Notes on Monte Carlo simulations of proteins

Thomas Simonson and Georgios Archontis; January 27, 2015

1 Methodological background

1.1 The physical model

We consider a polymer (or polypeptide) of n amino acids. Its sequence S is written $S = t_1 t_2 \cdots t_n$, where t_i is the biochemical "type" of amino acid i. We assume that each amino acid i can take on a few different types t, t', ... that form a set T_i. By varying the types at each position, we generate a large collection of sequences $\mathcal{S} = \{S | t_i \in T_i\}$.

For each of these sequences, there are two classes of structures: "folded" and "unfolded". For the folded form, all the sequences S share the same, precise, known geometry for the polypeptide backbone; only the sidechain positions can vary. Specifically, the sidechain of each amino acid i can explore a few discrete conformations r, r', ... called "rotamers", which form a set R_i (around 10 per type t_i). The energy of any particular conformation depends on the sequence S and the particular set of rotamers: $E_f = E_f(\{t_i, r_i\})$. The structure of the unfolded form is not specified, but its energy E_{uf} is known, and has the simple additive form:

$$E_{uf}(S) = \sum_{i=1}^{n} E_{uf}(t_i). \tag{1}$$

The simulation system will explicitly include one copy each of the folded and unfolded proteins; their sequences and (sidechain) conformations will fluctuate during the Monte Carlo simulation.

1.2 Monte Carlo exploration

The goal is to generate a Markov chain of states [1–3], such that the states are populated according to a Boltzmann distribution, as detailed below. We perform a Monte Carlo exploration for this system, where one possible elementary move is a "mutation": we modify the sidechain type $t \to t'$ at a chosen position i in the folded protein, assigning a particular rotamer r' to the new sidechain. At the same time, we perform the reverse mutation in the unfolded form, $t' \to t$. (Fig. 1). A different, equivalent way to describe

the simulation is to start from a particular folded sequence and perform mutation moves, as follows. The "mutation" consists in taking a folded copy of sequence $t_1 t_2 ... t_i ... t_n$ and swapping it into the unfolded conformation; at the same time, we take an unfolded copy of the mutant sequence $t_1 t_2 ... t'_i ... t_n$ and swap it into its folded conformation (with a random choice for the r'_i rotamer) (Fig. 1). The corresponding energy change has the form:

$$\Delta E = \Delta E_f - \Delta E_{uf} = (E_f(...t'_i, r'_i...) - E_f(...t_i, r_i...)) - (E_{uf}(t'_i) - E_{uf}(t_i)) \qquad (2)$$

ΔE measures the stability change due to the mutation (for the given set of rotamers). Another possible move is to simply change a rotamer r_i in one particular sequence in its folded form; the energy change is $\Delta E = \Delta E_f = E(...t_i, r'_i...) - E(...t_i, r_i...)$. This procedure assumes that for each folded sequence sampled during a simulation, there is also an unfolded copy in the system. With this interpretation of the simulation, "mutation" moves can be seen as taking place in conformational space, so that ordinary statistical mechanics apply.

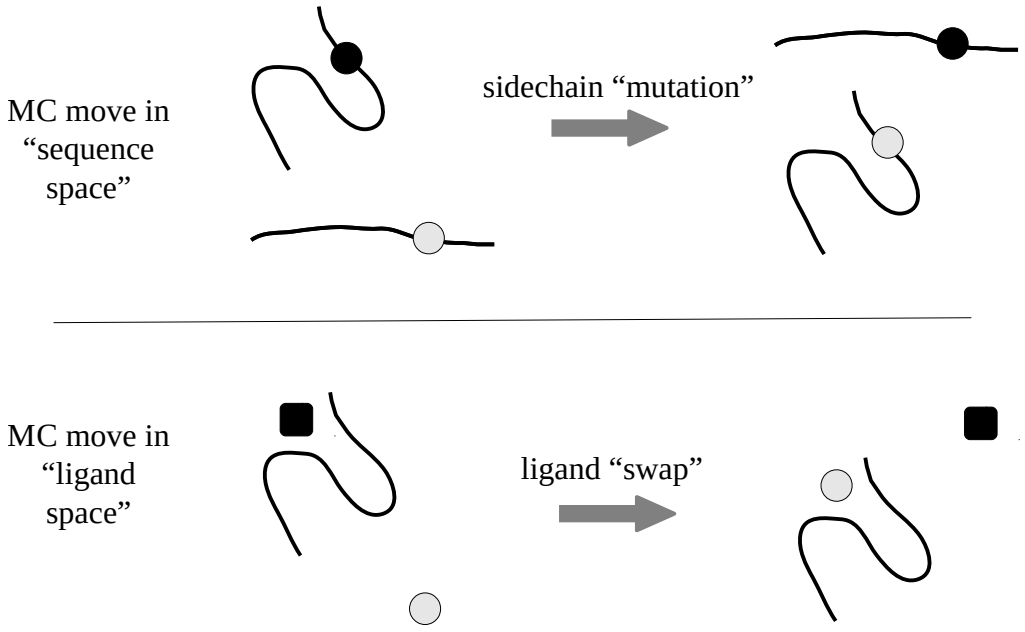

Figure 1: **Protein mutation** (above) in the folded and unfolded states. **Ligand mutation** (below) in the bound and unbound states.

1.3 The statistical ensemble

The distribution of states is largely determined by the probabilities for selecting and accepting moves. Let $\alpha(o \to n)$ be the probability to select a move between two states o and n; let $acc(o \to n)$ be the acceptance probability. The overall probability of the move is

$$\pi(o \to n) = \alpha(o \to n)\, acc(o \to n) \tag{3}$$

The classic Metropolis scheme [1–3] chooses the acceptance probability as

$$acc(o \to n) = \exp(-\beta \Delta E) \quad \text{if } \Delta E > 0; 1 \text{ otherwise} \tag{4}$$

where $\Delta E = E(n) - E(o)$. This leads to

$$\frac{acc(o \to n)}{acc(n \to o)} = \exp(-\beta \Delta E) \tag{5}$$

If we assume the simulation obeys **detailed balance**, we have the condition

$$N(o)\pi(o \to n) = N(n)\pi(n \to o), \tag{6}$$

where $N(o)$ is the equilibrium population of state o (respectively, n). We then have

$$\frac{N(n)}{N(o)} = \frac{\pi(o \to n)}{\pi(n \to o)} = \frac{\alpha(o \to n) acc(o \to n)}{\alpha(n \to o) acc(n \to o)} = \exp(-\beta \Delta E) \frac{\alpha(o \to n)}{\alpha(n \to o)} \tag{7}$$

If the probabilities to select moves are symmetric, $\alpha(o \to n) = \alpha(n \to o)$, we are left with the Boltzmann distribution where the states are populated (at equilibrium) according to their Boltzmann factors $\exp(-\beta E) = \exp(-\beta(E_f - E_{uf}))$.

In the general case where the probabilities to select moves are not symmetric, we can modify the acceptance probabilities as follows:

$$acc(o \to n) = \exp(-\beta \Delta E) \frac{\alpha(o \to n)}{\alpha(n \to o)} \quad \text{if } \Delta E > 0; 1 \text{ otherwise} \tag{8}$$

and obtain

$$\frac{N(n)}{N(o)} = \exp(-\beta \Delta E), \tag{9}$$

so that Boltzmann statistics are again respected. Recall that detailed balance is *assumed.*

1.4 Type and rotamer changes

For a mutation move o → n (at a particular position in the polypeptide chain), we perform the forward mutation in the folded structure and the reverse mutation in the unfolded structure (Fig. 1). We define the probability $\alpha(o \to n)$ as follows:
(a) select a new type t' with equal probabilities $\alpha_t(o \to n) = \frac{1}{N}$ for all N possible types;
(b) choose a rotamer r' for the folded structure with equal probabilities $\alpha_{\rm fr}(o \to n) = \frac{1}{M_f(t')}$ for all $M_f(t')$ possible folded-state rotamers.
(c) select a rotamer r for the unfolded structure with equal probabilities $\alpha_{\rm ufr}(o \to n) = \frac{1}{M_{uf}(t)}$ for all $M_{uf}(t)$ possible unfolded-state rotamers.
The overall probability to choose the particular o → n move is therefore

$$\alpha(o \to n) = \alpha_t(o \to n)\alpha_{\rm fr}(o \to n)\alpha_{\rm ufr}(o \to n) \tag{10}$$

Notice that in applications, $\alpha_t(o \to n)$ is usually symmetrical (equal probabilities for forward and backward mutations); $\alpha_{\rm fr}(o \to n)$ and $\alpha_{\rm ufr}(o \to n)$ are usually not symmetrical, since the old and new sidechain types can have different numbers of possible rotamers, respectively, in the folded and unfolded states. The product $\alpha_{\rm fr}(o \to n)\alpha_{\rm ufr}(o \to n)$ is symmetrical if the folded/unfolded rotamer numbers are equal and dissymmetrical otherwise:

$$\alpha_r(o \to n) = \alpha_{\rm fr}(o \to n)\alpha_{\rm ufr}(o \to n) = \frac{1}{M_{uf}(t)M_f(t')} \tag{11}$$

One possible assumption for the unfolded sidechains is that there is a single, dominant rotamer, $M_{uf}(t) = 1$, in which case $\alpha_r(o \to n) \neq \alpha_r(o \to n)$ in general. A better assumption is that there are several rotamers of equal energy, $M_{uf}(t) > 1$. The simplest assumption is that the number of rotamers is the same in the folded and unfolded states, which leads to symmetrical move probabilities, $\alpha(o \to n) = \alpha(n \to o)$. Various other assumptions are possible; for example, the number of folded rotamers might depend on the amino acid *position* (and not just its type), and thus differ from $M_{uf}(t)$ in general.

If the folded and unfolded states have different rotamer numbers, we should use the more general form (8) of the move acceptance probabilities, which can be rewritten:

$$acc(o \to n) = \exp(-\beta\Delta E)\frac{M_{uf}(t)}{M_f(t')} \quad \text{if } \Delta E > 0; 1 \text{ otherwise} \tag{12}$$

Another way to write the acceptance probability is to introduce a shifted unfolded energy F_{uf} as follows:

$$F_{uf}(S) = \sum_{i=1}^{n}\left(E_{uf}(t_i) + kT\log\frac{M_{uf}(t_i)}{M_f(t_i)}\right) \tag{13}$$

The acceptance probabilities are then

$$acc(o \to n) = \exp(-\beta(\Delta E_f - \Delta F_{uf})) \;\text{ if }\; \Delta E > 0; 1 \;\text{ otherwise} \tag{14}$$

This has the form of the classic Metropolis scheme; the rotamer numbers have been "absorbed" into the exponential, with the shifted unfolded energy F_{uf} replacing the earlier E_{uf}. There is a subtle point, though: while F appears in the exponent, it is E that determines whether the move is uphill.

From now on, unless otherwise mentioned, we will limit ourselves to the simplest unfolded state model: each sidechain has the same number of rotamers as in the folded state, and all its unfolded rotamers have the same energy. With this assumption, we can use the plain Metropolis acceptance probabilities. Importantly, even though we make an assumption about the unfolded rotamers, there is no need to actually model the unfolded structure in the simulation, since the unfolded energy depends only on the amino acid types, not on a particular structure. The contribution of each sidechain to the unfolded energy E_{uf} includes an enthalpy term (eg, a tripeptide energy) and a conformational entropy term ($-kT \log M_{uf}$, due to the existence of multiple rotamers). In practice, the values $E_{uf}(t_i)$ will be determined empirically, by fitting simulation data to experimental properties such as natural amino acid abundancies.

From the Boltzmann distribution, Eq. (9), it is easy to compute the probability of having a particular type t at position i in the polypeptide, by summing over the rotamers r available to that type:

$$\frac{N(t')}{N(t)} = \frac{\sum_{r'} \exp(-\beta E(t', r'))}{\sum_r \exp(-\beta E(t, r))} = \frac{\sum_{r'} \exp(-\beta E_f(t', r'))}{\sum_r \exp(-\beta E_f(t, r))} \frac{\exp(\beta E_{uf}(t'))}{\exp(\beta E_{uf}(t))} \tag{15}$$

We denote $F_f(t)$ the energy of the system averaged over the rotamers of t (at position i), which should be viewed in fact as a free energy:

$$\exp(-\beta E_f(t)) = \sum_r \exp(-\beta E_f(t, r)), \tag{16}$$

so that finally

$$N(t) = A \exp(-\beta(E_f(t) - E_{uf}(t))) \tag{17}$$

where A is a constant (common to all states).

1.5 Ligand mutations

Consider a protein:ligand complex; suppose the ligand can be bound or unbound, and can have several types l, l', ..., each with its own set of $M_b(l)$, $M_{ub}(l)$ rotamers. A ligand mutation is similar to the protein mutations above: we simultaneously change the bound type from l to l', and the unbound type from l' to l, effectively swapping ligands in the binding site (Fig. 1). For each one, bound and unbound, we randomly select a rotamer from among the $M_b(l')$, $M_{ub}(l)$ possibilites. The probability of selecting a particular ligand mutation move is

$$\alpha(o \to n) = \alpha_t(o \to n)\alpha_r(o \to n) \tag{18}$$

$$\alpha_r(o \to n) = \alpha_{\rm br}(o \to n)\alpha_{\rm ubr}(o \to n) = \frac{1}{M_b(l')M_{ub}(l)} \tag{19}$$

In general, the number of bound and unbound rotamers are different: $M_b(l) \neq M_{ub}(l)$, so that $\alpha(o \to n) \neq \alpha(n \to o)$. Thus, one should either use the more general acceptance probabilities (12), or replace the unbound ligand energy E_{ub} by a "shifted" value F_{ub}, analogous to the unfolded protein case, Eq. (13).

The energy change associated with a ligand mutation can be written:

$$\Delta E = \Delta E_b - \Delta E_{ub} = (E_b(l', r') - E_b(l, r)) - (E_{ub}(l') - E_{ub}(l)) \tag{20}$$

Here, we have assumed for simplicity that the unbound energy does not depend on the rotamer (equistable unbound rotamers). More generally, the energy $E_{ub}(l)$ of an unbound ligand includes several contributions:

$$E_{ub}(l) = E_{\rm solv}(l) - TS_{\rm conf}(l) - TS_{\rm ext}(l) + kT\log\rho(l) \tag{21}$$

The first term is an internal and solvation energy $E_{\rm solv}$, normally obtained from a physical energy function (eg, the "XPLOR" energy). The second is a conformational entropy term, which is straightforward to obtain. For example, if we assume the rotamers of the unbound ligand all have the same energy, this term is just $-kT\log M_{ub}(l)$. The third term is associated with the external, rotational and translational entropy of the ligand in the standard state. It can be obtained from a Rigid Rotor Harmonic Oscillator model [4, 5], and will cancel out when the ligands l, l' have a similar size and shape, as in many applications. The last term represents the concentration-dependent part of the translation entropy; it is normally the same for all ligand types and does not contribute to the mutation energy ΔE.

1.6 Acid/base mutations

A special kind of "mutation" occurs when we modify the protonation state of a sidechain. The corresponding Monte Carlo move is slightly different from the protein or ligand mutations discussed so far. There is no need to perform the reverse mutation in the unfolded protein or an unbound ligand. In particular, there is no need to randomly choose a rotamer for un unfolded or unbound sidechain or ligand, and this alters slightly the move probabilities $\alpha(o \to n)$. However, we do use an unbound ligand as an auxiliary in the energy calculation (Fig. 2), and its rotamer numbers play a role. Specifically, we consider a small molecule (or "model compound") that ressembles the sidechain being protonated, and that is at a distant location in solution. We add and subtract two values for its protonation energy: one estimated with our energy function and the other taken from experiment—a classic trick to remove systematic errors from the model [6–8]. Computed with our energy function, the unbound model compound's energy E_{ub} has the same form as for the ligand above, Eq. (21), including a contribution from its rotamers. Derived from experiment, the (free) energy change upon protonating the model compound is $\Delta E_{ub}^{\exp}$ = -2.303 kT (pK_a - pH), where pK_a is the experimental acid/base constant, kT is the thermal energy, and a pH-dependence appears explicitly.

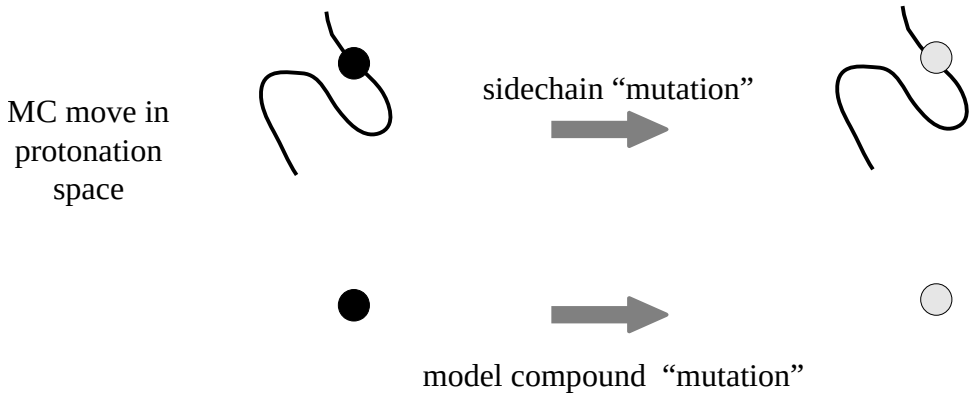

Figure 2: **Protonation change** in the protein (above) and a model compound (below).

The other element that determines move acceptances is the probability $\alpha(o \to n)$ to select a trial move. Here, we first choose the new protonation state, t', then the new

rotamer r':

$$\alpha(o \to n) = \alpha_t(t \to t')\,\alpha_r(r') = \frac{1}{N_i - 1}\frac{1}{M(t')}, \tag{22}$$

where N_i is the number of possible types (protonation states) at position i and $M(t')$ is the number of rotamers for the mutant type. The move probabilities are dissymetrical in general:

$$\frac{\alpha(o \to n)}{\alpha(n \to o)} = \frac{M(t)}{M(t')} \tag{23}$$

The acceptance probabilities are chosen as above:

$$\begin{aligned} acc(o \to n) &= \exp(-\beta\Delta E)\frac{M(t)}{M(t')} \quad \text{if } \Delta E > 0 & (24)\\ &= \exp(-\beta\Delta F) \quad \text{if } \Delta E > 0 & (25)\end{aligned}$$

where

$$F(t) = E(t) + kT \log M(t) \tag{26}$$

is a shifted energy that absorbs the rotamer numbers, as above. Notice that in (25), while ΔF appears in the exponent, ΔE is used to determine whether the move is uphill.

As an example, consider the mutation of an Asp sidechain from its ionized form "ASP" to its protonated form "ASH". The protonated form ASH has $n_p = 4$ rotamers for every ASP rotamer (Fig. 3): $M_b(\text{ASH}) = M_{ub}(\text{ASH}) = n_p M_b(\text{ASP}) = n_p M_{ub}(\text{ASP})$. For each form, we assume all the unbound rotamers have the same energy $E_{\text{solv}}(\text{l})$. The two Asp forms have essentially the same size and shape, so that the entropy contributions $S_{\text{ext}}(\text{l})$ are the same for both. The energy change for an ASP $\to$ ASH mutation can be written:

$$\begin{aligned} \Delta E &= \Delta E_b - \Delta E_{ub} + \Delta E_{ub}^{\text{exp}} & (27)\\ &= \Delta E_b - \left(\Delta E_{\text{solv}} - kT \log \frac{M(\text{ASH})}{M(\text{ASP})}\right) + 2.303kT(pK_a - pH) & (28)\end{aligned}$$

where $M(\text{ASP})$, $M(\text{ASH})$ are the number of ASP, ASH rotamers. ΔE_b is computed with our energy function and depends on the particular set of rotamers and protonation states in the rest of the protein. ΔE_{solv} is a constant (rotamer-independent), computed ahead of time with our energy function. The change in the shifted energy F is

$$\Delta F = \Delta E_b - \Delta E_{\text{solv}} + 2.303kT(pK_a - pH) \tag{29}$$

The rotamer numbers no longer appear explicitly, and the plain Metropolis test can be used for the acceptance probabilities, with one subtle point: the unshifted, not the shifted energy determines whether a move is uphill. The two differ by $kT \log n_p = 0.8$ kcal/mol in this example.

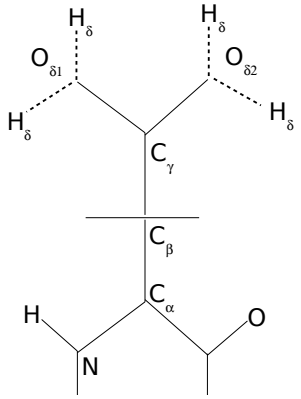

Figure 3: **Aspartate rotamers.** A single ASP rotamer corresponds here to four ASH rotamers, with the four different H_δ positions shown.

1.7 Hybrid mutation moves

We consider next the possibility of using "hybrid" moves for amino acid mutations. One idea is to use moves that have two stages:

(a) select a new type t' and rotamer r' at some position i;

(b) perform N steps of Monte Carlo in rotamer space (no mutations)

At the end of stage (b), apply a modified Metropolis test, accepting or rejecting the entire two-stage move; rejection returns us to where we were before (a). This two-stage scheme could be advantageous, facilitating mutations by allowing the nearby structure to relax before we make a decison to accept or reject. The same idea could be applied to other kinds of moves, such as changing the polypeptide backbone geometry. The question is whether there is a simple acceptance rule that will lead to Boltzmann populations.

In the limit where stage (b) is very long, we could in principle estimate the free energy for the mutated system,

$$\exp(-\beta E_f^\infty(t)) = \frac{1}{T}\sum_n \exp(-\beta E_f(t, \{r_n\})), \tag{30}$$

where the sum on the right is over the Monte Carlo steps n performed during stage (b), T is the run length, and $\{r_n\}$ represents the complete set of rotamers sampled at step n (one per amino acid in the polypeptide). Since stage (b) is very long, the energy is denoted

with an exponent ∞. Suppose we accept or reject the mutation according to

$$acc(t \to t') = \exp(-\beta(\Delta E_f^\infty - \Delta E_{uf})) \text{ if } \Delta E_f^\infty > \Delta E_{uf}); 1 \text{ otherwise} \tag{31}$$

It is easy to see that in this case, the amino acid types will be distributed with Boltzmann statistics.

In the case where stage (b) has a finite length T, the quantity $\exp(-\beta E_f^T(t))$ is defined in the same way but the rotamer sum or average is done over a much shorter Monte Carlo run. This short MC run is stochastic, so that $\exp(-\beta E_f^T(t))$ can be viewed as a random quantity that would converge to $\exp(-\beta E_f^\infty(t))$ if T became very large. For this reason, we switch to the notation $\mathcal{E}_f^T(t)$, where $\mathcal{E}_f^T(t)$ is a random variable. Suppose we accept or reject the mutation according to

$$acc(t \to t') = \exp(-\beta(\Delta \mathcal{E}_f^T - \Delta E_{uf})) \text{ if } \Delta \mathcal{E}_f > \Delta E_{uf}; 1 \text{ otherwise} \tag{32}$$

The energy $\mathcal{E}_f^T$ in the Metropolis test is now a random quantity. Given what we know about $\mathcal{E}_f$, can we show that this leads to a Boltzmann distribution of amino acid types?

References

[1] Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. Equation of state calculations by fast computing machines. *J. Chem. Phys. 21* (1953), 1087–1092.

[2] Frenkel, D., and Smit, B. *Understanding molecular simulation, Chapter 3.* Academic Press, New York, 1996.

[3] Grimmet, G. R., and Stirzaker, D. R. *Probability and random processes.* Oxford Universiity Press, 2001.

[4] Hill, T. *Introduction to Statistical Thermodynamics.* Addison-Wesley, Reading, Massachussetts, 1962.

[5] Simonson, T. The physical basis of ligand binding. In *In Silico Drug Discovery and Design: Theory, Methods, Challenges, and Applications*, C. Casavotto, Ed. CRC Press, 2015, ch. 1.

[6] Aleksandrov, A., Polydorides, S., Archontis, G., and Simonson, T. Predicting the acid/base behavior of proteins: A constant-pH Monte Carlo approach with Generalized Born solvent. *J. Phys. Chem. B 114* (2010), 10634–10648.

[7] Polydorides, S., and Simonson, T. Monte Carlo simulations of proteins at constant pH with generalized Born solvent, flexible sidechains, and an effective dielectric boundary. *J. Comput. Chem. 34* (2013), 2742–2756.

[8] Sham, Y., Chu, Z., and Warshel, A. Consistent calculations of pK_a's of ionizable residues in proteins: semi-microscopic and microscopic approaches. *J. Phys. Chem. B 101* (1997), 4458–4472.