Simulating protein evolution using Metropolis sampling

1 Review of the work by Sella and Hirsh

The following is based on the Sella-Hirsh formalism (G. Sella and A. E. Hirsh, PNAS 102:9541–9546, 2005).

We consider a population of size N evolving in discrete time-steps according to Wright-Fisher sampling. The overall mutation rate μ is sufficiently low such that $\mu N \ll 1$. In this case, we can describe the evolution of the entire population by keeping track of a single reference sequence. The fitness of genotype i is f_i, and the mutation rate from i to j is μ_{ji}. We assume a symmetric mutation matrix, $\mu_{ij} = \mu_{ji}$. The matrix C_{ij} defines the graph of genotypes connected by single-point mutations; $C_{ij} = 1$ if i and j differ by exactly one point mutation, and $C_{ij} = 0$ otherwise. Note that C_{ij} is also symmetric.

The fixation probability for a mutation from i to j is given by

$$\pi(i \to j) = \frac{1 - (f_i/f_j)^2}{1 - (f_i/f_j)^{2N}} \,. \tag{1}$$

To simulate the evolutionary process, all we have to do is start at a genotype i, randomly draw a mutation j, accept it with probability $\pi(i \to j)$, and repeat. Time increases by one unit every time we draw a mutation, regardless of whether we accept it or not.

However, note that in the limiting case of a neutral mutation, where $f_i = f_j$, the fixation probability is $1/N$:

$$\lim_{f_j \to f_i} \pi(i \to j) = \lim_{x \to 1} \frac{1 - x^2}{1 - x^{2N}} = \lim_{x \to 1} \frac{2x}{2Nx^{2N-1}} = \frac{1}{N} \,. \tag{2}$$

Therefore, if a fitness landscape has a lot of neutral or nearly-neutral genotypes, on average we have to test N mutations to accept one. Since we usually want N to be at least a 1000 or so, this is a large number of mutations we throw away, in particular if generating and testing mutations is slow. Note that this problem disappears if all mutations are either neutral or lethal, because then all non-zero fixation probabilities are $1/N$. In this case, we can advance time N units at once and simply accept every viable mutation we generate.

The question now is whether we can be more clever and somehow accept neutral mutations at a faster rate even in the general case. To investigate this, we first define the Markov process that we are modeling. It is described by the matrix W_{ji},

$$W_{ji} = \begin{cases} NC_{ji}\mu_{ji}\pi(i \to j) & \text{for } i \neq j \\ 1 - \sum_{k \neq i} W_k i & \text{for } i = j \end{cases} \,, \tag{3}$$

which gives the transition probability from state i to state j in the Markov chain. This Markov process has stationary frequencies

$$P_i = \frac{f_i^{2N-2}}{\sum_j f_j^{2N-2}} \,, \tag{4}$$

which follows from the fact that

$$\frac{W_{ji}}{W_{ij}} = \frac{f_j^{2N-2}}{f_i^{2N-2}} = \frac{P_j}{P_i} \,. \tag{5}$$

Another way to express this is to say that detailed balance is satisfied,

$$W_{ji} P_i = W_{ij} P_j \,. \tag{6}$$

To derive Eq. (5), we note that

$$\begin{aligned} \frac{W_{ji}}{W_{ij}} = \frac{\pi(i \to j)}{\pi(j \to i)} &= \frac{\frac{f_j^2 - f_i^2}{f_j^2}}{\frac{f_j^{2N} - f_i^{2N}}{f_j^{2N}}} \frac{\frac{f_i^{2N} - f_j^{2N}}{f_i^{2N}}}{\frac{f_i^2 - f_j^2}{f_i^2}} \\ &= \frac{\frac{1}{f_j^2}}{\frac{1}{f_j^{2N}}} \frac{\frac{1}{f_i^{2N}}}{\frac{1}{f_i^2}} = \frac{f_j^{2N-2}}{f_i^{2N-2}} \,. \end{aligned} \tag{7}$$

Note how the symmetric matrices C_{ij} and μ_{ij} cancelled in this derivation. This implies that we can rescale W_{ij} with any symmetric matrix and still satisfy detailed balance/have the same steady-state frequencies.

2 Re-scaling rates along all edges in the genotype graph

We now rescale all non-zero paths in the matrix W_{ij}. Specifically, we write

$$W'_{ij} = \tau_{ij} W_{ij} \qquad \text{for } i \neq j \tag{8}$$

and $W'_{ii} = 1 - \sum_{k \neq i} W'_{ki}$. We define τ_{ij} as

$$\tau_{ij} = \begin{cases} \frac{1}{\pi(i \to j)} & \text{if } f_j > f_i \\ \frac{1}{\pi(j \to i)} & \text{otherwise} \end{cases} \,. \tag{9}$$

Note that $\tau_{ij} = \tau_{ji}$ by definition. The scaling factor τ_{ij} is defined such that we are rescaling transitions along the (i, j) edge with the inverse of the probability of fixing the higher-fitness genotype when starting out at the lower-fitness genotype.

We can incorporate the scaling factor τ_{ij} into the fixation probability by writing

$$\pi'(i \to j) = \tau_{ij} \pi(i \to j) = \begin{cases} 1 & \text{for } f_j > f_i \\ \left(\frac{f_j}{f_i}\right)^{2N-2} & \text{otherwise} \end{cases} \,. \tag{10}$$

To derive the second case, we have made use of Eq. (7). Note that this new fixation probability looks exactly like the Metropolis-Hastings criterion. The transition from the lower-fitness to the higher-fitness genotype always happens with rate 1, and the transition from the higher-fitness to the lower-fitness genotype is exponentially suppressed.

The advantage of Eq. (10) over Eq. (1) for simulation is that we now accept every neutral or beneficial mutation with probability 1, and hence simulation becomes much more effective. The downside of Eq. (10) is that the evolutionary process has been distorted. In effect, we have increased the rate at which neutral mutations get accepted, and hence we'll see faster accumulation of mutations while evolving neutrally than we ordinarily would.

In the limit of neutral/lethal evolution, where every genotype has either fitness $f > 0$ or fitness 0, Eq. (10) tells us that we should accept all viable mutations with probability 1. Thus, this equation provides an alternative derivation of the idea, mentioned at the beginning of this document, that under purely neutral/lethal evolution we can accept all viable mutations.

3 Application to protein evolution

We now apply this formalism to the evolution of proteins under a soft threshold model. Under this model, protein fitness is modeled as a sigmoidal function. Very stable proteins have fitness 1, but fitness declines as stability passes through a threshold value. (This kind of model was first proposed by: P. Chen and E. I. Shakhnovich. Lethal Mutagenesis in Viruses and Bacteria. Genetics 183:639–650, 2009. See also Wylie and Shakhnovich PNAS 2011 and Serohijos et al., Cell Reports 2012.)

Specifically, we describe the fitness of protein i using a Fermi function,

$$f_i = \frac{1}{e^{\beta(\Delta G_i - \Delta G_{\text{thresh}})} + 1}\,, \tag{11}$$

where β is the inverse temperature, ΔG_i is the stability of protein i, and ΔG_{thresh} is the stability threshold at which the protein has lost 50% of its activity. (Chen and Shakhnovich used $\Delta G_{\text{thresh}} = 0$, but in general we may want to allow for other threshold values.)

It is useful to introduce the log-transformed fitness x_i,

$$x_i = \log(f_i) = -\log[e^{\beta(\Delta G_i - \Delta G_{\text{thresh}})} + 1]\,. \tag{12}$$

We can write the fixation probability (Eq. 10) in terms of x_i as

$$\pi'(i \to j) \approx \begin{cases} 1 & \text{for } x_j > x_i \\ e^{-2N(x_i - x_j)} & \text{otherwise} \end{cases}\,. \tag{13}$$

(We have now also made the approximation $2N - 2 \approx 2N$.) This formulation shows that for very unstable proteins ($\Delta G_i \gg \Delta G_{\text{thresh}}$), the population size N and the inverse temperature β have similar effects on fitness. In this case, we have

$$x_i = -\beta(\Delta G_i - \Delta G_{\text{thresh}}) \tag{14}$$

and hence

$$e^{-2N(x_i - x_j)} = e^{2N\beta(\Delta G_i - \Delta G_j)}\,. \tag{15}$$

(But keep in mind that this equation is not generally true.) Thus, to analyze the behavior of the system, it should be fine to set $\beta = 1$ and vary N only.

For multi-protein complexes, we can calculate an x_i for each protein and then add them up. This corresponds to multiplying the fitnesses f_i. Thus, if we have a complex with individual fitness contributions $x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(n)}$, we calculate the overall fitness as

$$x_i = \sum_k x_i^{(k)} \,. \tag{16}$$

4 Summary

Putting everything together, we calculate the fitness of a complex as

$$x_i = \sum_k x_i^{(k)} \,, \tag{17}$$

where

$$x_i^{(k)} = -\log[e^{\beta^{(k)}(\Delta G_i^{(k)} - \Delta G_{\text{thresh}}^{(k)})} + 1] \tag{18}$$

is the fitness contribution of chain k. (In the most general case, we may have chain-specific constants $\beta^{(k)}$ and $\Delta G_{\text{thresh}}^{(k)}$, though setting $\beta^{(k)} = 1$ for all k should be fine.) We accept a mutation from state i to state j with probability

$$p_{\text{accept}} = \begin{cases} 1 & \text{for } x_j > x_i \\ e^{-2N(x_i - x_j)} & \text{otherwise} \end{cases} \,, \tag{19}$$

where N is the simulated effective population size. N should be 1000 or larger. (If necessary, decrease β to get reasonable behavior.)