.k.a. binless WHAM/UWHAM
- Derived from WHAM by taking the limit of bin widths → 0





$$L_{MBAR}(f^1, \dots, f^S) = \prod_{k=1}^{S} \prod_{n=1}^{N^k} \exp[-b^k(x_n^k) + f^k]\, \mu(x_n^k)$$

(It's convex w.r.t. the $f^k$)

$$\mu(x) = 1/\sum_{l=1}^{S} N^l \exp[-b^l(x) + f^l]$$

Shirts, Michael R., and John D. Chodera. "Statistically optimal analysis of samples from multiple equilibrium states." *The Journal of chemical physics* 129.12 (2008): 124105

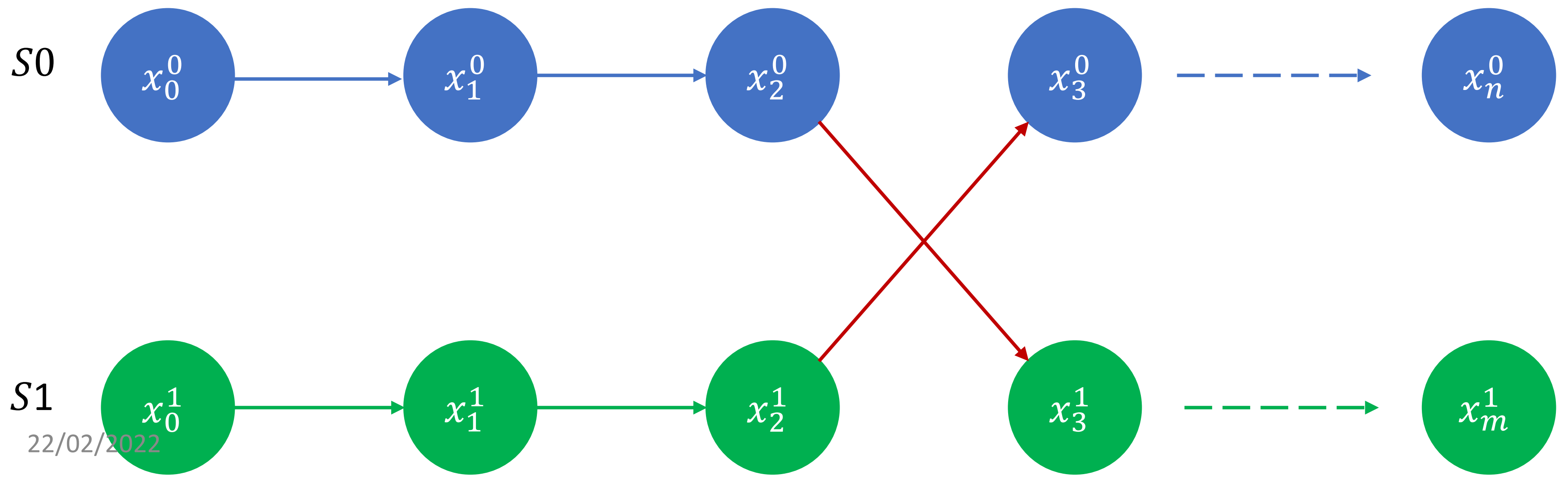
# TRAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]
```

$$\begin{tikzcd}
S0 & x_0^0 \arrow[r] & x_1^0 \arrow[r] & x_2^0 \arrow[r] & x_3^0 \arrow[r, dashed] & x_n^0 \\
S1 & x_0^1 \arrow[r] & x_1^1 \arrow[r] & x_2^1 \arrow[r] & x_3^1 \arrow[r, dashed] & x_m^1
\end{tikzcd}$$

# TRAM - input

```
dtrajs= [np.ndarray([0, 1, 2, 3, 2, 3, ...]),
         np.ndarray([3, 4, 2, 3, 4, 4, ...]), ...]

ttrajs= [np.ndarray([0, 0, 0, 1, 0, 0, ...]),
         np.ndarray([1, 1, 1, 0, 1, 1, ...], ...]
```



$S0$
$x_0^0$
$x_1^0$
$x_2^0$
$x_3^0$
$x_n^0$
$S1$
$x_0^1$
$x_1^1$
$x_2^1$
$x_3^1$
$x_m^1$