Geometrical free-energy methods: strengths and limitations

Giacomo Fiorin, Fabrizio Marinelli

Temple Materials Institute, Philadelphia, PA

National Heart, Lung and Blood Institute, Bethesda, MD

Enhanced sampling and Free-energy Workshop
Urbana-Champaign, September 25-29 2017

Different methods to compute FEs

Thermodynamic integration (Kirkwood, 1935):

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

Free-energy perturbation (Zwanzig, 1954)

$$e^{-\Delta F/kT} = \left\langle e^{-(U_{\lambda=1} - U_{\lambda=0})/kT} \right\rangle$$

Probability-based (umb. samp., MBAR, metadyn…)

$$e^{(F_1 - F_0)/kT} = (\Sigma_{[\lambda=1]} P_i) / (\Sigma_{[\lambda=0]} P_j)$$

Jarzynski's identity (Jarzynski, 1997, used with SMD)

$$e^{-(F_1 - F_0)/kT} = \left\langle e^{-W_{0\rightarrow 1}/kT} \right\rangle_{non-eq}$$

Thermodynamic integration

Choose a “reaction coordinate” λ which goes from our chosen initial state (λ = 0) to the final state (λ = 1) (Kirkwood, 1935).

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda$$

Now how to obtain the derivative of the free energy dF/dλ?

Thermodynamic integration

$$\Delta F = \int_{\lambda=0}^{\lambda=1} \frac{dF}{d\lambda} d\lambda = \int_0^1 \frac{d}{d\lambda}(-kT \times \ln(Z)) d\lambda =$$

$$= \int_0^1 -kT \times \frac{1}{Z} \frac{d}{d\lambda}\left(\Sigma_i e^{-Ei/kT}\right) d\lambda =$$

$$= \int_0^1 -kT \times \frac{1}{Z}\left(\Sigma_i e^{-Ei/kT} \frac{-1}{kT} \frac{dE_i}{d\lambda}\right) d\lambda =$$

$$= \int_0^1 \frac{1}{Z}\left(\Sigma_i e^{-Ei/kT} \frac{dE_i}{d\lambda}\right) d\lambda = \int_{\lambda=0}^{\lambda=1} \left\langle \frac{dE}{d\lambda} \right\rangle d\lambda$$

Thermodynamic integration

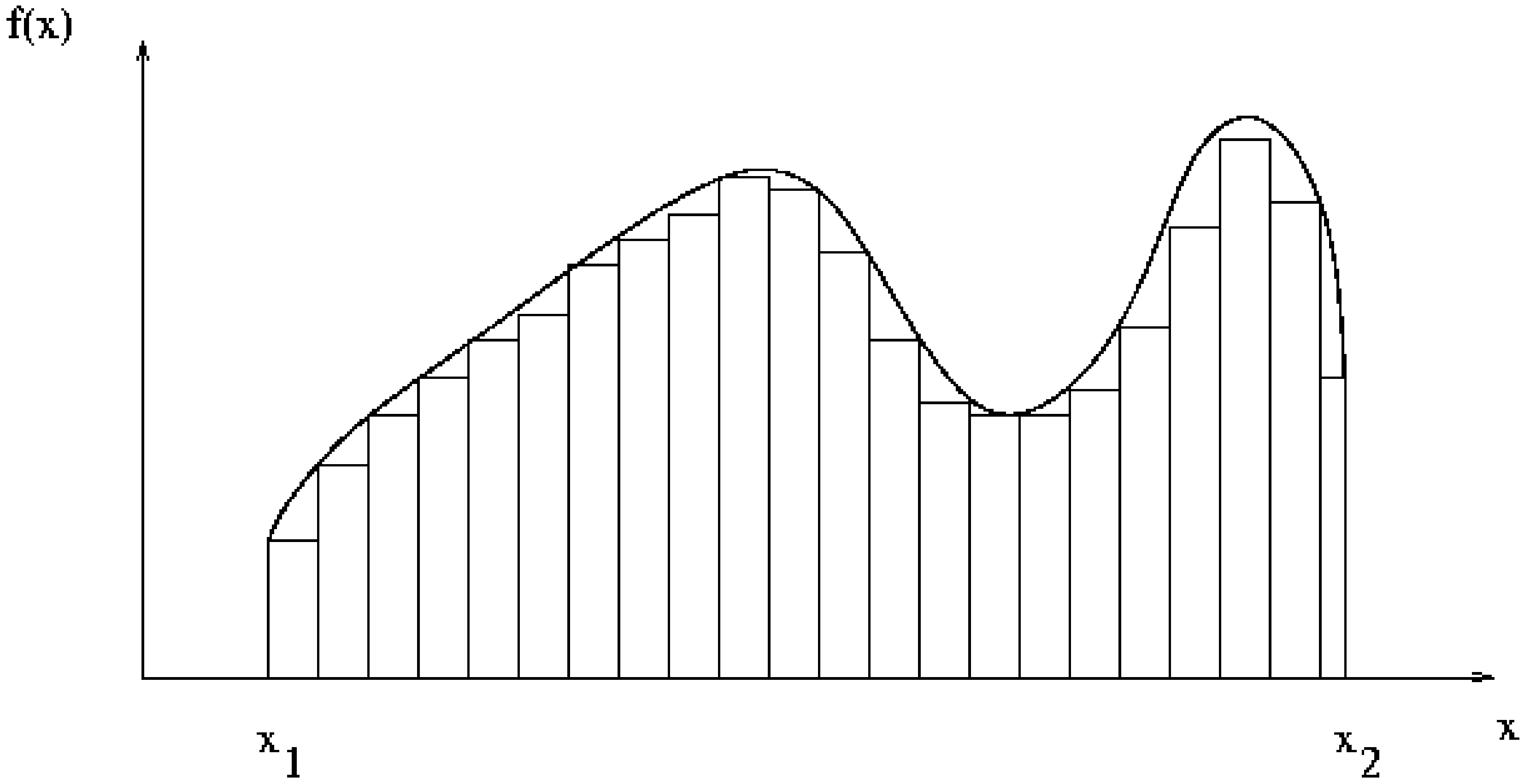
Choose a “reaction coordinate” λ which goes from our chosen initial state (λ = 0) to the final state (λ = 1) (Kirkwood, 1935).

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

where $f_\lambda = \langle dU/d\lambda \rangle$ is the *“thermodynamic force”* acting on the variable λ.

Note that the integral is numeric, i.e. a sum.

Thermodynamic integration



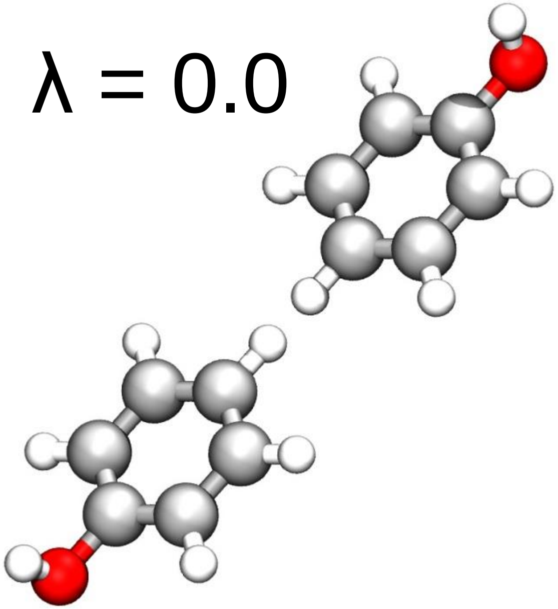
Two approaches for TI

- **Constrained approach**: for each λ = 0, 0.01, 0.02, … carry out *a simulation at constant λ-value*, calculate f_λ = ⟨dU/dλ⟩, integrate it.
- **Unconstrained approach**: same as above, but letting the system *diffuse freely across λ-values while collecting* ⟨dU/dλ⟩ *on a grid*, integrate it.

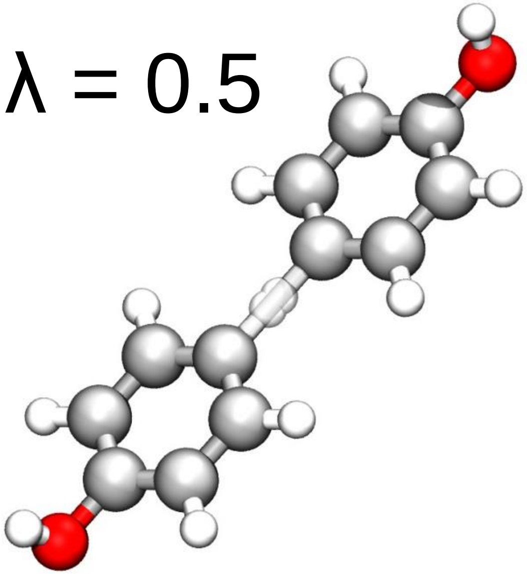
When λ is easy to choose (1)

Alchemical transformations: we simulate two systems, A and B, with a combined energy function: $U(\lambda) = U_A\times(1-\lambda) + U_B\times\lambda$

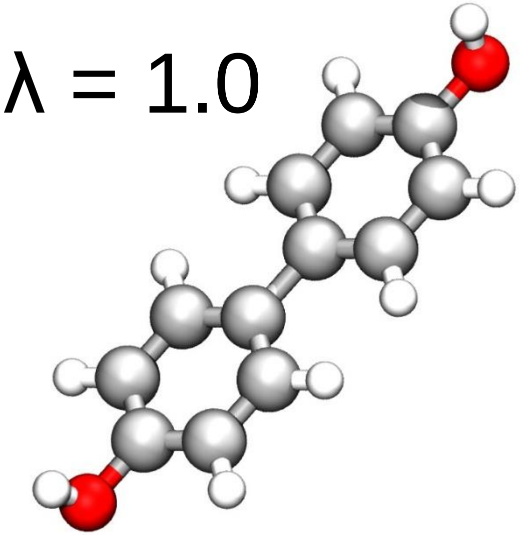
λ = 0.0



λ = 0.5

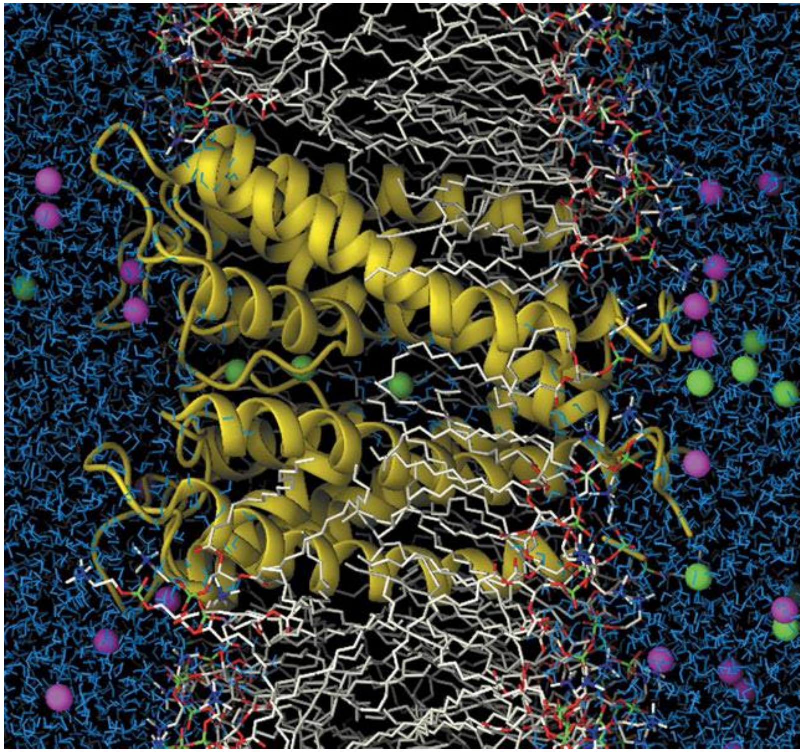


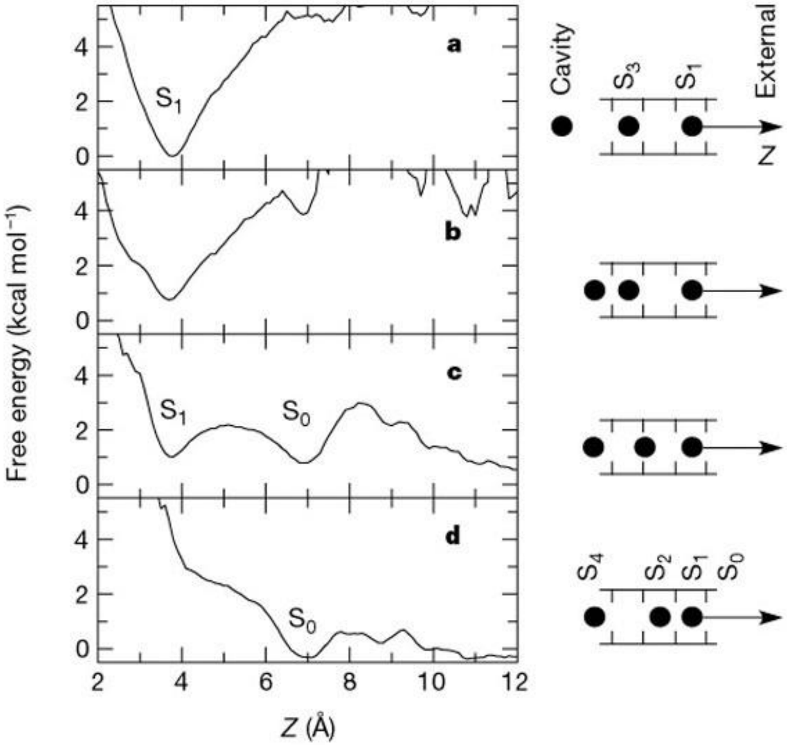
λ = 1.0



When λ is easy to choose (2)

Permeation through a channel: set λ equal to the trans-membrane coordinate.

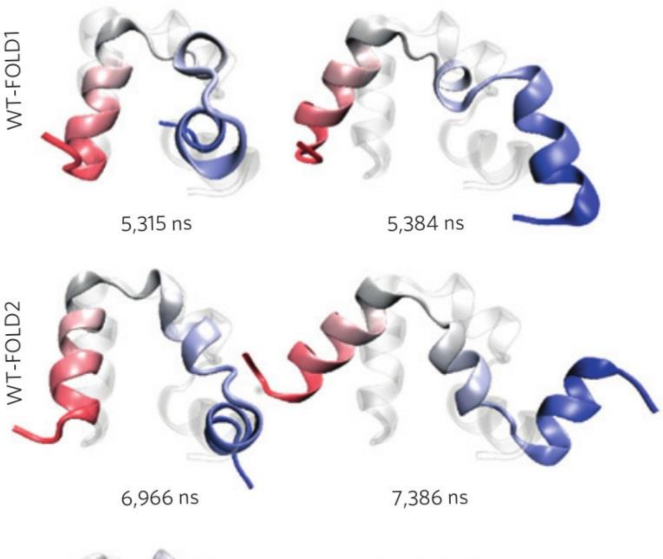




(Berneche & Roux, Nature **414**:73-77, 2001)

When λ is NOT easy to choose

Protein folding: good luck…



Freddolino et al, Nature Physics **6**:751-758 (2010)

Inverse gradients in TI

If we write the total derivative $f_\lambda = \langle dU/d\lambda \rangle$ in terms of Cartesian forces:

$$\Delta F = \int \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda = \int \left\langle \frac{\partial U}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \lambda} \right\rangle d\lambda$$

Two issues with the "inverse gradient" $d\mathbf{x}/d\lambda$:

1) It is not unique: its Cartesian components that are orthogonal to $\partial U/\partial \mathbf{x}$ have no effect.
2) It is rarely constant with λ, because $\lambda(\mathbf{x})$ is rarely a linear function: we need to calculate it.

How to use the inverse gradients?

The constrained integral:

$$\int \left\langle \frac{\partial U}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \lambda} \right\rangle d\lambda = \int \left(\int_{\mathrm{x} \in \lambda} \frac{\partial U}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial \lambda} p(\mathbf{x}) d\mathbf{x} \right) d\lambda$$

can be simplified by *changing coordinates* from **x** to (λ,**q**) :

$$\int \left(\frac{\partial U}{\partial \lambda} + \frac{\partial U}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial \lambda} \right) d\lambda d\mathbf{q}$$

which is easiest if U is a simple expression of λ (e.g. if λ is the radial distance).

Jacobian term in TI

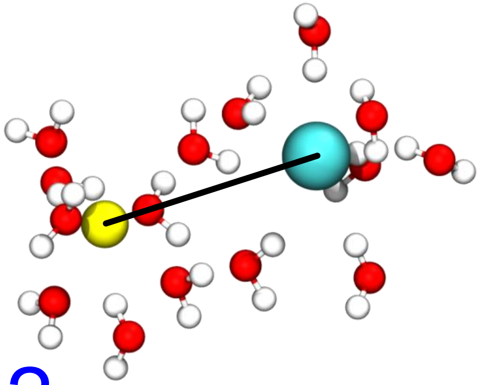
Any change in coordinates for a multi-dimensional integral comes with a Jacobian term. See Carter et al (1989):

$$\left\langle \frac{dU}{d\lambda} \right\rangle = \frac{\partial U}{\partial \lambda} - kT \frac{\partial \ln|J(\lambda,\mathbf{q})|}{\partial \lambda}$$

where the second term is purely geometric (does not depend on the internal energy function U).

Jacobian of radial distance

$$\left\langle \frac{dU}{d\rho} \right\rangle = \left\langle \frac{\partial U}{\partial \rho} \right\rangle - kT \frac{\partial \ln |J(\rho, \mathbf{q})|}{\partial \rho} =$$

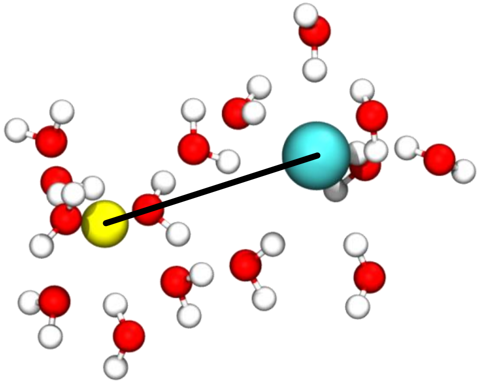


$$= \left\langle \frac{\partial U}{\partial \rho} \right\rangle - kT \frac{\partial \ln\left(\rho^2 \sin(\theta)\right)}{\partial \rho} = \left\langle \frac{\partial U}{\partial \rho} \right\rangle - kT \frac{2}{\rho}$$

If we integrate it from ρ = 0 to ρ = r, the Jacobian term will give −2kT×ln(r).

We shouldn't forget it when computing binding free energies by a distance variable.

Wait though: what about g(r)?



We had learned from textbooks that:

$F(r) = -kT \times \ln(g(r))$

and because $g(\infty) \rightarrow 1$, $F(\infty) \rightarrow 0$.

Note 1: g(r) is usually computed from an *infinite number of ion pairs*. *For just one specific pair*, the PMF ***does go as −2kT×ln(r)***.

Note 2: *it's not just a definition problem*, as you will notice if you try calculating the PMF of a RMSD variable near RMSD = 0.

Different methods to measure FEs

Thermodynamic integration (Kirkwood, 1935):

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

Free-energy perturbation (Zwanzig, 1954)

$$e^{-\Delta F/kT} = \langle e^{-(U_{\lambda=1} - U_{\lambda=0})/kT} \rangle$$

Probability-based (umb. samp., MBAR, metadyn…)

$$e^{(F_1 - F_0)/kT} = (\Sigma_{[\lambda=1]} P_i) / (\Sigma_{[\lambda=0]} P_j)$$

Jarzynski's identity (Jarzynski, 1997, used with SMD)

$$e^{-(F_1 - F_0)/kT} = \left\langle e^{-W_{0 \rightarrow 1}/kT} \right\rangle_{\text{non-eq}}$$

Free energy perturbation (FEP)

Calculating dU/dλ on hundreds of points may be difficult, or expensive.

Zwanzig equation (finite difference):

$$e^{-\Delta F/kT} = \left\langle e^{-(U_B - U_A)/kT} \right\rangle$$

and now where we calculate ⟨·⟩ is crucial. By convention we use a simulation of “A”:

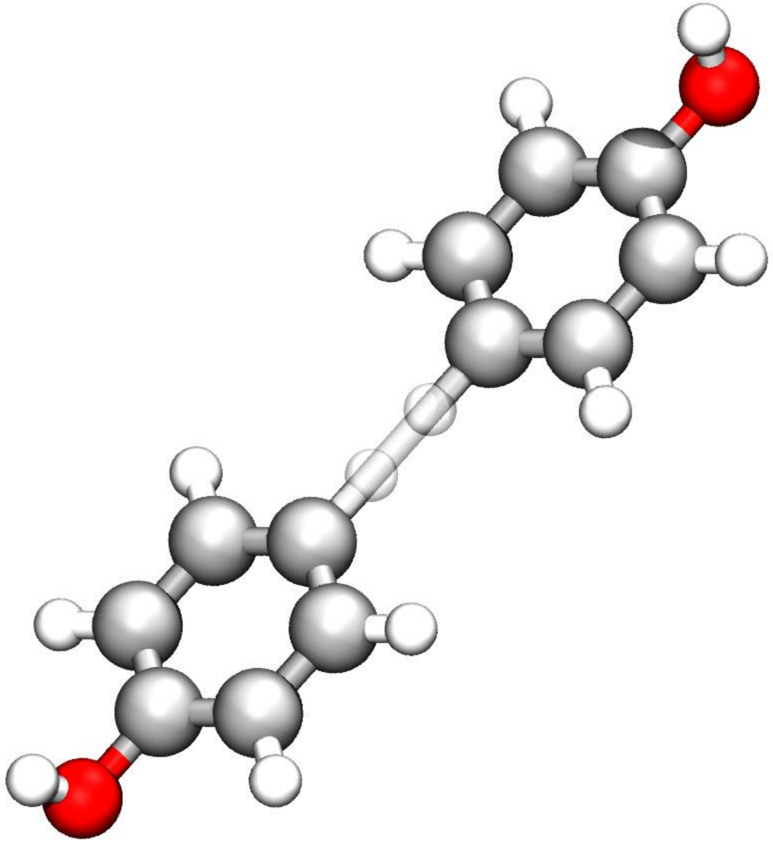
$$e^{-\Delta F/kT} = \left\langle e^{-(U_B - U_A)/kT} \right\rangle_A$$

(Alchemical) FEP

Calculate both U_A and U_B, but only move using forces from U_A (or one of the states anyway).

At λ = 0 (pure "A") U_B may be quite large, but it is only a problem of statistical convergence.

$$e^{-\Delta F/kT} = \left\langle e^{-(U_B - U_A)/kT} \right\rangle_A$$



Different methods to measure FEs

Thermodynamic integration (Kirkwood, 1935):

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

Free-energy perturbation (Zwanzig, 1954)

$$e^{-\Delta F/kT} = \left\langle e^{-(U_{\lambda=1} - U_{\lambda=0})/kT} \right\rangle$$

➡ **Probability-based** (umb. samp., MBAR, metadyn…)

$$e^{(F_1 - F_0)/kT} = (\Sigma_{[\lambda=1]} P_i) / (\Sigma_{[\lambda=0]} P_j)$$

Jarzynski’s identity (Jarzynski, 1997, used with SMD)

$$e^{-(F_1 - F_0)/kT} = \left\langle e^{-W_{0 \rightarrow 1}/kT} \right\rangle_{non-eq}$$

Probability based methods

Canonical ensemble: for any microstate v,

$p(v) = e^{-E_v/kT} / \sum_i e^{-E_i/kT} = e^{-E_v/kT}/Z,$

where $Z = \sum_i e^{-E_i/kT} = e^{-F/kT}$

Considering only states at λ = 0 or λ = 1:

$e^{-(F_1-F_0)/kT} = Z_1/Z_0 =$

$= (Z \times \sum_{[\lambda=1]} p(i)) \,/\, (Z \times \sum_{[\lambda=0]} p(j)) =$

= **(# of times λ = 1) / (# of times λ = 0)**

Umbrella sampling

Some configurations (i.e. λ-values) are poorly sampled. Torrie and Valleau, 1977: *add a biasing function w(λ)* to sample them more often.

Example: if $U(\lambda=1) - U(\lambda=0) \approx 8$ kcal/mol
$\rightarrow p(\lambda=1) \approx 10^{-6}$, $p(\lambda=0) \approx 1$

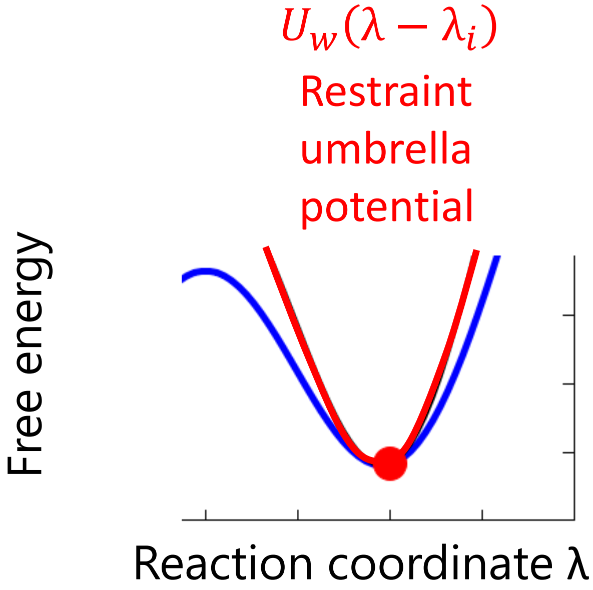
If we add the biasing potential*:*

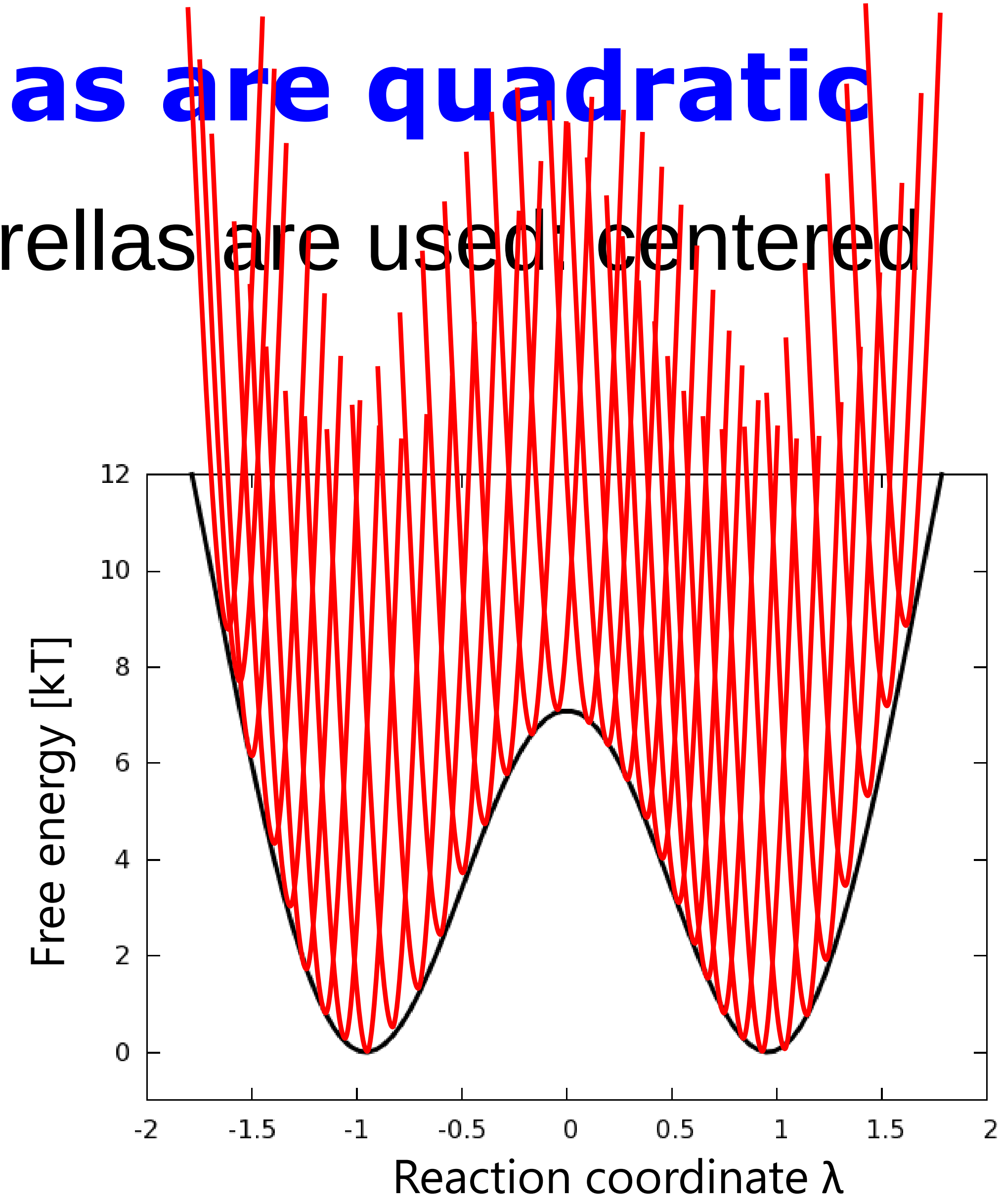
$U_w(\lambda) = -8$ kcal/mol + (16 kcal/mol)$\times(\lambda-1)^2$,
$w(\lambda=1) = 10^6$, $w(\lambda=0) = 10^{-6}$,

we now get $p(\lambda=1) \approx 1$, $p(\lambda=0) \approx 10^{-6}$

Most umbrellas are quadratic

Typically many umbrellas are used: centered at values λ_i:





Unbiasing umbrella sampling

A set of biasing functions w(λ) can sample all λ-values, but the equilibrium is changed.

Need to unbias all measurements to get the canonical distribution back.

$$e^{(F_1-F_0)/kT} = \left(\Sigma_{[\lambda=1]}\, p(i)\right) / \left(\Sigma_{[\lambda=0]}\, p(j)\right) =$$

$$= \left(\Sigma_{[\lambda=1]}\, p(i)\times\frac{w_i}{w_i}\right) / \left(\Sigma_{[\lambda=0]}\, p(j)\right) \approx$$

$$\approx \langle 1/w \rangle_{[w-\text{biased}]}$$

Free energy from U.S.

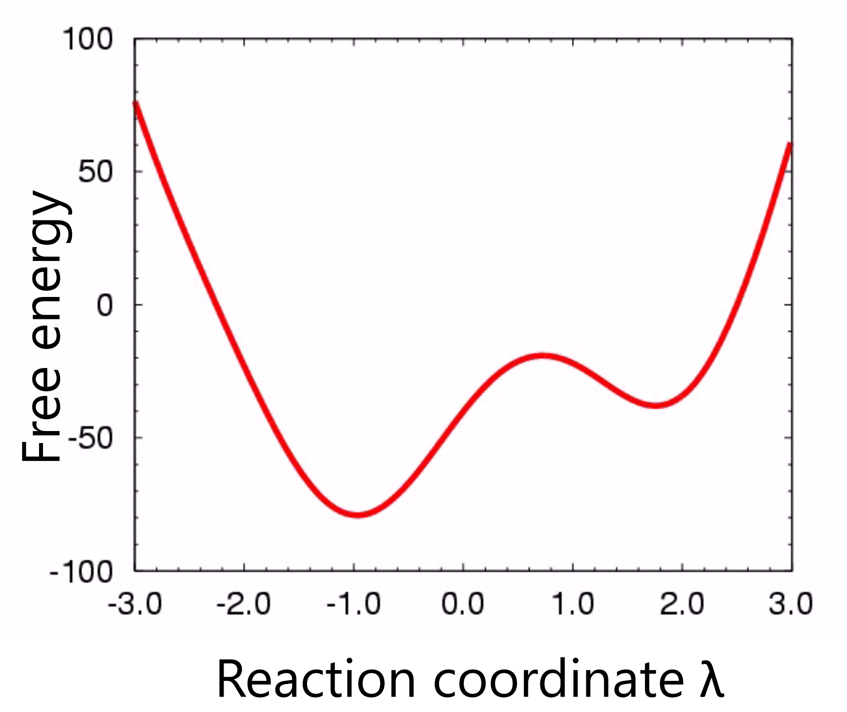
Because you’ll likely need different w(λ) for different regions of λ, unbiasing and merging all biased histograms $n_w(\lambda)$ can be laborious.

- **Weighted-histogram analysis method (WHAM) (Kumar et al, 1992)**: combine histograms to calculate p(λ) and its *(under)estimated error*. (Roux, 1995)
- **(Bayesian) bootstrap (Rubin, 1981)**: calculate a theoretical distribution of p(λ) that is *compatible* with the set of histograms.

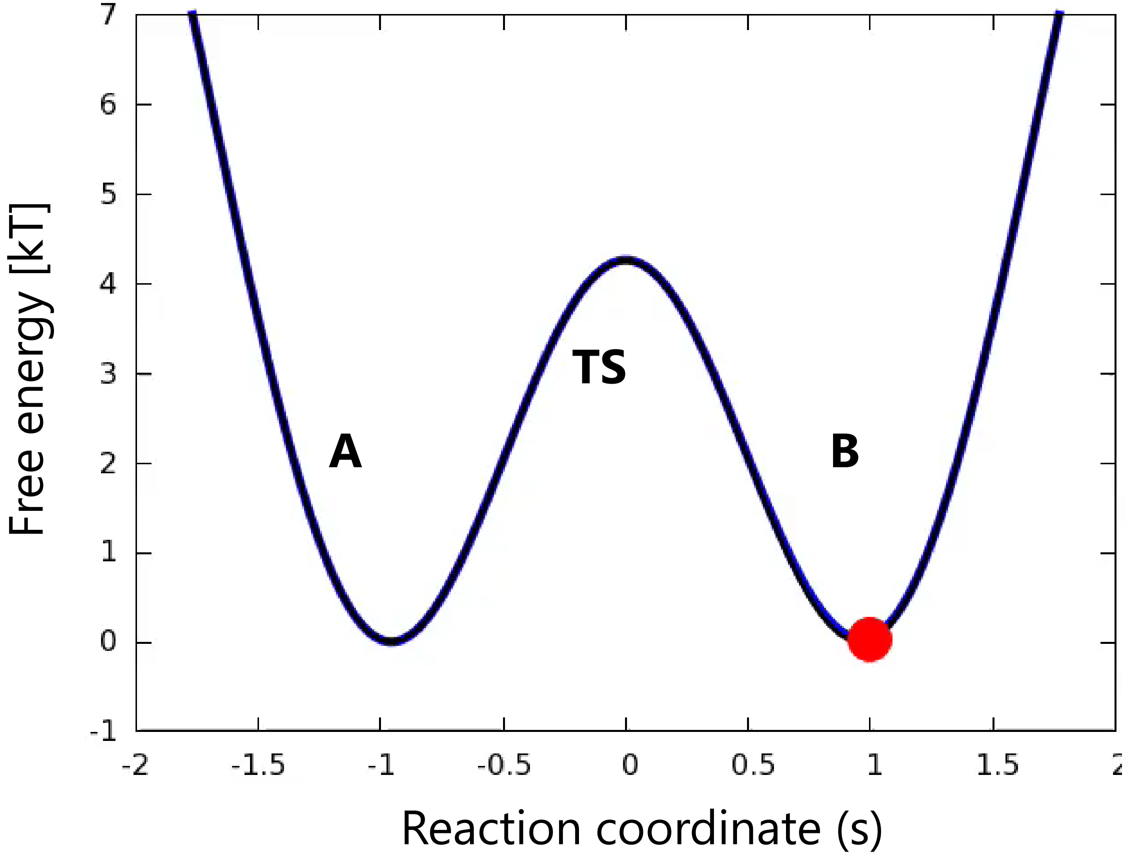
Adaptive methods: metadynamics

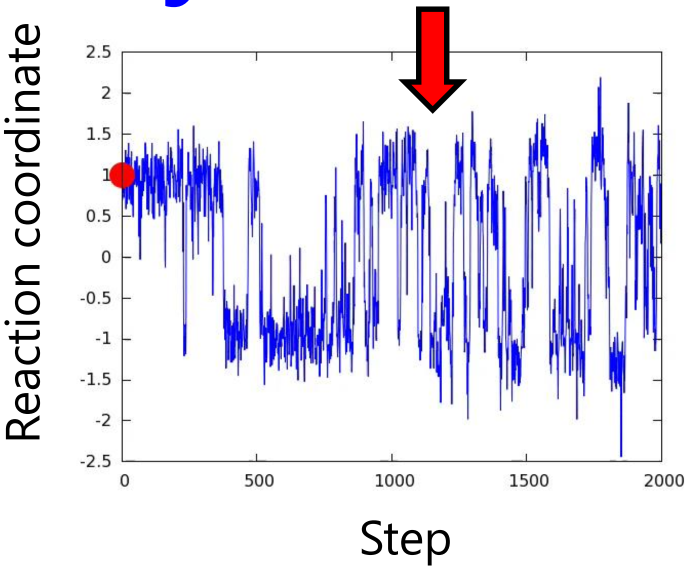
- Local elevation
  (Huber et al, 1994)
- Conformational flooding
  (Grubmüller, 1995)
- Metadynamics
  (Laio & Parrinello, 2002)

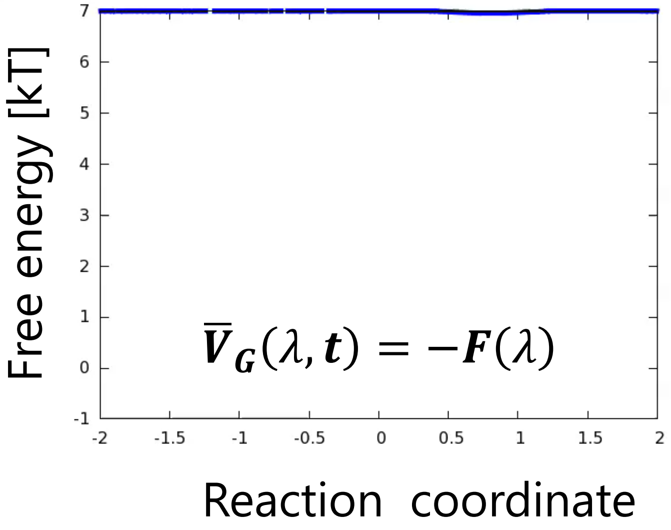
A repulsive potential (e.g. a Gaussian) raises the energy of states already visited, forcing the exploration of new ones.



Watch the trajectory!







$$\boldsymbol{\overline{V}_G(\lambda, t) = -F(\lambda)}$$

$$\boldsymbol{V_G(\lambda, t) = \sum_{t'=\tau, 2\tau\ldots}^{t} W\, exp\left\{-\frac{[\lambda - \lambda(\mathbf{X}_{t'})]^2}{2\sigma^2}\right\}}$$

Free energies from metadynamics

- The finite Gaussian functions create noise even after convergence. Practical approach: use the average of the instantaneous PMFs (but only after convergence).
- Well-tempered MTD artificially speeds up convergence: ensure that you know the largest barrier and use it accordingly.
- Multiple-walkers and replica-exchange MTD always improve sampling: ensure that replicas do travel through phase space.

Different methods to measure FEs

Thermodynamic integration (Kirkwood, 1935):

$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

Free-energy perturbation (Zwanzig, 1954)

$$e^{-\Delta F/kT} = \left\langle e^{-(U_{\lambda=1} - U_{\lambda=0})/kT} \right\rangle$$

Probability-based (umb. samp., MBAR, metadyn…)

$$e^{(F_1 - F_0)/kT} = (\Sigma_{[\lambda=1]} P_i) / (\Sigma_{[\lambda=0]} P_j)$$

➡ **Jarzynski’s identity** (Jarzynski, 1997, used with SMD)

$$e^{-(F_1 - F_0)/kT} = \left\langle e^{-W_{0 \rightarrow 1}/kT} \right\rangle_{\text{non}-\text{eq}}$$

Jarzynski's formula

So far we assumed that the system is always in equilibrium, with or without bias.

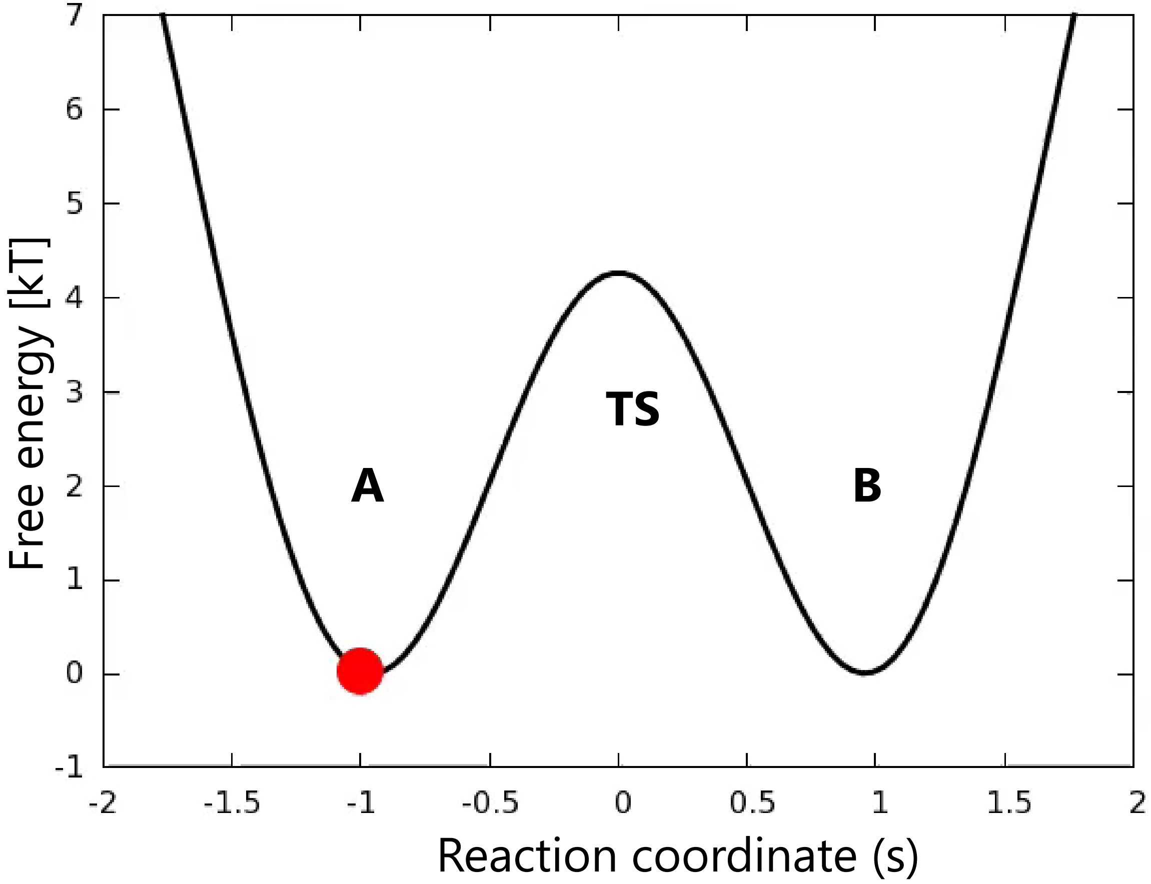
That requires an infinitely long simulation!

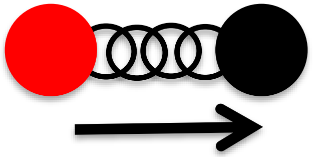
Jarzyinski's identity:

$$e^{-(F_1-F_0)/kT} = \left\langle e^{-W_{0\to1}/kT} \right\rangle_{\mathrm{non-eq}}$$

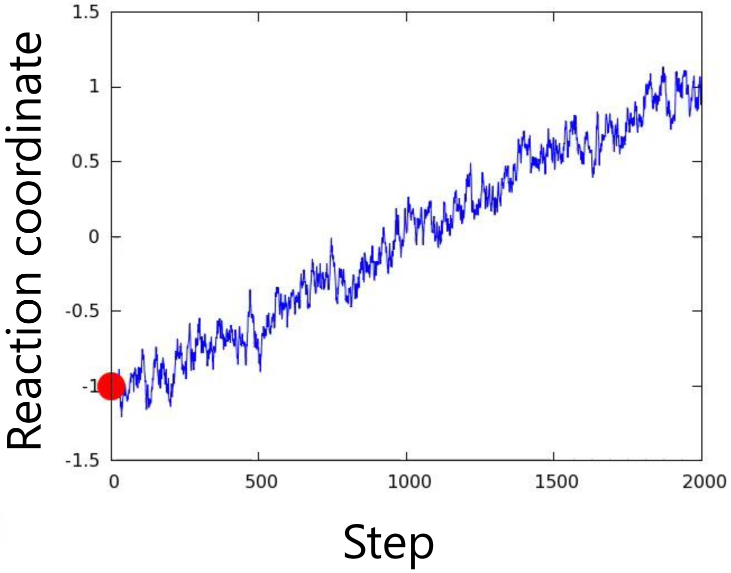
We can achieve the same with an ensemble of "short" simulations (maybe 10^{23} of them?)

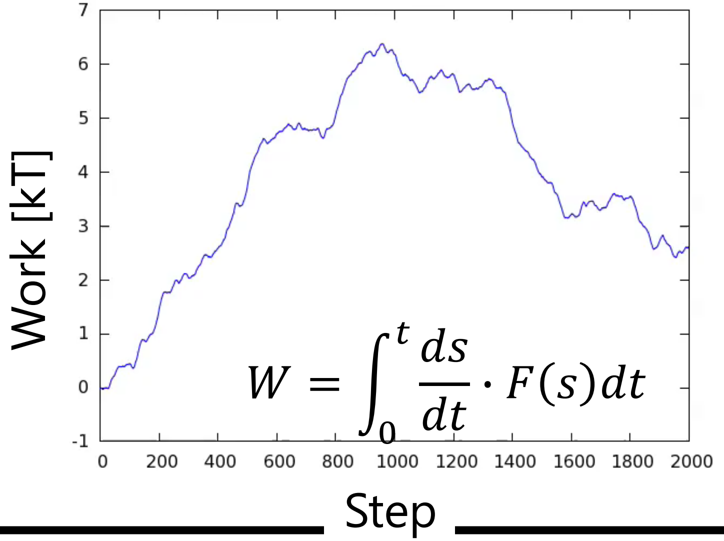
Steered MD: trajectory is set





Pulling Force



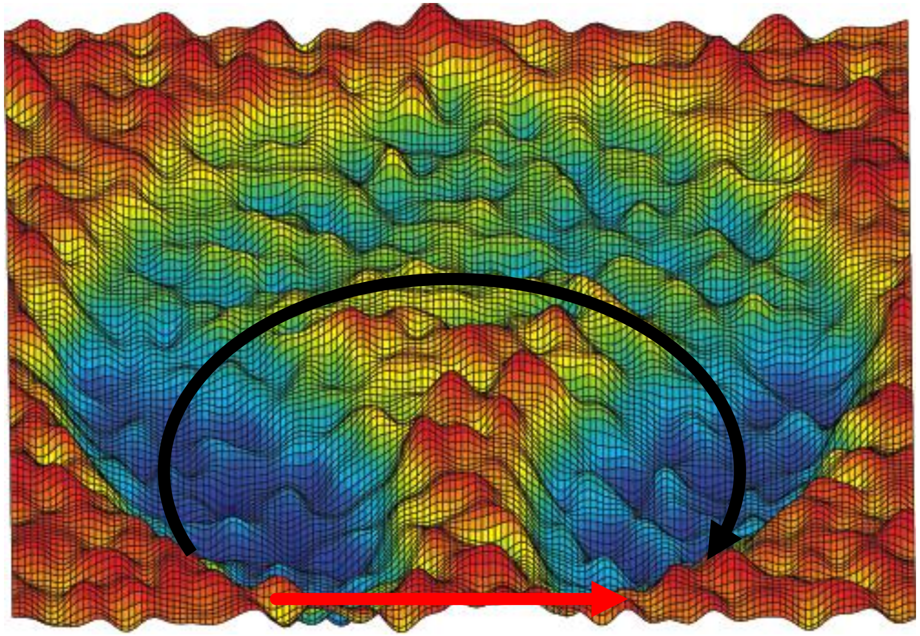


$$W = \int_0^t \frac{ds}{dt} \cdot F(s) dt$$

Don’t forget: many dimensions!

How to model a path through the space of configurations of a large molecular system is the *least trivial problem in simulation*.

The shortest path may not be a line!



E *et al*, PRB 2002

More than one dimension

Simplifying 1000s of degrees of freedom into “the” reaction coordinate λ is not impossible with time and effort, but still difficult!

Perhaps we don’t need to find “the one”.

We use two or more *“collective variables”* that define a small enough space: “the” reaction coordinate λ will live in this space.

(But not more than four or five, please.)

More than one dimension

Nearly all *"good"* free energy methods can deal with two or more collective variables.

Arguably this is most straightforward with the *probability-based methods*: the difficulty is only collecting a multi-dimensional histogram, $n(\lambda,\zeta)$:

$$n(\lambda_0,\zeta_0)/n(\lambda_1,\zeta_1) \;=\; p(\lambda_0,\zeta_0)/p(\lambda_1,\zeta_1) \;= \\ = e^{(F(\lambda_1,\zeta_1)-F(\lambda_0,\zeta_0))/kT}$$

How to use each method in NAMD?

Thermodynamic integration (Kirkwood, 1935):

Colvars, Alchemical module

Free-energy perturbation (Zwanzig, 1954)

Alchemical module

Probability-based (umb. samp., MBAR, metadyn…)

Colvars, TclForces, Plumed (w. patch)

Jarzynski's identity (Jarzynski, 1997, used with SMD)

Colvars, TclForces, Plumed (w. patch)

Final recommendations

- Aim for reproducibility: if you can't converge to the results of an existing study, beware of possible statistical flukes (yours or theirs!).
- Use automation to your advantage: there will always be more compute cores available.
- No conclusive evidence of one method being better *all the time*. *But for a chosen problem,* some can be better than others.
- And we can always combine them…

New Colvars feature: add-on TI

The **thermodynamic integration (TI)** free energy estimator is now **used by all biases** (not only ABF), when supported by the variable.

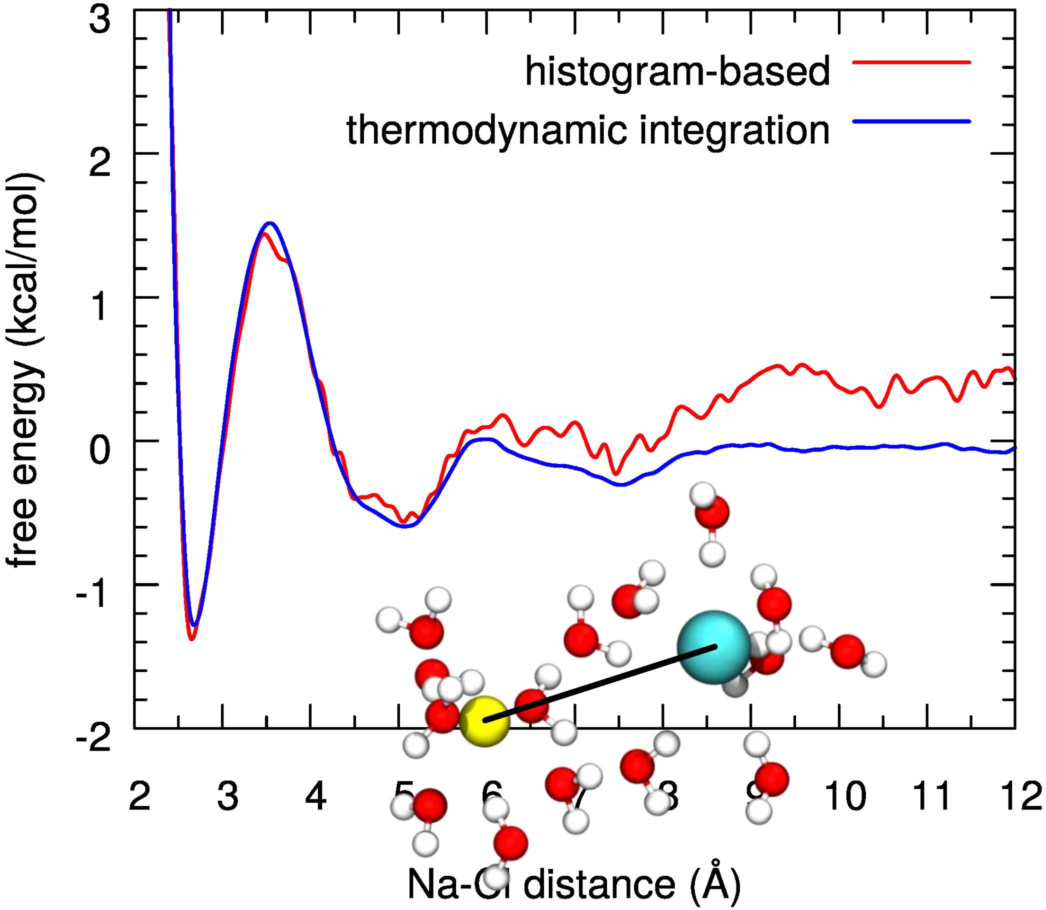
$$\Delta F = \int_{\lambda = 0}^{\lambda = 1} \frac{dF}{d\lambda} d\lambda = \int_{\lambda = 0}^{\lambda = 1} \left\langle \frac{dU}{d\lambda} \right\rangle d\lambda$$

Example: a metadynamics simulation produces:

- `output.pmf` (obtained from summing together the Gaussian functions).
- `output.ti.pmf` (obtained from projecting total atomic forces on collective variables).

The code will check whether the functions used supports internal force projection.

New Colvars feature: add-on TI



The Colvars module

- A library permanently linked to **NAMD**, **LAMMPS** (package: **USER-COLVARS**) and **VMD**.
- Implements differentiable ***collective variables / order parameters / reaction coordinates***.
- **Popular methods** with uniform syntax: *umbrella sampling, steered MD, metadynamics, Hamiltonian TI/FEP, conformational thermodynamic integration, adaptive biasing force (ABF), temp-accelerated CV sampling, string method in CV space.*
- ***Mostly, to escape comfortable energy minima.***

How to use a collective variable?

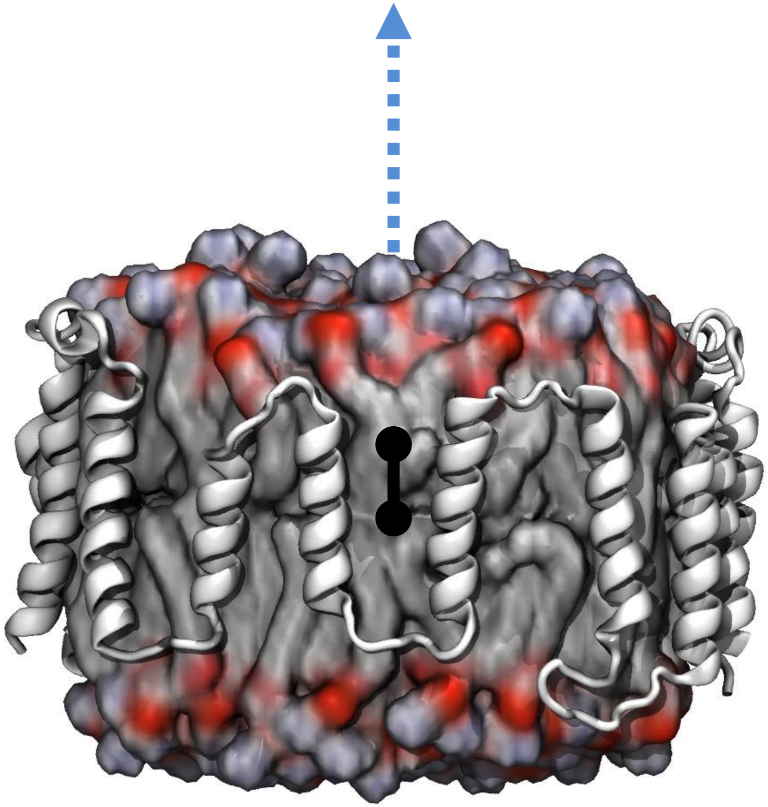
1) Choose an existing primitive function, or "component" (e.g. ξ = `rmsd { … }`).
2) Define a polynomial of components (e.g. $\xi = d_1 - d_2$ via `componentCoeff` = ±1, `componentExp` = 1).
3) Use the Lepton library by Peter Eastman (e.g. `customFunction` ξ = exp(–RMSD)).
4) Write your own function ξ in a scripting language (Tcl, and Python soon) via `scriptedFunction` (and ξ may even be the Cartesian coordinates!).
5) Write a new function or wrapper in C or C++ (*Colvars' source code is extensively documented*).

How many atoms?

ApoA1 benchmark on TACC Stampede.

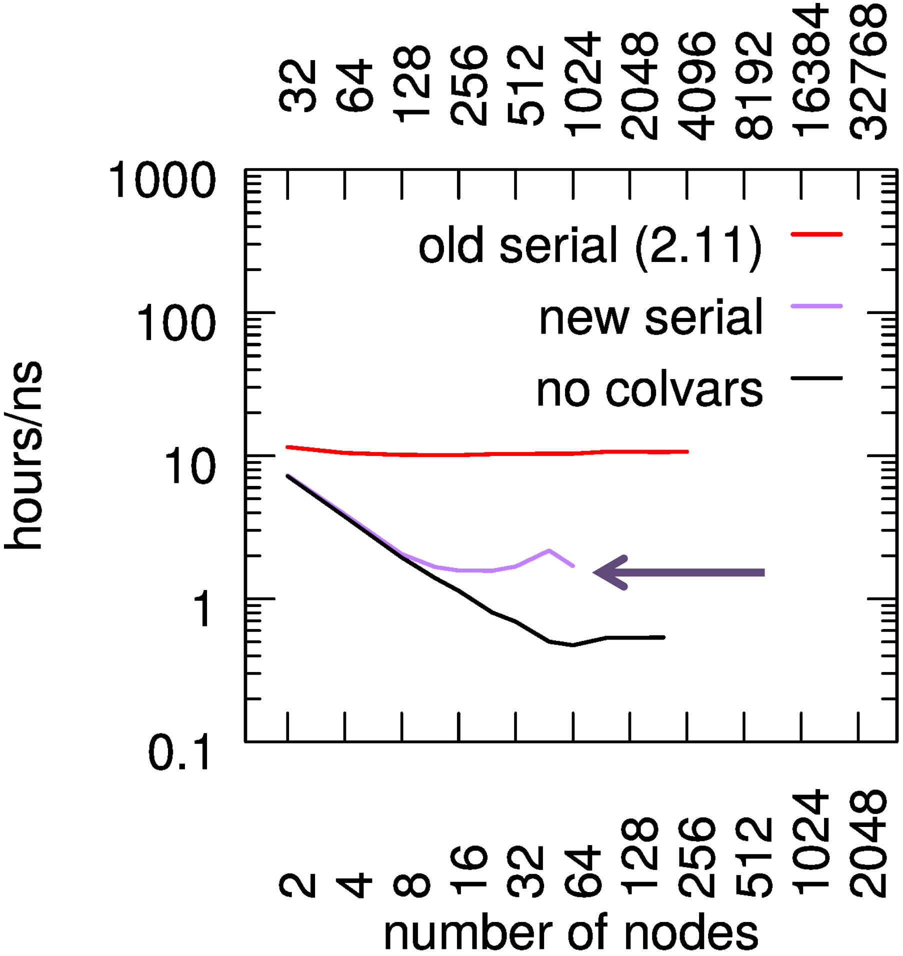
One collective variable: displacement between COMs of protein backbone (1600 atoms) and lipid atoms (8300).

Harmonic potential and system forces (i.e. the TI estimator of ABF) are both enabled.



Revised NAMD interface (2.12)

- Index arrays that are updated on demand as atoms are requested during a run (NAMD is scripted!).
- Inline functions to access coordinates and forces.
- Some variables are efficiently parallelized over nodes.
- Shared-memory processing of variables and biases.

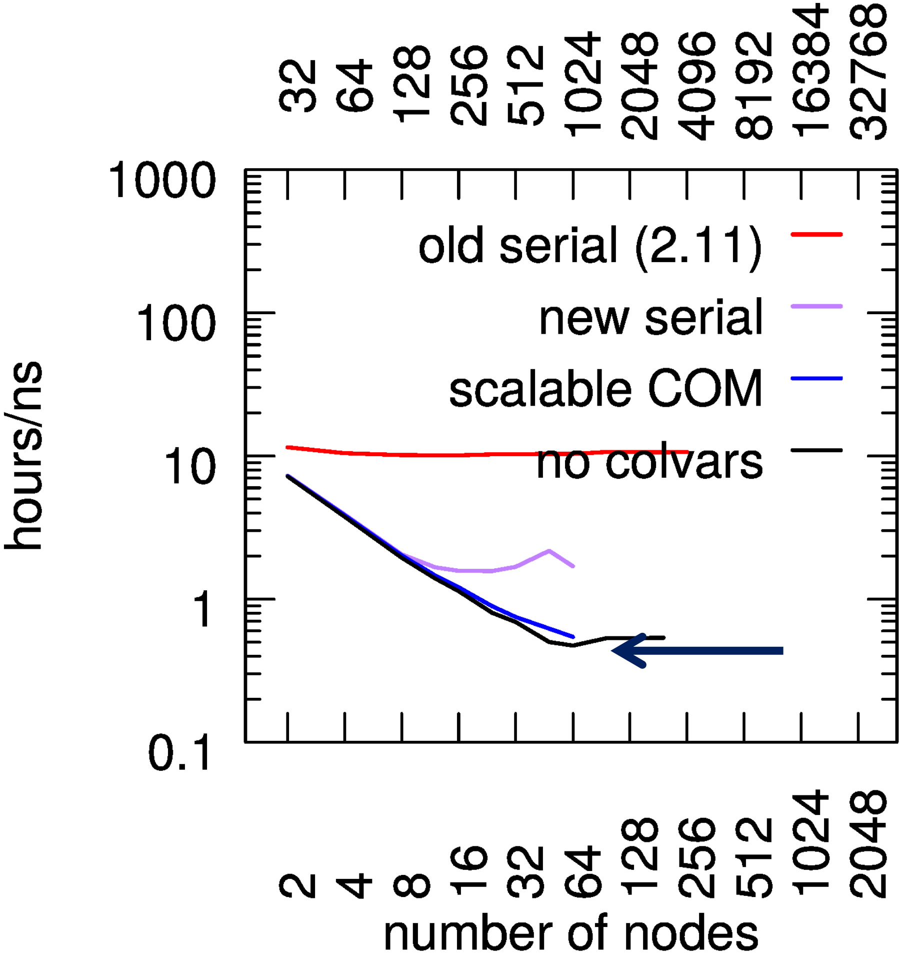


Compute COMs in parallel

Off-load calculation of COMs to NAMD computes: communicate few data structures.

For now parallelization applies only to functions of COMs: distances, angles, dihedrals, etc.
(More coming)

Enabled automatically in NAMD 2.12 (`scalable on`).



Shared-memory processing

Use CkLoop functionality recently added to Charm++ for NAMD, or standard OpenMP for LAMMPS.
Distribution of work applies to:

- Many colvar components/functions: useful for complex Tcl-based restraints (e.g. path collective variables);
- Many restraints or biases: structural refinements (e.g. replica-averaged ensembles);
- You can also break up large variables into chunks to compute in parallel, and combine with `componentCoeff`, `customFunction`, or `scriptedFunction`.
  (This may be done automatically in the future.)

Acknowledgements

- **Jérôme Hénin** (CNRS, Paris, France)
- Jim Phillips, John Stone, Chris Harrison, and Klaus Schulten (UIUC)
- Chris Chipot (UIUC $\rightleftharpoons$ U Nancy)
- Axel Kohlmeyer and Michael Klein (Temple U)
- Jeff Comer (KSU)
- *Anyone who made changes to the code.*
- **Fabrizio Marinelli, Lucy Forrest** and **José Faraldo-Gomez** (NIH)

Resources

- Repository for source code and accessories (scripts, minimal examples): https://github.com/Colvars/colvars
- Webpage with user guides (PDF & HTML): https://colvars.github.io/
- Ready-to-run input for VMD and NAMD: https://github.com/Colvars/examples
- Other tutorials in this workshop: ABF, protein-ligand binding, metadynamics, PMFs, string, REUS, AMS, complex pathways