Biophysical origin of long-range functional constraints in enzyme evolution

Julian Echave

1 Main goal

Explain the long-range dependence of site-specific substitution rates with distance from the active site in enzymes.

1.1 Approach

I will use two approaches:

1. **Active Site Conformation Constraint (ASCC)**: Fitness depends on the probability that the active site reaches a given active conformation.

2. **Enzyme-Substrate Interaction Constraint (ESIC)**: Fitness depends on the (free) energy of interaction between the active site and a model ligand.

2 Elastic Network Models

An Elastic Network Model (ENM) represents a given protein as a 3D network of nodes connected by harmonic springs. There may be one or more nodes per amino acid. The potential energy function is

$$\boxed{V = \sum_{i<j} V_{ij}^{\mathtt{min}} + \frac{1}{2} k_{ij} (d_{ij} - l_{ij})^2,} \tag{1}$$

where $V_{ij}^{\mathtt{min}}$, k_{ij}, and l_{ij} are, respectively, the minimum energy, the force constant, and the equilibrium length of the spring $i \leftrightarrow j$. The parameters of the model are set in such a way that the minimum-energy conformation is the protein's known native conformation.

2.1 Quadratic expansion

A specific protein conformation is represented by the column vector of Cartesian coordinates of the N nodes: $\mathbf{r} = (x_1, y_1, z_1, ..., x_N, y_N, z_N)^T$. The position vector of node i is $\mathbf{r}_i = (x_i, y_i, z_i)^T$. The distance between nodes i and j is $d_{ij} = ||\mathbf{r}_j - \mathbf{r}_i||$.

Let $\mathbf{r}^0$ be an arbitrary conformation. The second-order expansion of the ENM potential (1) around $\mathbf{r}^0$ is

$$\boxed{V \sim V(\mathbf{r}^0) - \mathbf{F}(\mathbf{r}^0)(\mathbf{r} - \mathbf{r}^0) + \frac{1}{2}(\mathbf{r} - \mathbf{r}^0)^T \mathbf{K}(\mathbf{r}^0)(\mathbf{r} - \mathbf{r}^0),} \tag{2}$$

where $\mathbf{F}$ is the force (row) vector and $\mathbf{K}$ is the Hessian:

$$\mathbf{F}(\mathbf{r}) = -\frac{\partial V}{\partial \mathbf{r}}, \tag{3}$$

$$\mathbf{K}(\mathbf{r}) = \frac{\partial^2 V}{\partial^2 \mathbf{r}}. \tag{4}$$

The derivatives of d_{ij} are given by:

$$\frac{\partial d_{ij}}{\partial \mathbf{r}_j} = \mathbf{e}_{ij}^T \tag{5}$$

$$\frac{\partial^2 d_{ij}}{\partial \mathbf{r}_i \partial \mathbf{r}_j} = \frac{\mathbf{e}_{ij}\mathbf{e}_{ij}^T - \mathbf{I}}{d_{ij}} \tag{6}$$

$$\tag{7}$$

where

$$\mathbf{e}_{ij} = \frac{\mathbf{r}_j - \mathbf{r}_i}{d_{ij}} \tag{8}$$

is a unit vector in the direction of $\mathbf{r}_{ij}$.

From (3) and (5), we find:

$$\boxed{\mathbf{F}_i(\mathbf{r}) = \sum_{j \neq i} k_{ij}(d_{ij} - l_{ij})\mathbf{e}_{ij}^T.} \tag{9}$$

Note that since $\mathbf{e}_{ij} = -\mathbf{e}_{ji}$,

$$\sum_i \mathbf{F}_i = 0; \tag{10}$$

There is no external force acting on the network.

From (4), using (5) and (6) we find:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \dots & \mathbf{K}_{1N} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \dots & \mathbf{K}_{2N} \\ \vdots & \vdots & \dots & \vdots \\ \mathbf{K}_{N1} & \mathbf{K}_{N2} & \dots & \mathbf{K}_{NN} \end{bmatrix}, \tag{11}$$

where the 3×3 matrices $\mathbf{K}_{ij}$ are:

$$\mathbf{K}_{i \neq j} = -k_{ij}\left[\mathbf{e}_{ij}\mathbf{e}_{ij}^T + \left(1 - \frac{l_{ij}}{d_{ij}}\right)\left(\mathbf{e}_{ij}\mathbf{e}_{ij}^T - \mathbf{I}\right)\right], \tag{12}$$

$$\mathbf{K}_{ii} = -\sum_{j \neq i} \mathbf{K}_{ij}, \tag{13}$$

2.2 Equilibrium Conformation

The ENM energy (1) is minimum at the conformation $\mathbf{r}^e$ that satisfies:

$$\mathbf{F}_i(\mathbf{r}^e) = \mathbf{0} \text{ for } i = 1 \dots N. \tag{14}$$

According to this equation, at $\mathbf{r}^e$ the net force acting on each node is zero.

Replacing (9) in (14) we find:

$$\mathbf{F}_i(\mathbf{r}^e) = \sum_{j \neq i} k_{ij}(d_{ij}^e - l_{ij})\mathbf{e}_{ij}^{e\,T} = \mathbf{0}. \tag{15}$$

This is a system of $3 \times N$ equations.

Given the parameters $\{l_{ij}\}$ solving equations (15) we can obtain the equilibrium conformation $\mathbf{r}^e$. These equations are non-linear because $d^e_{ij} = ||\mathbf{r}^e_j - \mathbf{r}^e_i||$. To find the minimum, we can use a steepest descent approach. For example, let $\mathbf{r}^0_n$ be a given conformation in iteration step n. The minimum of the quadratic approximation (2) is at:

$$\mathbf{r}^0_{n+1} = \mathbf{r}^0_n + \mathbf{K}^{-1}(\mathbf{r}^0_n)\mathbf{F}^T(\mathbf{r}^0_n) \tag{16}$$

Under certain conditions, we expect this algorithm to converge to $\mathbf{r}^e$, for which $\mathbf{F}_i(\mathbf{r}^e) = 0$.

2.3 Quadratic expansion around the minimum

From (2) and (15) it follows:

$$\boxed{V \sim V(\mathbf{r}^e) + \frac{1}{2}(\mathbf{r} - \mathbf{r}^e)^T \mathbf{K}(\mathbf{r}^e)(\mathbf{r} - \mathbf{r}^e),} \tag{17}$$

where (from (1)) the minimum energy is:

$$\boxed{V(\mathbf{r}^e) = \sum_{i<j} V^{\mathtt{min}}_{ij} + \frac{1}{2} k_{ij}(d^e_{ij} - l_{ij})^2.} \tag{18}$$

The second term of eq. (18) represents frustration. In general one would not expect all springs to be relaxed at the equilibrium conformation. Since the frustration term is zero or positive, the minimum energy is equal or larger than the sum over pairs of minimum energies.

2.4 ENM parameters l_{ij}

For a protein of known equilibrium conformation, given $\mathbf{r}^e$ any set of parameters $\{l_{ij}\}$ that satisfies (15) has the same equilibrium conformation and, therefore, is a valid ENM model.

2.4.1 Non-frustrated ENMs

For the special case $l_{ij} = d^e_{ij}$ equation (15) is trivially satisfied: all terms are zero by construction; at $\mathbf{r}^e$ all springs are relaxed and the network is non-frustrated. As far as I know, all ENMs developed so far are non-frustrated.

2.4.2 Frustrated ENMs

In general, it is reasonable assume some degree of frustration in proteins. Moreover, according to the mutational model (to be discussed below), as we introduce mutations frustrations are introduced. Even if we start an evolutionary trajectory with a non-frustrated network, as we introduce mutations the network will become frustrated. Therefore, a non-frustrated model is not general enough for my purposes. More on frustrated ENMs in A

3 Statistical mechanics

3.1 Boltzmann distribution

The Boltzmann conformational probability density function is given by:

$$\rho(\mathbf{r}) = \frac{e^{-\beta V(\mathbf{r})}}{Z}, \tag{19}$$

where the Z is the conformational partition function:

$$Z = \int e^{-\beta V(\mathbf{r})} d\mathbf{r} \tag{20}$$

Using the second-order expansion of the potential energy around the minimum, Equation (17), we find a normal distribution:

$$\rho(\mathbf{r}) = \frac{e^{-1/2(\mathbf{r}-\mathbf{r}^e)^T \mathbf{\Sigma}^{-1}(\mathbf{r}-\mathbf{r}^e)}}{\sqrt{|2\pi\mathbf{\Sigma}|}}, \tag{21}$$

where $|\mathbf{M}|$ is the determinant of matrix $\mathbf{M}$. The variance-covariance matrix $\mathbf{\Sigma}$ is related to the pseudo-inverse of the Hessian matrix:

$$\mathbf{\Sigma} = \beta^{-1}\mathbf{K}^{+}. \tag{22}$$

In general, the Hessian at equilibrium has 6 eigenvectors with eigenvalue 0. They correspond to 3 translations and 3 rotations. Thus, $\mathbf{K}$ is not invertible. However, the pseudo-inverse can be calculated using a spectral decomposition omitting the 6 zero eigenvectors.

3.2 Partition function

Comparing (21) and (19) with $V(\mathbf{r})$ given by (17) we find:

$$Z = e^{-\beta V(\mathbf{r}^e)} |2\pi\mathbf{\Sigma}|^{\frac{1}{2}} \tag{23}$$

In the normal-mode representation, $\mathbf{\Sigma}$ is diagonal with elements $\sigma_n^2 \delta_{mn} = (\beta\lambda_n)^{-1}\delta_{mn}$, where λ_n is the nth eigenvalue of $\mathbf{K}$. Then from the previous equation we find:

$$|2\pi\mathbf{\Sigma}|^{1/2} = \prod_n \sqrt{2\pi\sigma_n^2} = \prod_n \sqrt{\frac{2\pi}{\beta\lambda_n}} \tag{24}$$

Finally, replacing (24) into (23) we find:

$$Z = e^{-\beta V(\mathbf{r}^e)} \prod_n \sqrt{\frac{\beta\lambda_n}{2\pi}} \tag{25}$$

3.3 Thermodynamic functions

Is the stuff in this section valid without including the kinetic energy terms?

3.3.1 Internal energy

The internal energy U can be calculated from:

$$U = -\frac{\partial \ln Z}{\partial \beta} \tag{26}$$

Replacing (24) into (26) we find:

$$U = V(\mathbf{r}^e) + (3N - 6)\frac{1}{2\beta} \tag{27}$$

3.3.2 Helmholtz free energy

The free energy A can be calculated from:

$$A = -\beta^{-1} \ln Z \tag{28}$$

Replacing (24) into (28) we get:

$$A = V(\mathbf{r}^e) + \sum_n \frac{1}{2\beta} \ln \frac{\beta\lambda_n}{2\pi} \tag{29}$$

3.3.3 Entropy

From $A = U - TS$ it follows that $TS = U - A$. Then, using (27) and (29) we find:

$$TS = \frac{1}{2\beta} \sum_n \left(\ln \frac{2\pi}{\beta\lambda_n} + 1 \right). \tag{30}$$

3.4 Include kinetic energy term

Include kinetic energy terms.

4 Mutation models

We model a at a site mutation by perturbing the parameters of the ENM springs that connect that site to the rest of the network.

4.1 Linearly-Forced Elastic Network Model

According to the LFENM, a mutation is modeled by perturbing only the equilibrium lengths of springs attached to the mutated site: $l_{ij} \rightarrow l_{ij} + \delta l_{ij}$. The perturbations δl_{ij} are picked independently for each of the contacts of the mutated site, from the same distribution. The distribution is assumed to satisfy $< \delta l_{ij} >= 0$ and $Var(\delta l_{ij}) = \sigma^2$. In principle, the mutation will change also the other two parameters of the $i \leftrightarrow j$ contact, $V_{ij}^{\mathtt{min}}$ and k_{ij}. However, the LFENM assumes $\delta V_{ij}^{\mathtt{min}} = 0$ and δk_{ij} is calculated self-consistently from structure of the mutant.

The mutant's potential is:

$$V_{mut} = \sum_{i<j} \left[V_{ij}^{\mathtt{min}} + \frac{1}{2} k_{ij} [d_{ij} - (l_{ij} + \delta l_{ij})]^2 \right]. \tag{31}$$

The LFENM is obtained by expanding Eq. 31 up to second order. The potential is expressed in terms of "forces" directed along the contacts of the mutated site. The norm of such forces is:

$$f_{ij} = k_{ij} \delta l_{ij} \tag{32}$$

- *Quadratic form of LFENM potential ...*
- *Mutant's equilibrium structure ...*
- *Self-consistent calculation of* **K** *...*

4.2 Generalized perturbed ENM

Model in which a mutation perturbs all parameters of the springs of the mutated site.

5 Fixation probability

Basics of fixation probability for the Moran and/or Wright-Fisher processes.

6 Selection models

6.1 Stress model

When the mutant adopts the wild-type structure ${\mathbf{r^0}}_{wt}$, $d_{ij} = d^0_{ij}$, and the mutant's energy is:

$$V_{mut}({\mathbf{r^0}}_{wt}) = \frac{1}{2}\sum_{i<j} k_{ij}\Delta^2_{ij} \tag{33}$$

The stress model poses that the probability of accepting such a mutant is:

$$P_{accept} = e^{-\beta\frac{1}{2}\sum_{i<j} k_{ij}\Delta^2_{ij}} \tag{34}$$

6.2 New Stress model

New version of stress model where a fitness function is posed and Moran/Wright-Fisher fixation probabilities are used.

6.3 Active Site Conformation model

Selection model by restricting the active site conformation using the Cartesian coordinates of active residues.

6.4 Active Site Distance Constraints

Constrain the distance between catalytic residues, rather than the actual Cartesian coordinates.

The problem with constraining the Cartesian coordinates of each of the catalytic residues is that it might be too restrictive. It might be more reasonable to assume that activity depends on the shape of the active site, which might be better described by the distances between catalytic residues.

Let's assume that $\mathbf{d} = (d_{ij})$ i.e. a vector of distances between all pairs of catalytic residues. Then, we assume that fitness is proportional to the probability of folded protein in the right conformation, as defined by $\mathbf{d}^*$:

$$f = P(F) \times P(\mathbf{d} = \mathbf{d}^*|F) \tag{35}$$

We could calculate this analytically or numerically. In the latter case we could, for instance, generate several active-site Cartesian conformations using

the effective distribution of active-site residues and count how many lye in the cube $\mathbf{d}$ and $\mathbf{d} + \delta\mathbf{d}$.

Note that it is not clear how to define the cube. For instance, a cube in one coordinate system will map into a solid with different lengths according to the deformation energy in each direction if we normalize coordinates... I need to think about this.

6.5 Substrate-Enzyme interaction model

Put one or mode nodes to model the substrate. Base the fitness function on the interaction energy of this node with the rest of the protein.

7 Evolutionary Simulation

7.1 One evolutionary step

How would we simulate a single evolutionary time step? Let us define a time-step such that in a time-step there is a single substitution event for the whole protein (a substitution is an accepted mutation). To simulate such an event we do the following:

1. Pick one random site l
2. Introduce a "trial" mutation by obtaining a set of f_{lj} for each of the contacts of site l
3. Calculate the probability of accepting this trial mutation: $P_{accept} = e^{-\beta \frac{1}{2} \sum_{j \sim i} f_{lj}^2}$
4. Calculate the logical variable Accept = $P_{accept} \geq$ runif(1,0,1); Accept will be TRUE with probability P_{accept}, this is a way of implementing accepting mutations with this probability.
5. If Accept is TRUE, accept the mutation (i.e. the new wild-type is the mutant) and the evolutionary step is finished (i.e. we found an accepted mutation: one substitution). Else, reject the trial mutation and try again (i.e. go back to Step 1: pick a random site, etc.).

For future reference, let's assume that we implement this single-step using a function oneSubstitution(sites) that returns the mutated site l, and the force that simulates the mutation f_{lj} (or alternatively the force-vector $\mathbf{f}_l$).

7.2 An evolutionary path of N substitution at N different sites

We want to simulate a lineage (an evolutionary path) such that it starts at a known ancestor (our reference protein) and it ends when N sites have accepted mutations. ($N \leq L$, where L is the sequence length). Briefly, we need to repeat the single evolutionary step of the previous paragraph N times, making sure that the set of sites where we try mutations does not include sites that previously accepted mutations (this corresponds to sampling without replacement). The process would be, using pseudocode:

```
NonMutatedSites = seq(L)
MutatedSites = c()
for(subs in seq(N)) {
        step = oneSubstitution(NonMutatedSites)
    l = step$l
    f_l = step$f_l
    Remove l from NonMutatedSites
    Add l to MutatedSites
    Add f_l to list of forces (or matrix of forces)
}
```

7.3 Star tree

Our "experimental" data are sets of proteins one of which we consider as the "reference" protein. In principle, we should infer the phylogenetic tree and try to simulate structures with our model following a tree with the same topology. However, we assume that the results are not too sensitive with respect to tree topology so that we can approximate the tree by a "star tree". The common ancestor of all lineages of our star tree is our reference protein. Then, each lineage corresponds to a pair alignment of each protein with the reference. Thus, different lineages have different "branch lengths", were we assume the branch lengh is the number of sites in which the sequence of the (common) ancestor and the tip of the lineage differ.

A Frustrated ENMs

A.1 Determination of l_{ij} to model a frustrated ENM

How can we determine spring length parameters l_{ij} that account for frustration?

In general, we need l_{ij} that satisfy (15) and some other conditions (because there are more l_{ij} than equations). Some possible extra constraints and/or ways of obtaining l_{ij} are:

- Let $l_{ij} = d^e_{ij}$ for sequential contacts and for secondary-structure contacts and consider $l_{ij} \neq d^e_{ij}$ for tertiary contacts.

- Make l_{ij} depend only on the identity of the amino acids at i and j if i and j are in contact. Thus, we have 220 independent l_{ij} to be determined by solving the $3N-6$ independent equations (15). Since we have more equations than parameters, the system is over-determined. Solve using EM or ML algorithms. Using this approach, we could try to fit several pdb structures at the same time to find the l_{ij} matrix.

- Add constraints, such as a given average energy, and maximize the entropy w.r.t. l_{ij}. This will give a distribution of l_{ij}.

- Make all tertiary l_{ij} identical (i.e. non-sequence dependent), but pick which are different from zero (i.e. build the network so that with the constraint of identical l_{ij} has the required equilibrium conformation).

- Start with a given protein structure and $l_{ij} = d^e_{ij}$. Then, perturb all parameters by adding random δl_{ij} to each l_{ij}. In general, this will result in a set of parameters that do not satisfy (15). Starting with such parameters, minimize $\sum_i ||\mathbf{F}_i||$ (e.g. running a Monte Carlo simulation by adding random δl_{ij} to the initial l_{ij} values).

- Run evolutionary simulations accepting mutations according the Stress Model (i.e. $P(\mathbf{r}^e)$). Mutant structures should remain in a neighborhood of the original wild-type structure and yet their l_{ij} parameters will diverge. Selection pressure will determine how close mutant structures are to the wild-type structure.

- Start with the relaxed ENM for the wild-type protein. Then, run evolutionary simulations with selection with a certain energy cutoff ($> V_{cut}$). Such simulations are obtained by adding δl_{ij} to to the springs of mutated sites and recalculating equilibrium structures. The mutants that result from such simulations are frustrated elastic networks.