bioexcel
Centre of Excellence for Computational Biomolecular Research

Horizon 2020
European Union Funding
for Research & Innovation

Advanced simulation protocols in GROMACS

Christian Blau

blau@kth.se

Ensemble scaling in biomolecular simulations

transitions in biomolecules are (very) infrequent

but transitions are often fast

can (smartly) combine sampling for many shorter simulations

Different problems, trivial parallelism

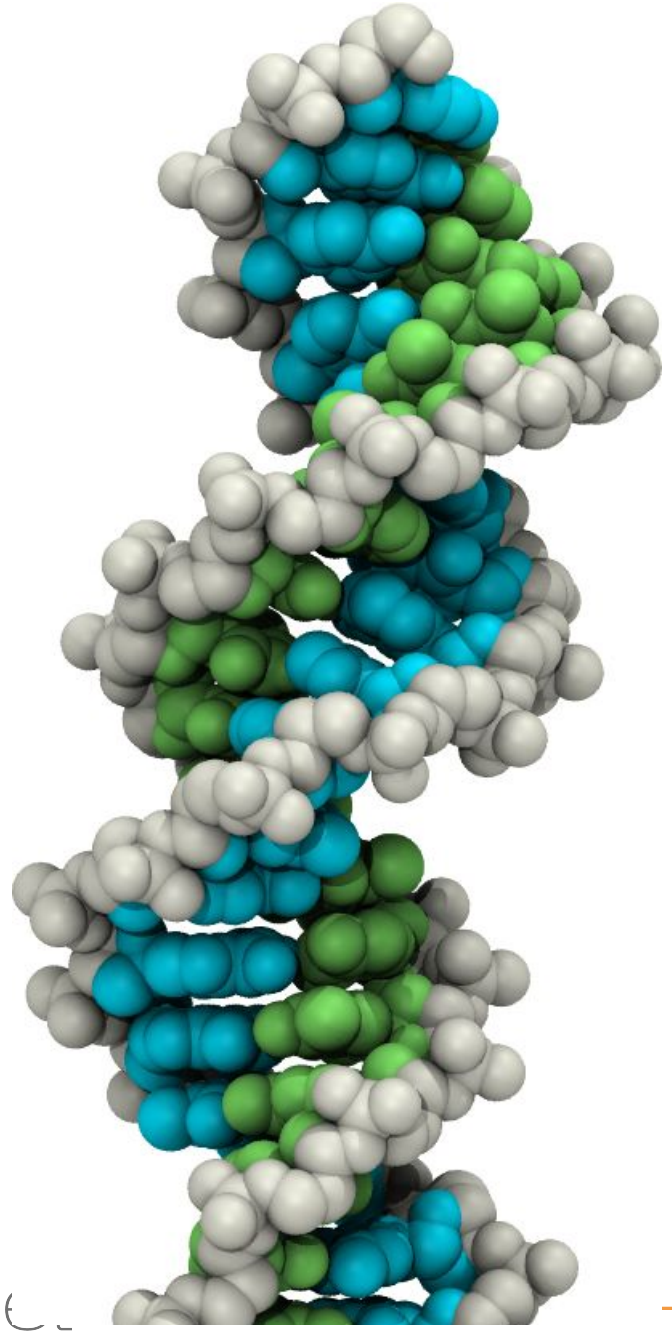
- Different drug molecules binding to the same protein
- Different mutations to the same protein

Ensemble parallelism

Processes in cells and experiments involve ensembles of large numbers of molecules

We can use ensembles in simulations:

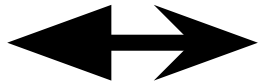
- Do N simulations instead of 1 (trivial, common practice)
- Combine information from N simulations to get answers faster
  - Markov state modelling
  - Aggregating data from multiple walkers



Example study:

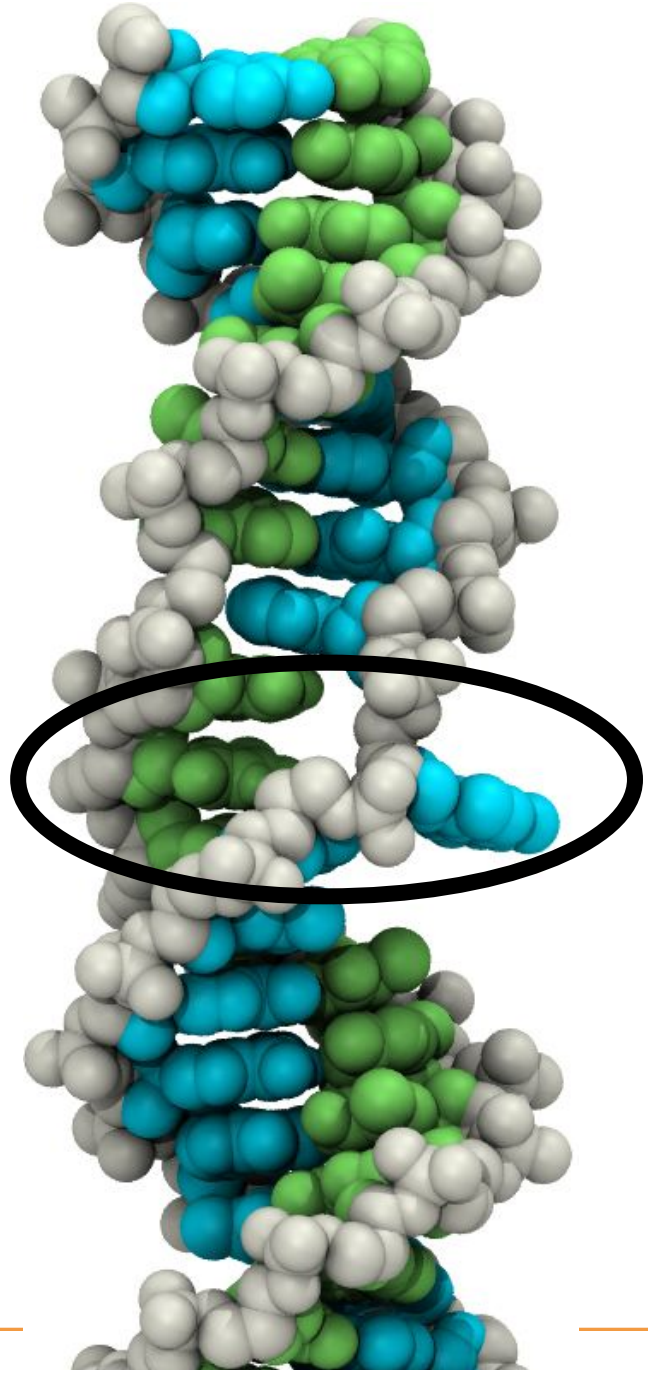
DNA base pair opening

$1\ ms^{-1}$



closing

$10^4\ ms^{-1}$



Different Approaches

General exploration

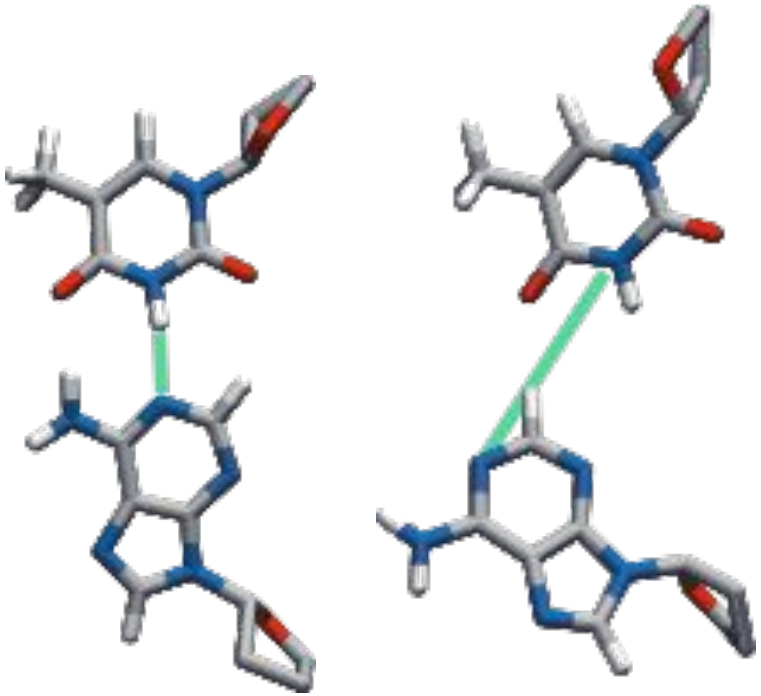
- Simple MD simulation
- Replica exchange protocols
- “Flooding”
- Conformational sampling

When you know your reaction coordinate

- Pulling
- Accelerated weighted histogram method

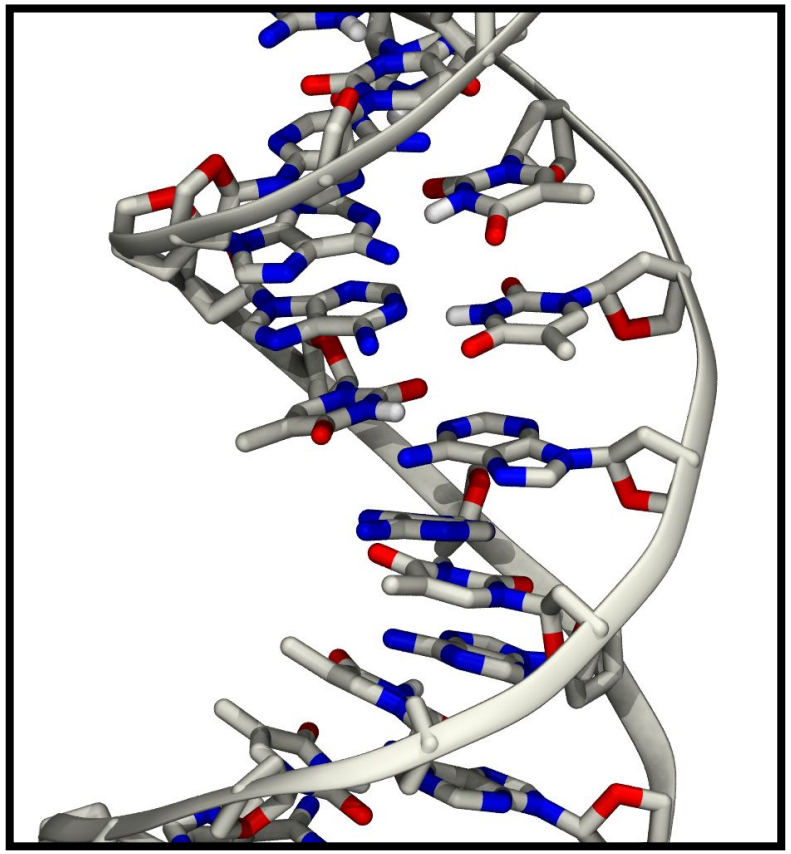
A reaction coordinate to enhance sampling

base-pair opening distance



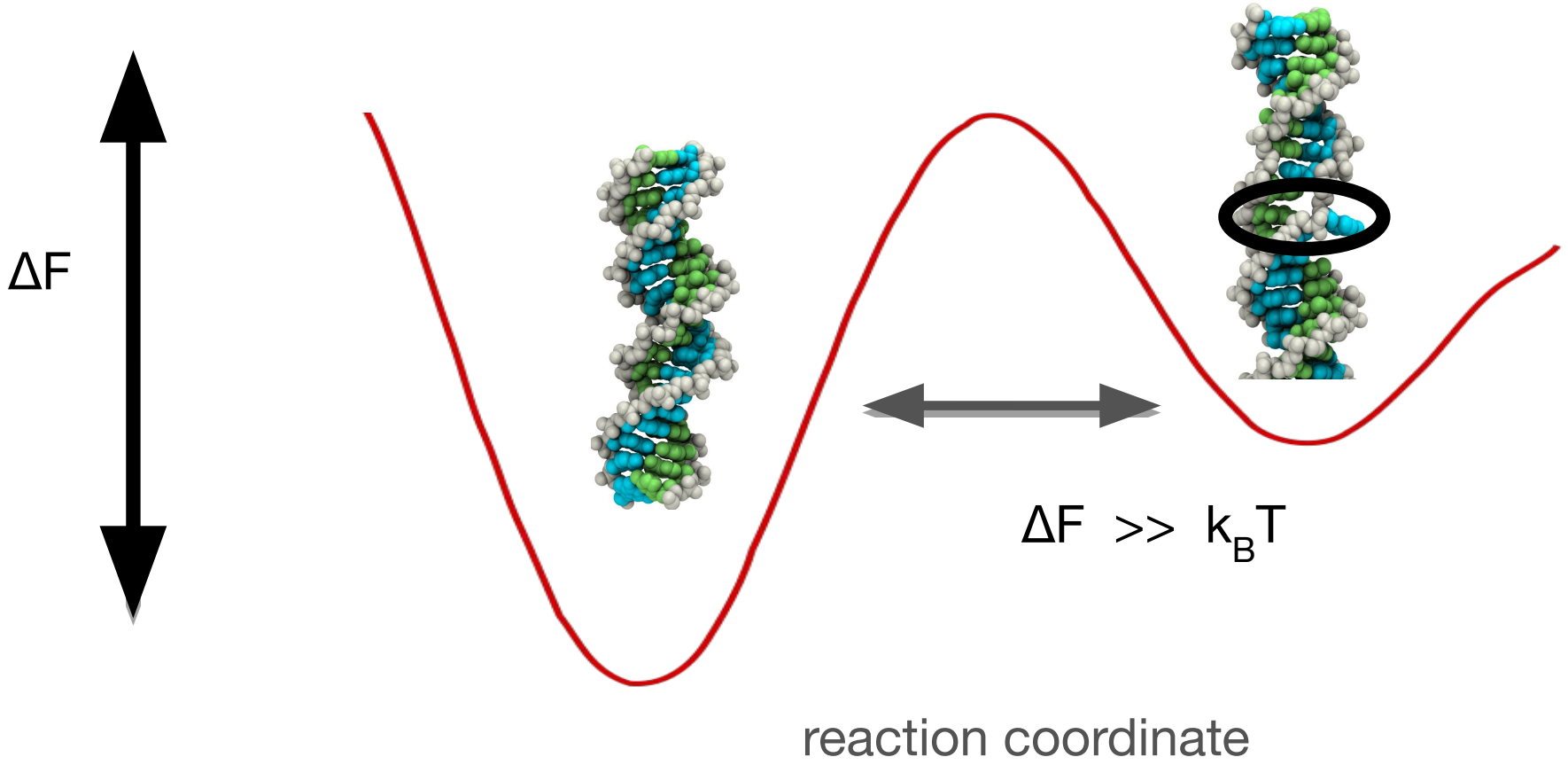
closed

open



Sampling a Boltzmann distribution:

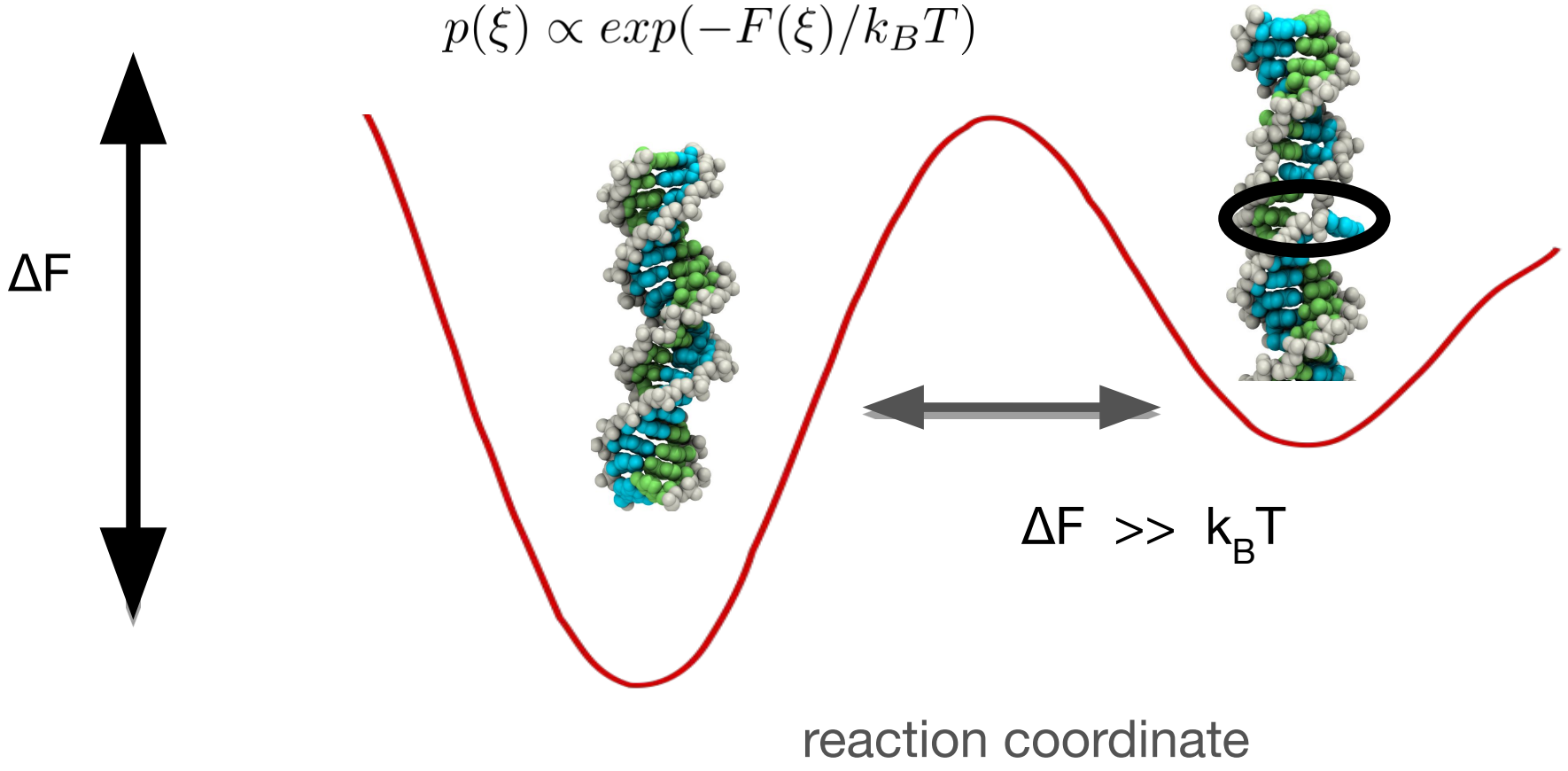
$$p(x) \propto exp(-V(x)/k_B T)$$



Sampling a Boltzmann distribution:

$$p(x) \propto exp(-V(x)/k_B T)$$

$$p(\xi) \propto exp(-F(\xi)/k_B T)$$

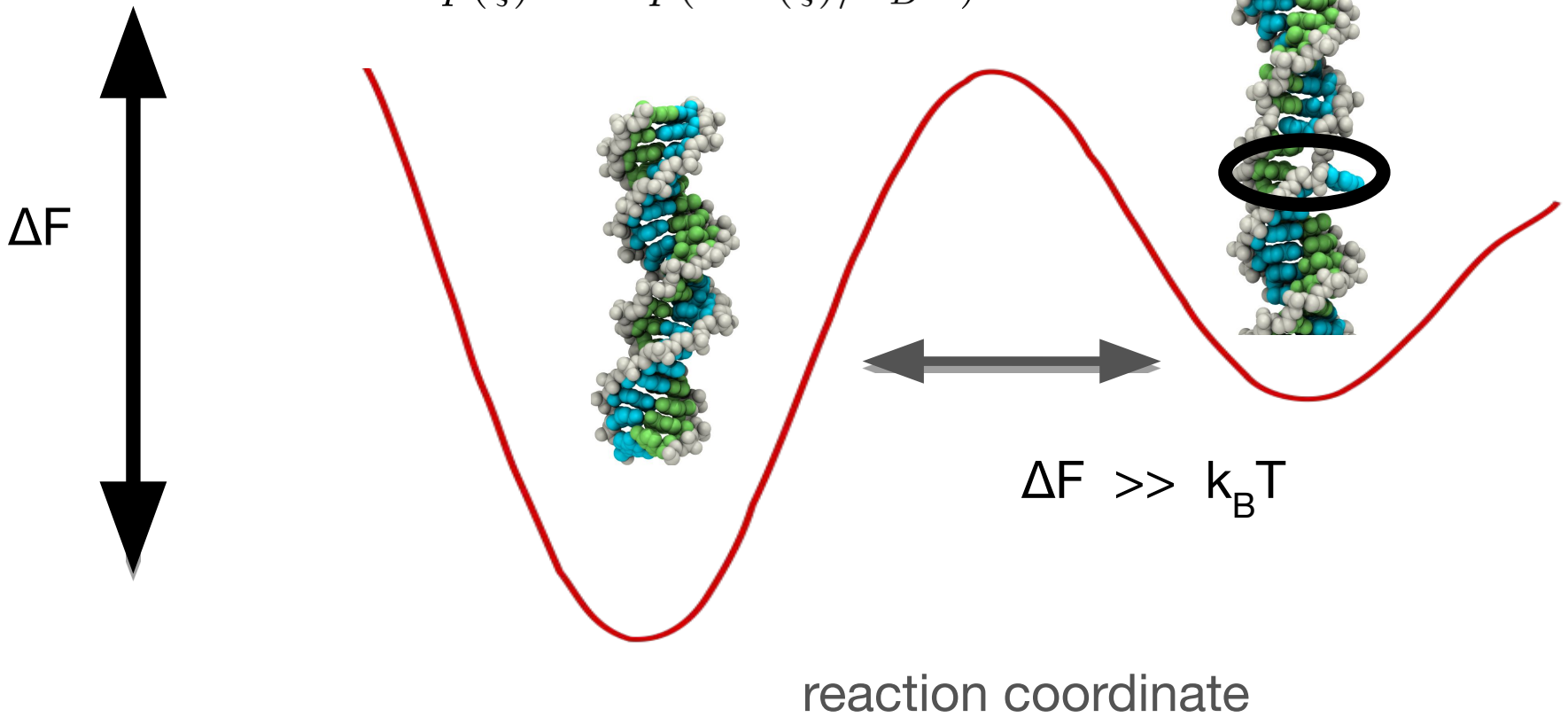


Sampling a Boltzmann distribution:

$$p(x) \propto exp(-V(x)/k_BT)$$

$$p(\xi) \propto exp(-F(\xi)/k_BT)$$

$$p(\xi) \propto \int exp(-F(\xi(x))/k_BT)dx$$

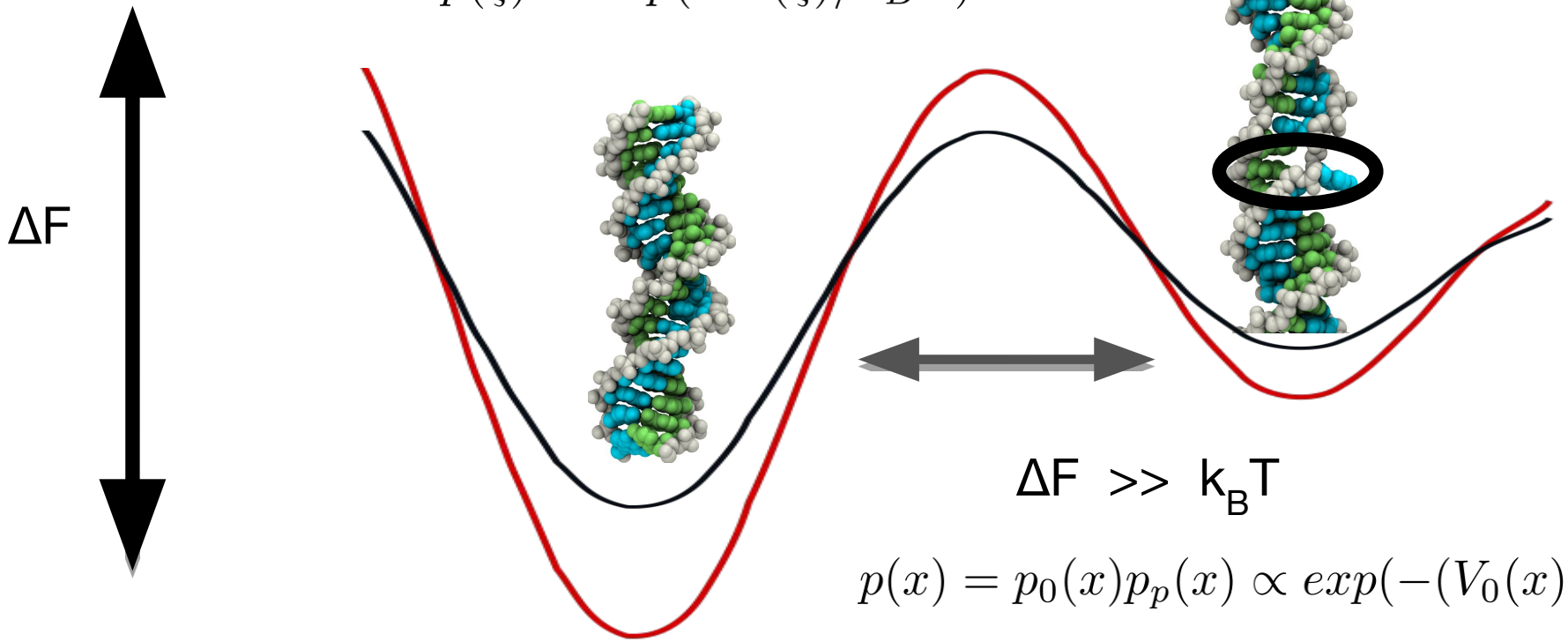


Sampling from a different Boltzmann distribution

$$p(x) \propto exp(-V(x)/k_BT)$$

$$p(\xi) \propto exp(-F(\xi)/k_BT)$$

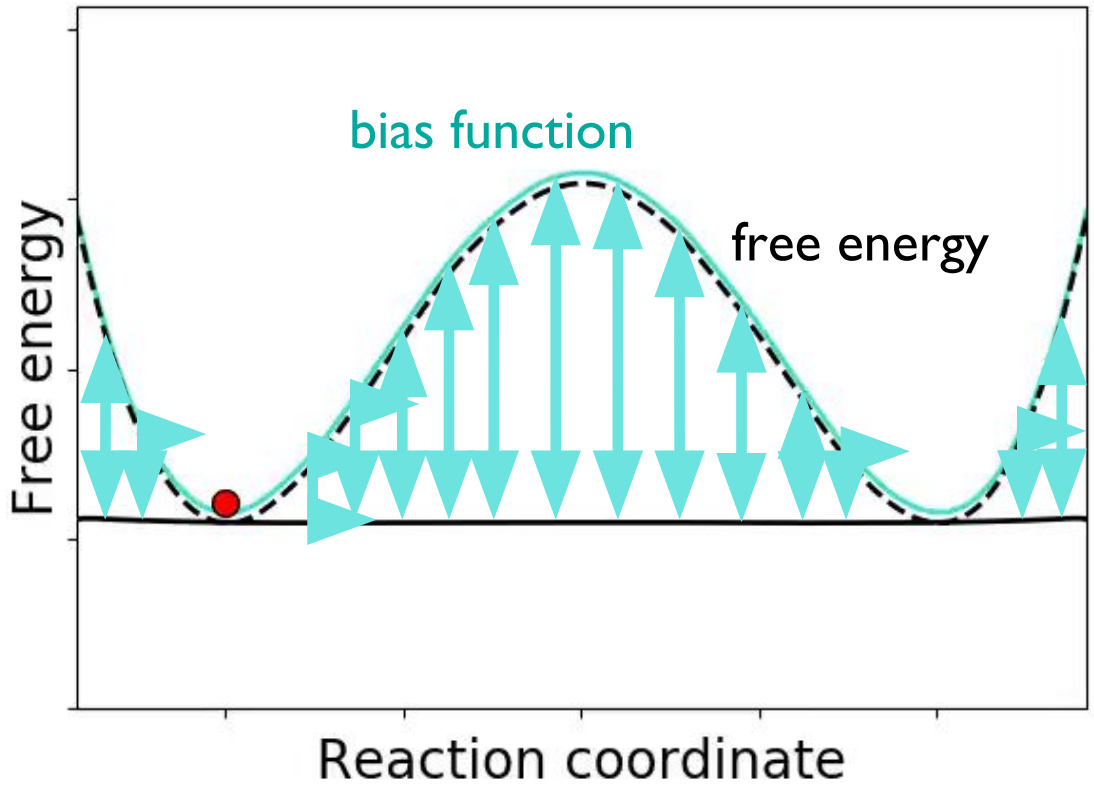
$$p(\xi) \propto \int exp(-F(\xi(x))/k_BT)dx$$



$$p(x) = p_0(x)p_p(x) \propto exp(-(V_0(x) + V_p(x))/k_BT)$$

Trick: add a bias potential to make the effective potential flat

Issue: the potential (or free-energy) is what we are after!



Need the free energy to apply the right bias

reaction coord. free energy

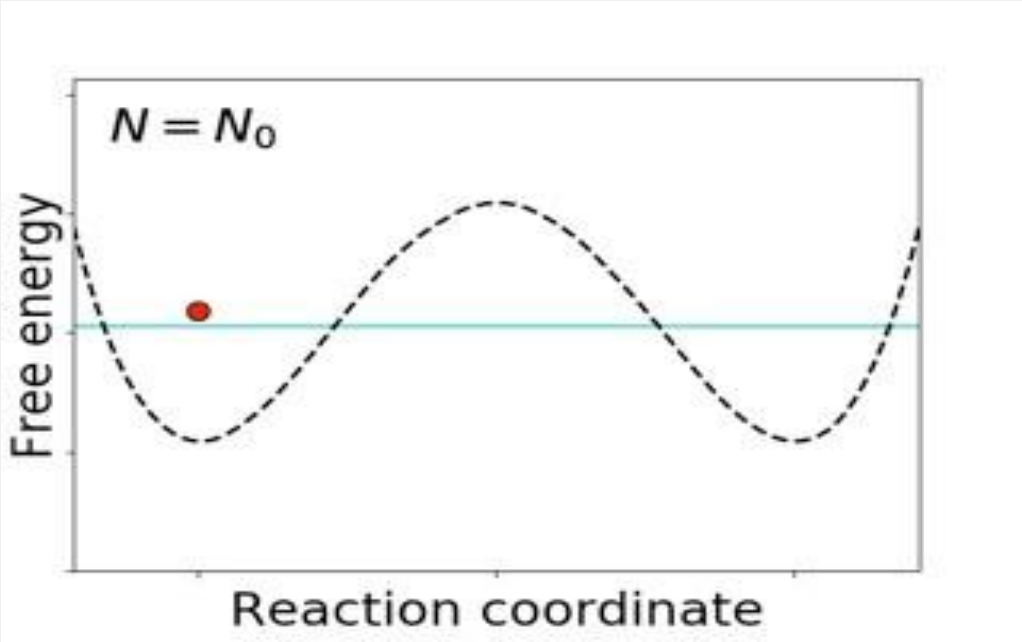
$$\rho(\xi) = e^{-\beta F(\xi) + g(\xi)} \quad \Leftrightarrow \quad g(\xi) = \beta F(\xi) + \ln \rho(\xi)$$

target distribution bias function

Calculating the free energy requires the bias,
the bias requires the free energy
— proceed adaptively!

Adaptively estimate free energy and applies the bias

1. Estimate free energy
2. Set bias
3. Collect (biased) samples
4. Update free energy estimate

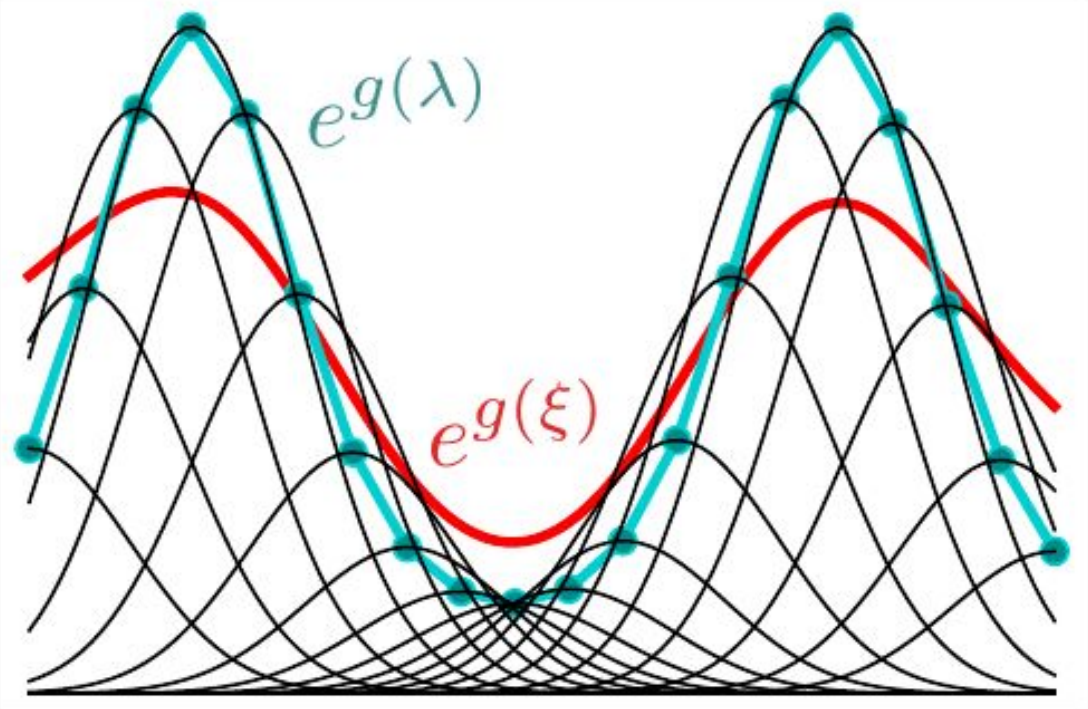


How is the bias represented and applied?

- Discretize ξ space with grid points λ
- Parameterize bias using Gaussian “basis” functions*

$$e^{g(\xi)} = \sum_{\lambda} e^{g(\lambda)} e^{-\frac{1}{2}\left(\frac{\xi-\lambda}{\sigma}\right)^2}$$

- Width σ sets resolution
- Bias force: $\nabla g(\xi(x))$



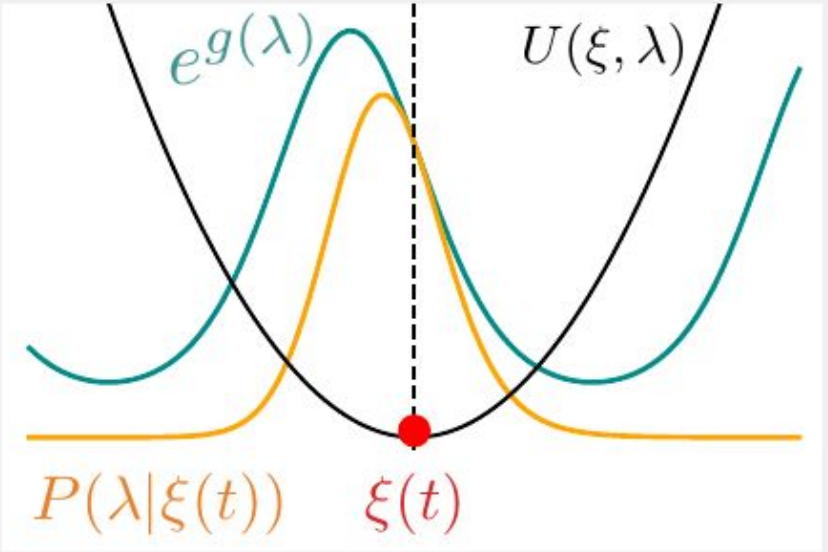
Physical interpretation of λ

$$e^{g(\xi)} = \sum_{\lambda} e^{g(\lambda)} e^{-\frac{1}{2}\left(\frac{\xi-\lambda}{\sigma}\right)^2}$$

Probabilistically the same as a having a “particle” at λ that experiences an external bias g(λ) and interacts with ξ through a harmonic potential

$$U(\xi,\lambda) = \frac{k}{2}(\xi-\lambda)^2, \qquad k = \frac{\beta}{\sigma^2}$$

force constant



λ stays close to ξ for large force constants:

$$P(\lambda|\xi(t)) \propto e^{g(\lambda)-\frac{1}{2}\beta k(\xi(t)-\lambda)^2}$$

AWH works on λ rather than ξ

The algorithm again, now with more detail

1. Estimate free energy 2. Set bias 3. Collect samples 4. Update free energy

estimate $F_n(\lambda) \approx F(\lambda)$ exact

$$e^{-\beta F(\lambda)} = \int e^{-\beta F(\xi)} e^{-\frac{1}{2}\left(\frac{\xi-\lambda}{\sigma}\right)^2}$$

"convolved free energy"

- The PMF F(ξ) is extracted by an on the fly reweighting procedure

AWH works on λ rather than ξ

The algorithm again, now with more detail

1. Estimate free energy **2. Set bias** 3. Collect samples 4. Update free energy

$$g_n(\lambda) = \beta F_n(\lambda) + \ln \rho(\lambda)$$

bias free energy estimate target

AWH works on λ rather than ξ

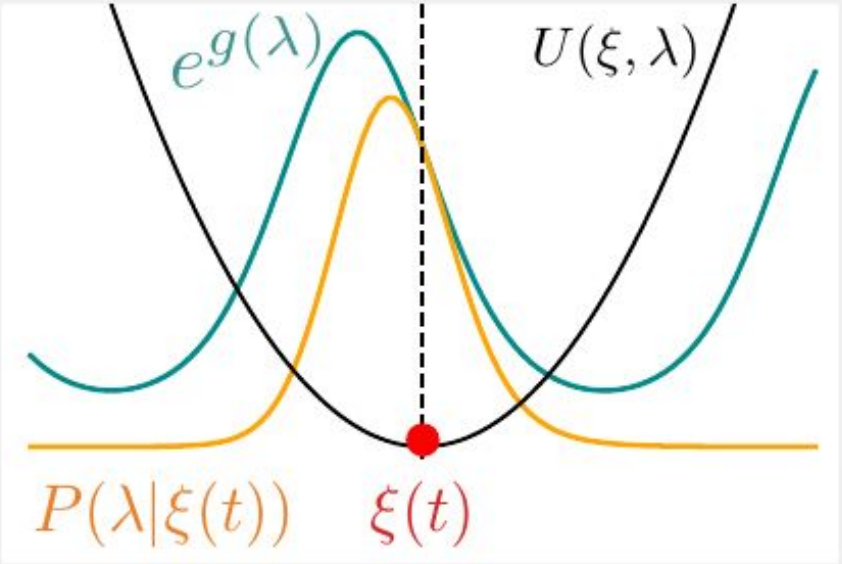
The algorithm again, now with more detail

1. Estimate free energy 2. Set bias **3. Collect samples** 4. Update free energy

sample at time t, bias g_n

$$w_n(\lambda|\xi(t)) = P_n(\lambda|\xi(t))$$
$$= e^{g_n(\lambda) - \frac{1}{2}\beta k(\xi(t)-\lambda)^2}$$

a "biased" Gaussian



AWH works on λ rather than ξ

The algorithm again, now with more detail

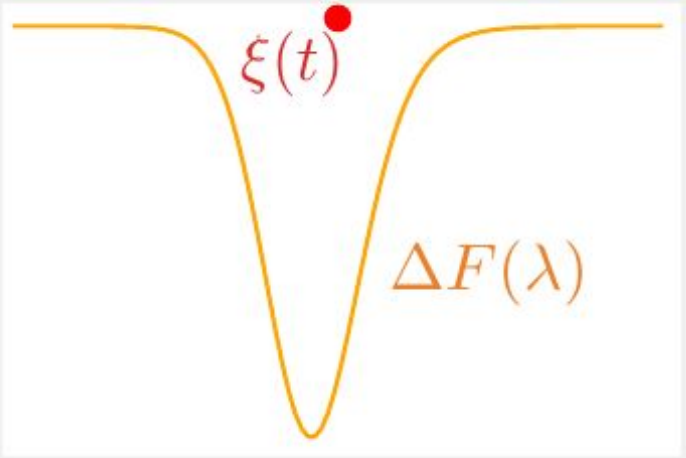
1. Estimate free energy 2. Set bias 3. Collect samples **4. Update free energy**

what was sampled

$$\Delta F_n(\lambda) \approx -\frac{\sum_t w_n(\lambda|t)}{N_n \rho(\lambda)}$$

prior number of samples

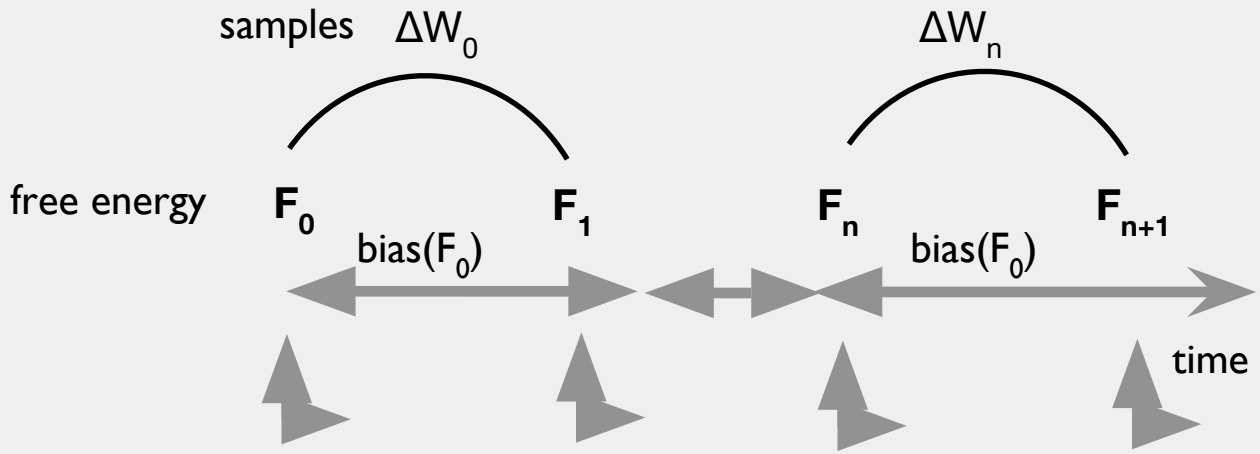
how we hoped to sample



Accelerated Weight Histogram Method

Iterative scheme to solve for the unknown bias / free-energy:

- collect samples (using MD)
- update the free-energy estimate



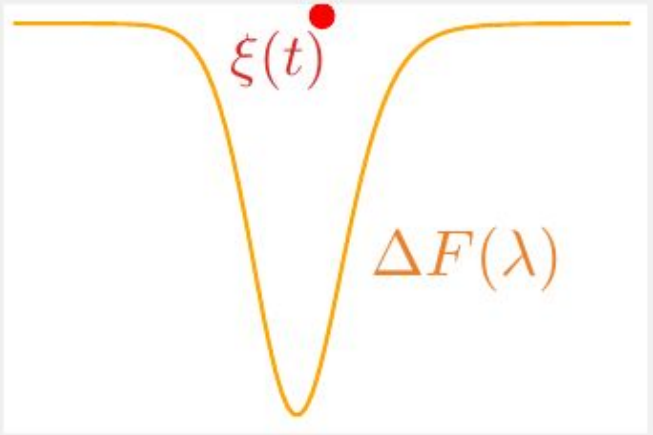
collected samples

$$\Delta F_n(\lambda) = -\ln\left(1 + \frac{\Delta W_n(\lambda)}{N_n \pi(\lambda)}\right) \sim \frac{1}{N_n}$$

prior number of samples

target distribution (flat)

The free energy update size



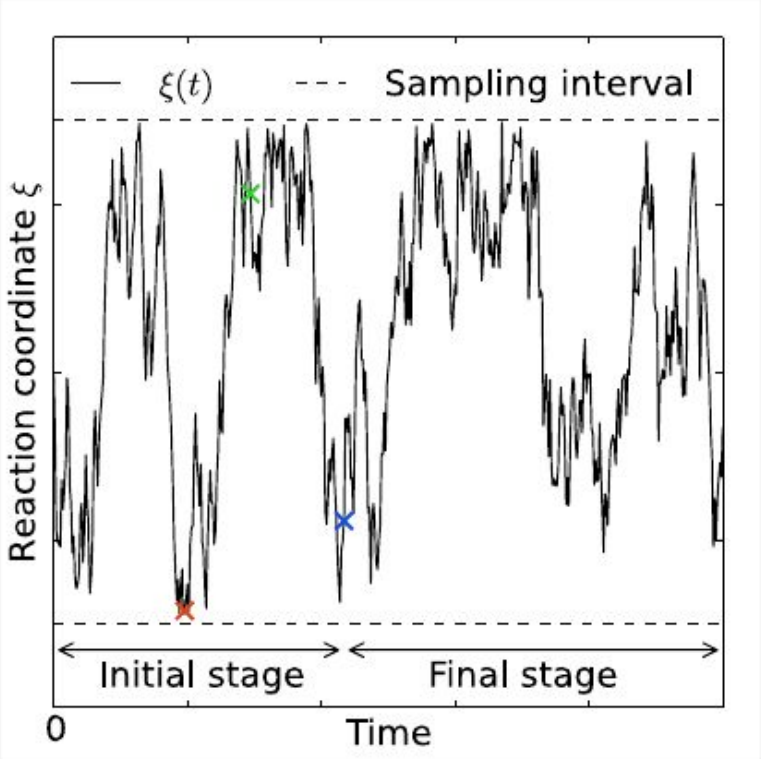
$$|\Delta F_n| \sim \frac{1}{N_n}$$

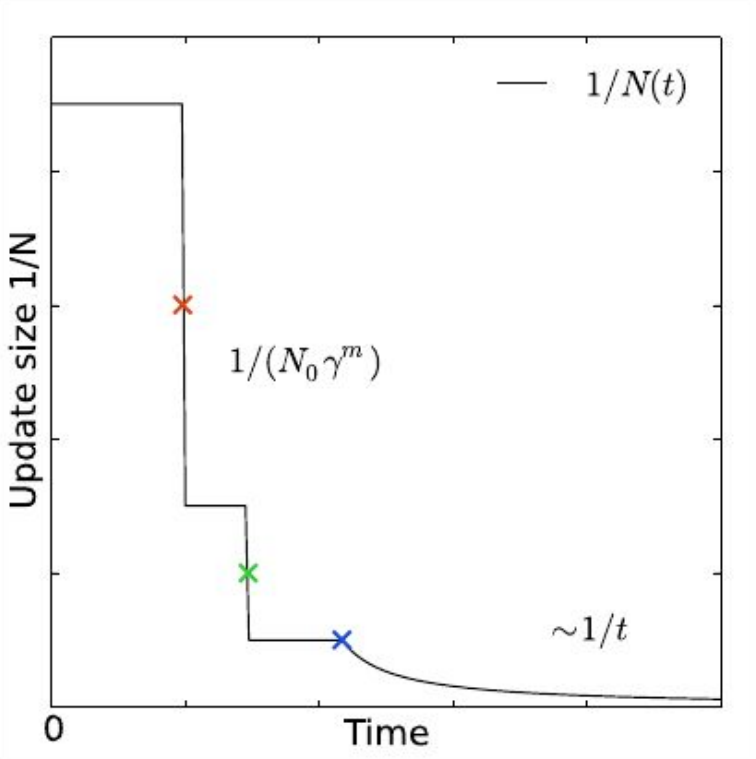
- N ~ total number of prior samples
- reflects the accuracy of the free energy estimate
- should grow at sampling rate

$$N_n \sim t$$

Adding robustness — the initial stage

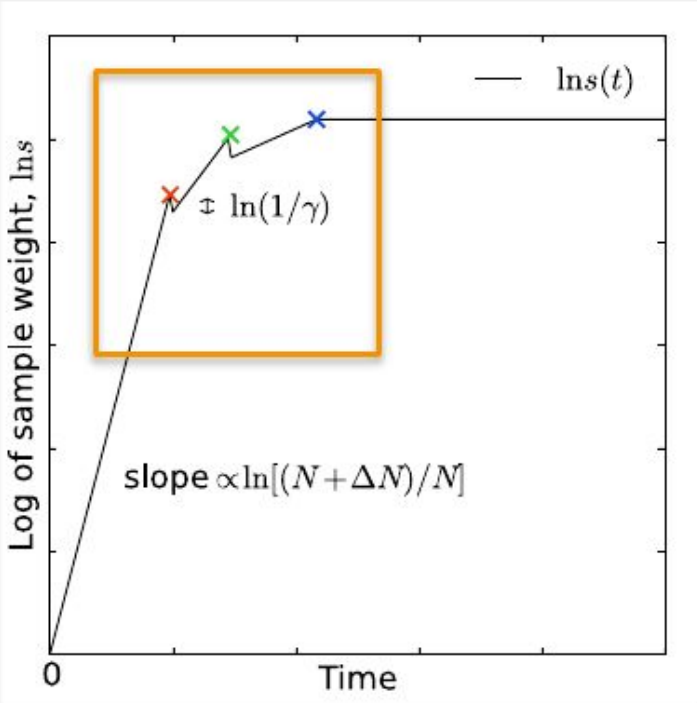
- Letting N grow at “naturally” decreases the update size too rapidly initially
- *The initial stage:* keep the update size large for the first few transitions.
- After each covering of the sampling interval $N_{n+1} = \gamma N_n$

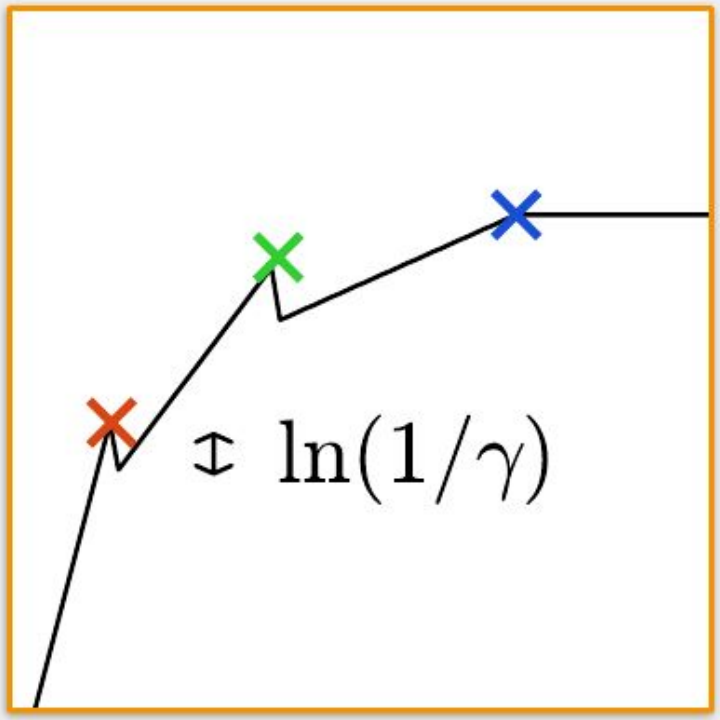




Exiting the initial stage

- Scaling N is corresponds to rescaling the current sample weight
- Scaling up, $N_{n+1} = \gamma N_n$, *decreases* sample weight
- Scaling down, $N_n = \text{const.}$, *increases* sample weight
- Exit initial stage when sample weight is no longer increasing



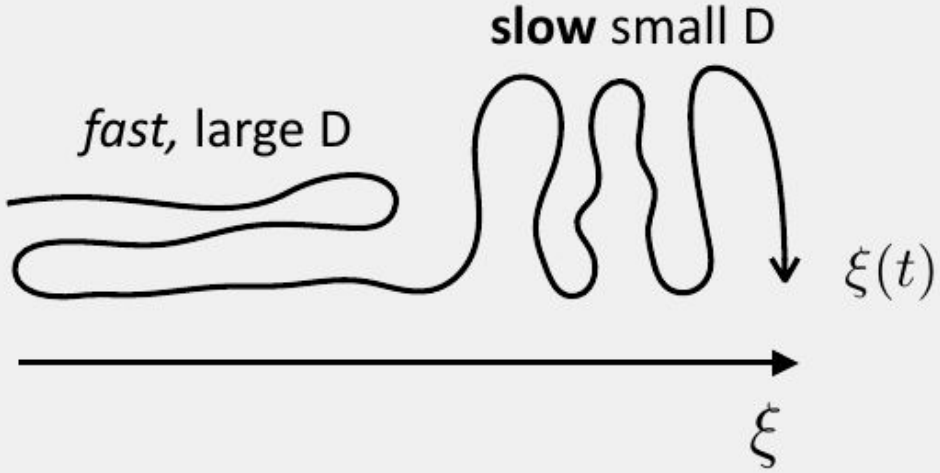


The initial update size $|\Delta F| \sim \frac{1}{N}$

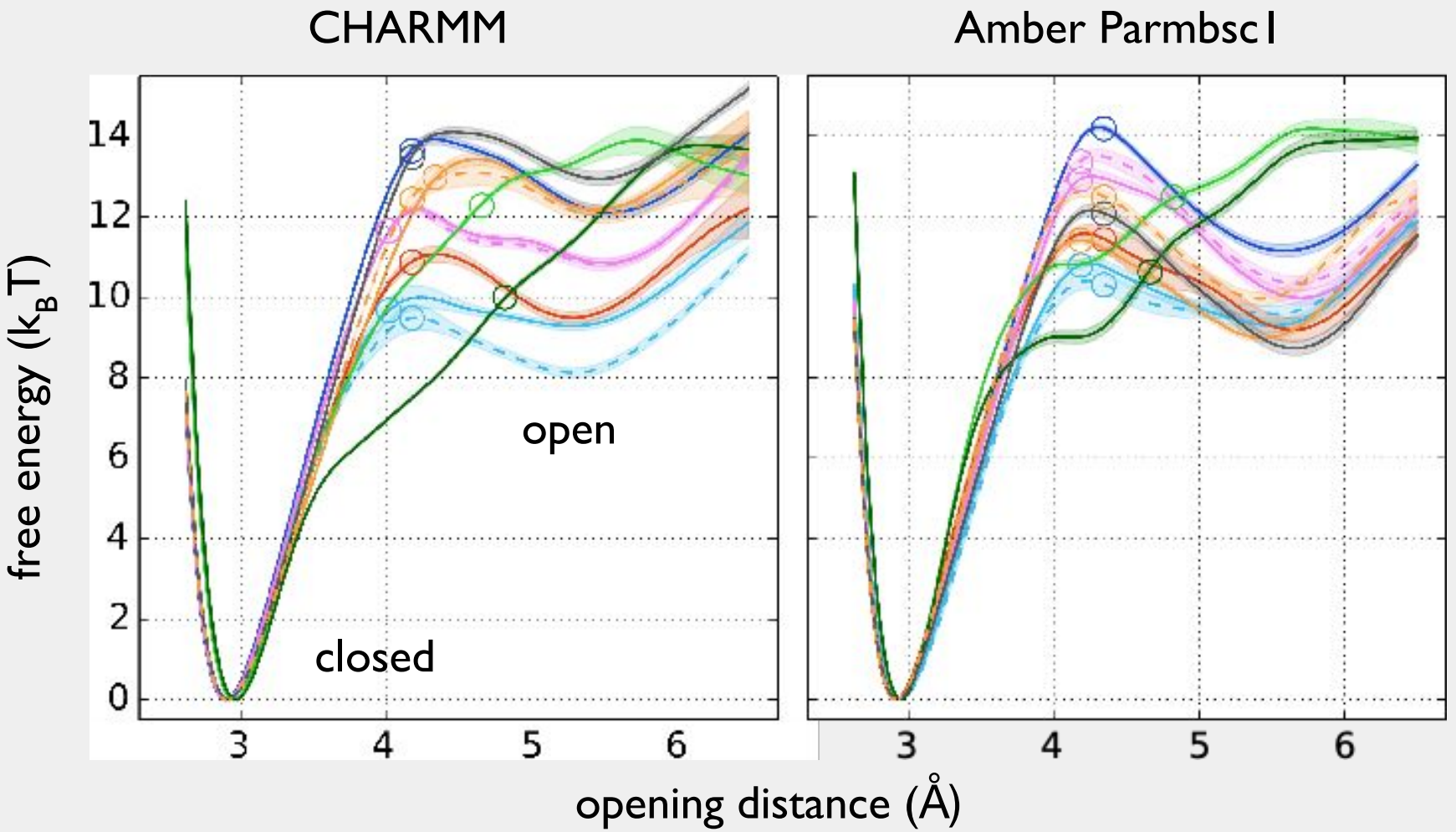
- Sets fluctuation of the free energy and the bias.
- Slow regions get larger fluctuations than fast ones (given N)
- Parameterized in AWH by a *diffusion constant* D

$$\frac{1}{N_0} \sim D$$

- Slow system — small D
- Fast system — large D



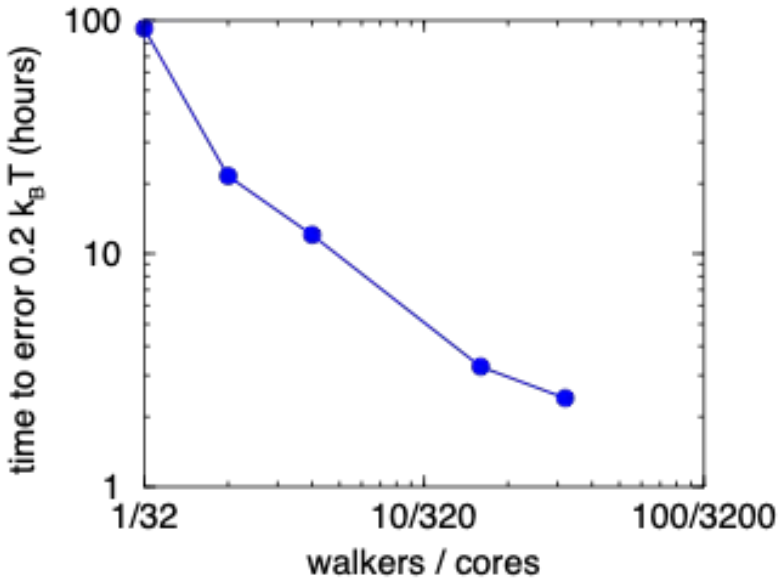
Results for different force fields (models) and base pairs

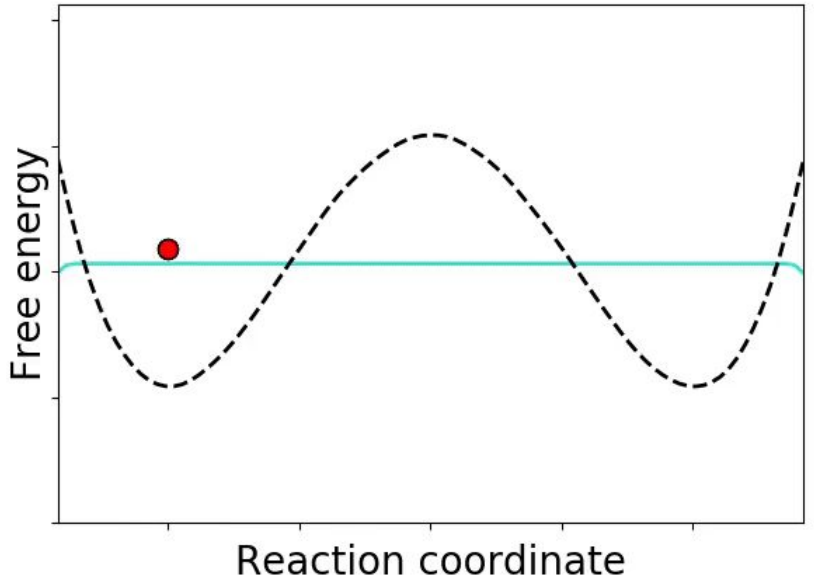


AWH with ensemble parallelism

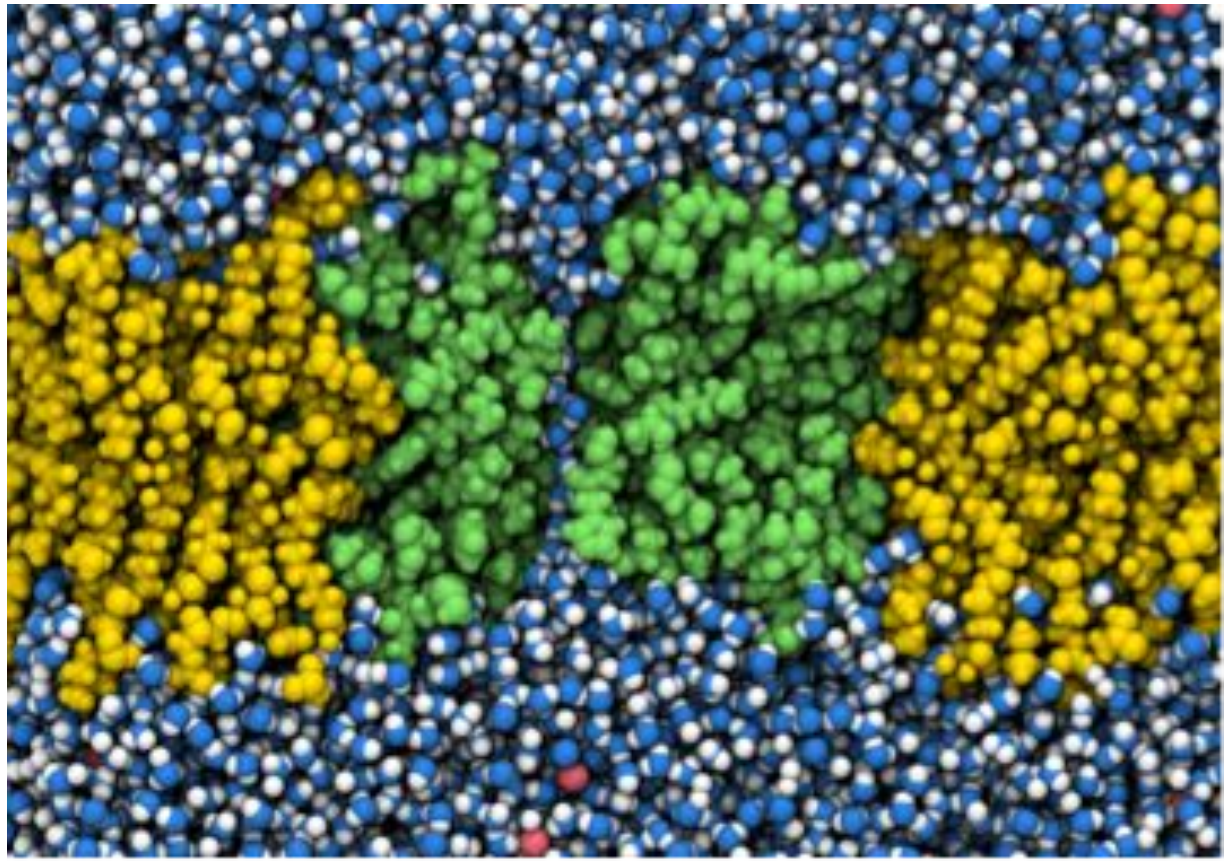
- AWH can give exponential acceleration of barrier crossings
- We can also use multiple walkers:
  - typically exchange data every 100 steps
    - note: at 0.5 ms per step this is **every 50 ms**

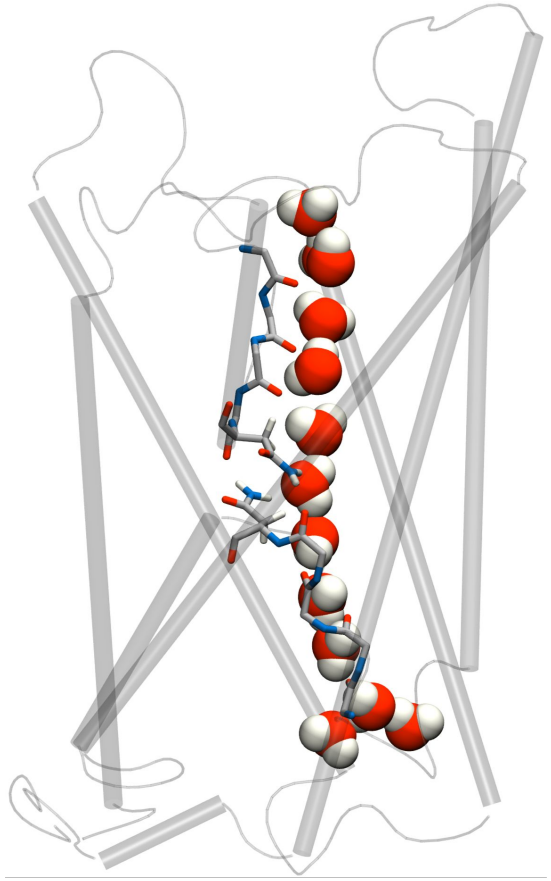
small system:
10000 atoms,
1 walker/node





Membrane protein: Aquaporin selectivity for water vs ammonia



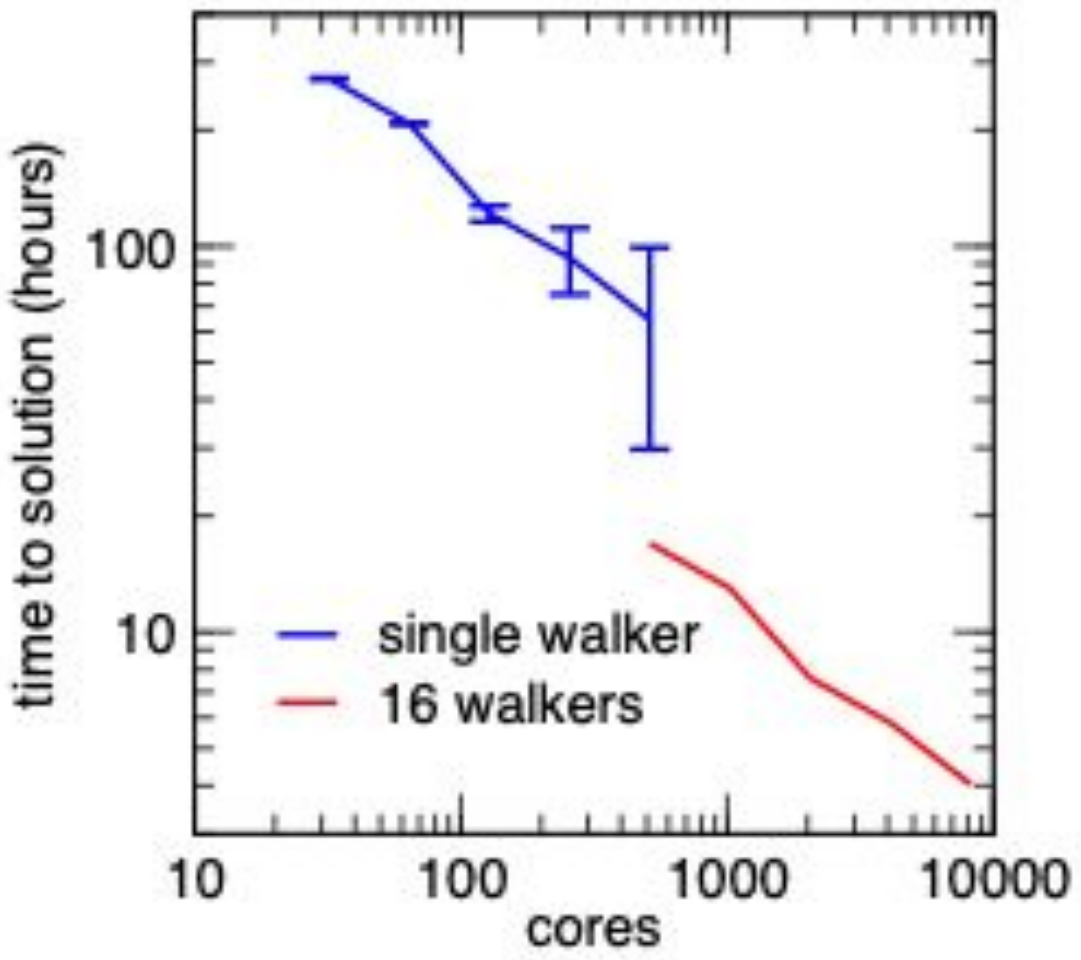


Aquaporin performance

system size: 80000 atoms

Using 16 walkers instead of 1:

- 4 times faster at 512 cores (avoids bad strong scaling)
- Time to solution at 8192 cores down to 4 hours!



bioexcel

Centre of Excellence for Computational Biomolecular Research

BioExcel Partners 2019

Horizon 2020
European Union Funding
for Research & Innovation

BioExcel is funded by the European Union Horizon 2020 program under grant agreements 675728 and 823830.