Simulated mechanical unfolding of single proteins

W. Trevor King, Guoliang Yang

Department of Physics, Drexel University

Motivation

Understanding a protein's free energy landscape is important to efficiently modeling protein folding behaviors. Due to polymer nature of the polypeptides, the protein folding free energy landscape is rugged[1][2]. Knowledge of this landscape can enhance our understanding of the protein folding/unfolding mechanisms. We develop a simulation framework for modeling the stochastic unfolding process. The framework allows access to a number of models, including Bell's and Kramers' unfolding models, and the freely-jointed and worm-like chain tension models, allowing automatic selection of the most appropriate model combination for a particular experimental dataset. Particularly promising are the Kramers' unfolding model fits, which allow the extraction of arbitrary unfolding energy landscape functions.

Unfolding rate models

There are two models in common usage for modeling the unfolding of simple proteins under mechanical force. The more basic is Bell's model, which treats the transition as a thermalized transition in a two state system:

$$k = k_0 e^{\beta F \Delta x} \ ,$$

where k_0 is the unforced unfolding rate, F is the applied force, Δx is the distance between the folded state and the transition state, and β is the inverse thermal energy[3].

Kramers' model treats the process as quasi-thermalised, overdamped diffusion:

$$k^{-1} = \frac{1}{D} \int_{-\infty}^{x_e} dy e^{\beta[U(y) - Fy]} \int_{-\infty}^{y} dz e^{-\beta[U(z) - Fz]} \ ,$$

where $1/D = \beta M \gamma$ is the inverse diffusion constant (and the Einstein-Smoluchowski relation), γ is the damping constant, x_e is the position that we define as "escaped", and $U(x)$ is the free energy as a function of the unfolding coordinate. In general, $U(x)$ is an arbitrary function of x, but without much loss of practicality we can approximate it as a natural cubic spline with a few tens of anchor points [4].

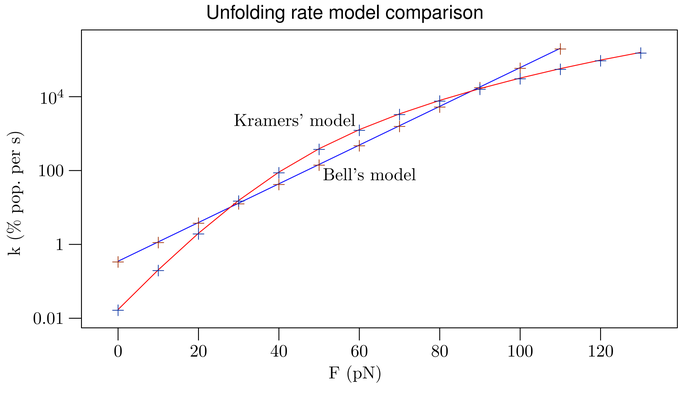
Because the double integral in the previous Kramers' expression for k can be computationally intensive, there exists the saddle-point-style approximation:

$$k = \frac{\omega_- \omega_+}{2\pi\gamma} \, e^{-\beta(\Delta U - F\Delta x)} = \frac{D\sqrt{U''(x_-)U''(x_+)}}{2\pi k_B T} \, e^{-\beta(\Delta U - F\Delta x)} \ ,$$

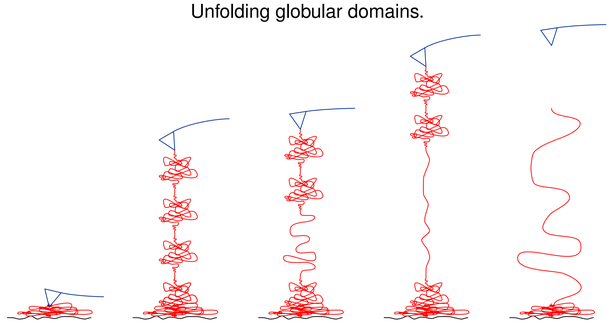
where $\omega_\pm \equiv \sqrt{U''(x_\pm)/M}$ are the oscillation frequencies at the well (x_-) and transition state (x_+), $\Delta U \equiv U(x_+) - U(x_-)$ is the energy barrier, and $\Delta x \equiv x_+ - x_-$[5][6].

By comparison with Bell's model and identifying $k_0 = \frac{\omega_- \omega_+}{2\pi\gamma} \, e^{-\beta\Delta U}$, we see that the two theories agree for $\beta\Delta U \gg 1$ (for which the saddle-point approximation is good) and $U''(x_\pm) \gg \Delta U / \Delta x^2$ (for which $x_\pm$ are insensitive to F)[4].

Unfolding rate model comparison



Unfolding globular domains.



Simulation mainloop

The simulation works on a list of *domains*. Each domain has a function determining it's unfolding rate under the current conditions (tension, temperature, ...). Each domain also is asigned folded and unfolded tension groups and any associated parameters. The tension groups allow for nonlinear tension models such as the worm-like chain. The main-loop of the function looks like

```
while (num_folded > 0)
  F = balance_tension(domains, environment, x);
  dt = determine_dt(domains, environment, vel, max_prob, max_dt);
  num_folded -= unfold_randomly(domains, environment, F, dt);
```

Tension models

The tension of unfolded protein domains, and occasionally even folded domains, are usually modeled as worm-like chains, continuous, elastically stiff entropic springs:

$$F = \frac{k_B T}{p} \left[\frac{1}{4} \left(\frac{1}{(1 - x/L)^2} - 1 \right) + \frac{x}{L} \right] \ ,$$

where p is the persistence length, and L is the contour length of the cable[7].

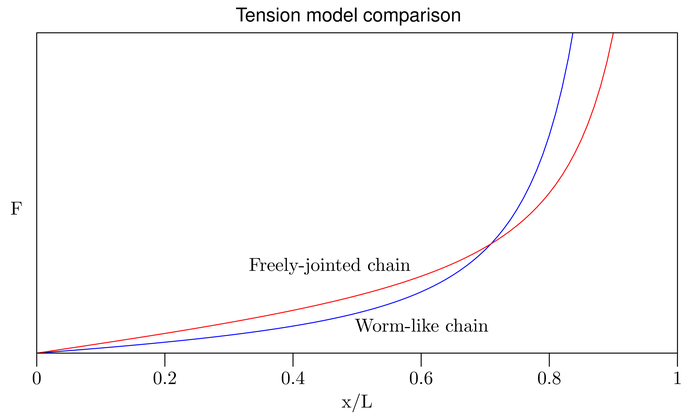
Occasionally proteins, and potentially folded domains, are modeled as freely-jointed chains, a series of rigid links with freely-rotating joints:

$$F = \frac{k_B T}{l} \, \mathcal{L}^{-1} \left(\frac{x}{L} \right) \ ,$$

where l is the length of a link, $L = Nl$ is the total length of the chain, and $\mathcal{L}^{-1}(\alpha)$ is the inverse Langevin function, where $\mathcal{L}(\alpha) \equiv \coth\alpha - \frac{1}{a}$ [7].

Because both these models are nonlinear, the main simulation loop cannot apply them to individual domains, but instead searches all the domains, and passes a list of, say, all worm-like chain domains to the worm-like chain handler. The handler goes through the list and extracts the total contour and persistence lengths, which it then plugs into the WLC function.

Tension model comparison



Tension balancing

The tension groups allow us to determine the tension of extending group i a distance x_i, but there may be multiple active tension groups (e.g. hookean AFM cantilever and WLC unfolded proteins). In order to determine the tension F of the entire chain for a stretched distance x, we need a method for determining the amount stretched by each group. For N groups, this amounts to solving

$$\begin{aligned} F_i(x_i) &= F \\ \sum_i x_i &= x \ , \end{aligned}$$

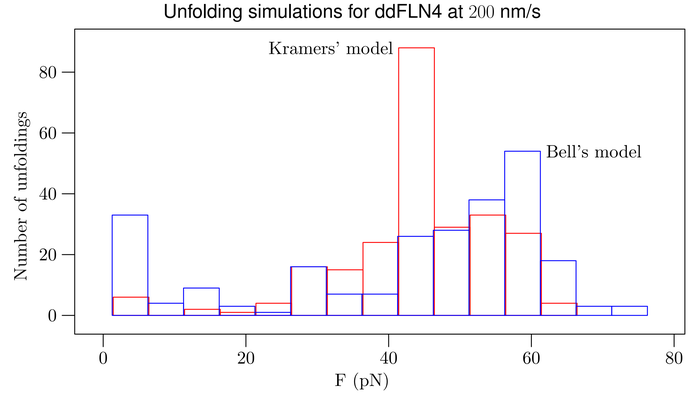
which is a system of $N+1$ possibly nonlinear equations for $N+1$ unknowns.

In general this is a root finding problem in $N+1$ dimensions (N should be small), but, by restricting ourselves to strictly monatonically increasing tension functions, the problem reduces to two-deep, nested one-dimentional root finding. From the first active group, we have $F = F_0(x_0)$. We can then solve $F_i(x_i) = F$ to get the appropriate x_i for the particular x_0. Therefore, $N-1$ one dimensional root searches will yield the one dimensional function $x(x_0)$, and one last one dimensional root search to solve $x(x_0) = x$ will yield the correct x_i and F.

This approach seems inefficient because of the nested rootfinding, but can be easily accelerated by storing computed solutions in a lookup table for future reference. We are currently working towards implementing this optimization.

Preliminary results

Unfolding simulations for ddFLN4 at 200 nm/s



References

[1] M. Mickler, R. I. Dima, H. Dietz, C. Hyeon, D. Thirumalai, and M. Rief, "Revealing the bifurcation in the unfolding pathways of GFP by using single-molecule experiments and simulations," *Proceedings of the National Academy of Sciences*, vol. 104, no. 51, pp. 20268–20273, 2007.

[2] J. N. Onuchic, N. D. Socci, Z. Luthey-Schulten, and P. G. Wolynes, "Protein folding funnels: the nature of the transition state ensemble.," *Fold Des*, vol. 1, no. 6, pp. 441–450, 1996.

[3] E. Evans and K. Ritchie, "Strength of a weak bond connecting flexible polymer chains.," *Biophys J*, vol. 76, pp. 2439–2447, May 1999.

[4] M. Schlierf and M. Rief, "Single-molecule unfolding force distributions reveal a funnel-shaped energy landscape.," *Biophys J*, vol. 90, pp. L33–L35, Feb. 2006.

[5] Y.-J. Sheng, S. Jiang, and H.-K. Tsao, "Forced kramers escape in single-molecule pulling experiments," *The Journal of Chemical Physics*, vol. 123, no. 9, p. 091102, 2005.

[6] P. Hänggi, P. Talkner, and M. Borkovec, "Reaction-rate theory: fifty years after kramers," *Rev. Mod. Phys.*, vol. 62, pp. 251–341, Apr 1990.

[7] J. W. Hatfield and S. R. Quake, "Dynamic properties of an extended polymer in solution," *Phys. Rev. Lett.*, vol. 82, pp. 3548–3551, Apr 1999.