Fast computational mutation-response scanning of proteins

Julian Echave

Instituto de Ciencias Físicas, Escuela de Ciencia y Tecnología, Universidad Nacional de San Martín, San Martín, Buenos Aires, Argentina

ABSTRACT

Studying the effect of perturbations on protein structure is a basic approach in protein research. Important problems, such as predicting pathological mutations and understanding patterns of structural evolution, have been addressed by computational simulations that model mutations using forces and predict the resulting deformations. In single mutation-response scanning simulations, a sensitivity matrix is obtained by averaging deformations over point mutations. In double mutation-response scanning simulations, a compensation matrix is obtained by minimizing deformations over pairs of mutations. These very useful simulation-based methods may be too slow to deal with large proteins, protein complexes, or large protein databases. To address this issue, I derived analytical closed formulas to calculate the sensitivity and compensation matrices directly, without simulations. Here, I present these derivations and show that the resulting analytical methods are much faster than their simulation counterparts.

INTRODUCTION

Protein function is fundamentally related to protein structure. For this reason, insight into protein function can be gained by studying the structural deformations caused by perturbations. This is at the basis of general experimental and theoretical approaches to study proteins. An experimental example is Deep Mutational Scanning, which allows studying the effects of large numbers of mutations (*Fowler & Fields, 2014*; *Livesey & Marsh, 2020*). Theoretically, various computational perturbation-response methods have been developed and used to study the effects of ligand binding and mutations (*Yilmaz & Atilgan, 2000*; *Ikeguchi et al., 2005*; *Zheng & Brooks, 2005*; *Echave, 2008*; *Atilgan & Atilgan, 2009*).

Ligand binding can be modelled using forces applied to the protein residues involved in binding (*Ikeguchi et al., 2005*; *Atilgan & Atilgan, 2009*). This has been used to study various interesting problems. The most straightforward is predicting the conformational change induced by the binding of a ligand, when the binding site is known (*Ikeguchi et al., 2005*; *Atilgan & Atilgan, 2009*; *Tamura & Hayashi, 2015*). A related application is the prediction of ligand-binding sites related to known or desired deformations (*Atilgan et al., 2010*; *Jalalypour et al., 2020*). Another important application is the identification of

Corresponding author
Julian Echave,
jechave@unsam.edu.ar

allosteric sites and allosteric communication networks (*General et al., 2014*; *Alfayate et al., 2019*; *Lake et al., 2020*).

Mutations can also be modelled as forces and predicting the resulting responses *Echave (2008)*. Mutation-response computations have been used for various problems. One example is the analysis and prediction of pathological mutations (*Nevin Gerek, Kumar & Banu Ozkan, 2013*; *Tiberti et al., 2018*). Another major application is the study of patterns of protein evolutionary divergence (*Echave, 2008*; *Echave & Fernández, 2010*; *Nevin Gerek, Kumar & Banu Ozkan, 2013*; *Marcos & Echave, 2020*).

In this paper, I focus on mutation-response methods. I consider two cases, mutation-reponse scanning and double mutation-response scanning. In mutation-response scanning, protein sites are scanned over, for each site many random mutations (modelled as forces) are introduced, the resulting deformations are calculated, and deformations are averaged over to obtain a sensitivity matrix, **S** (*Echave, 2008*; *General et al., 2014*) (Element S_{ij} of **S** measures the mean structural deformation of site i due to mutations at site j.) In double mutation-response scanning, pairs of sites are scanned over, random mutations are introduced, the resulting deformations are calculated, and the minimum deformations are used to calculate a compensation matrix, **D** (*Tiberti et al., 2018*) (Element D_{ij} of **D** measures the degree to which mutating site i can be compensated by mutating site j.) Because they are based on averaging and maximizing over several simulated mutations, I will call the previous methods *simulation-based Mutation-Response Scanning* (sMRS) and *simulation-based Double Mutation-Response Scanning* (sDMRS).

The previous simulation-based methods are not very computationally costly for small to medium proteins. However, the computational cost of sMRS and sDMRS simulations increases with increasing protein size. Therefore, calculations may become prohibitive for very large systems (e.g., supra-molecular complexes, like a ribosome or a virus capsid) or large sets of proteins (e.g., scanning the whole human proteome to detect potential pathological mutations). To alleviate this problem, faster methods are needed.

The purpose of the present paper is to present faster alternatives to sMRS and sDMRS. This article presents two analytical methods, aMRS and aDMRS, that allow, respectively, the calculation of **S** and **D** using closed-formed analytical formulas, without performing simulations. In the following sections, I describe the simulation methods, derive the analytical alternatives, and assess the analytical methods by comparison with their simulation-based counterparts.

METHODS

In the following sections, I derive the formalism of Mutation Response Scanning (MRS) and Double Mutation Response Scanning (DMRS).

Covariance matrix

At finite temperature the protein fluctuates, sampling an ensemble of conformations. Let a specific backbone conformation be specified by the position vector $\mathbf{r} = (x_1, y_1, z_1, \ldots x_N, y_N, z_N)^T$, where (x_i, y_i, z_i) are the Cartesian coordinates of the alpha carbon (C_α) of site i, N is the number of sites, and super-index T denotes matrix or vector transposition.

The native ensemble can be characterized by the *native structure*, $\mathbf{r}^0 = \langle \mathbf{r} \rangle$, and by the *covariance matrix*:

$$\mathbf{C} \equiv \langle (\mathbf{r} - \mathbf{r}^0)(\mathbf{r} - \mathbf{r}^0)^T \rangle \tag{1}$$

where $\langle \cdots \rangle$ is the average over conformations.

The covariance matrix is determined by the protein's energy landscape. For simplicity, in this work I use the energy function of the Anisotropic Network Model (ANM) (*Atilgan et al., 2001*). This model represents the protein as a network of amino acids connected by harmonic springs. Specifically, each residue is represented by a single node placed at its C_α, and pairs of nodes that are within a cut-off distance R_0 are connected with springs of force-constant k. The ANM energy function is

$$V(\mathbf{r}) = \frac{1}{2} \sum_{ij} k(\|\mathbf{r}_j - \mathbf{r}_i\| - \|\mathbf{r}_j^0 - \mathbf{r}_i^0\|)^2 \tag{2}$$

where $\mathbf{r}_x$ is the position vector of node x, $\mathbf{r}^0_x$ its equilibrium position, k is the spring force constant, and the sum runs over all contacts ij.

The covariance matrix can be derived from Eq. (2). First, a second-order Taylor expansion of (2) leads to

$$V(\mathbf{r}) \approx \frac{1}{2} (\mathbf{r} - \mathbf{r}^0)^T \mathbf{K} (\mathbf{r} - \mathbf{r}^0) \tag{3}$$

where $\mathbf{K} = (d^2 V/d\mathbf{r}^2)_{\mathbf{r}=\mathbf{r}^0}$ is the Hessian matrix. Then, assuming a Boltzmann distribution of conformations $\rho(\mathbf{r}) = e^{-V(\mathbf{r})/k_B T}$ with $V(\mathbf{r})$ given by (3), it follows that

$$\mathbf{C} = k_B T \mathbf{K}^{-1} \tag{4}$$

where k_B is Boltzmann's constant, T the absolute temperature, and $\mathbf{K}^{-1}$ is the Hessian's pseudo-inverse ($\mathbf{K}$ is not invertible because it has 6 zero eigenvalues corresponding to rotations and translations). Given a protein of known native structure $\mathbf{r}^0$, and parameters R_0 and k, $\mathbf{K}$ is calculated differentiating (2), then $\mathbf{C}$ is obtained using (4).

Linear response approximation

The covariance matrix determines the conformational shift that results from applying a force to one or more protein atoms. An arbitrary force can be represented by a vector $\mathbf{f}$ with one component for each of the coordinates that represent the protein's conformation. For small $\mathbf{f}$, the structural response can be calculated using the Linear Response Approximation (LRA) (*Ikeguchi et al., 2005*; *Echave, 2008*):

$$\Delta \mathbf{r}^0 = \frac{\mathbf{C}}{k_B T} \mathbf{f} \tag{5}$$

Equation (5) allows the prediction of the effect of any given force $\mathbf{f}$ with the sole knowledge of $\mathbf{C}$.
Mutation-response scanning

The aim of Mutation-Response Scanning (MRS) is to analyse how protein structure responds to point mutations. In the methods that I consider here, given a protein, mutations are modelled using forces, the resulting structural responses are calculated using the Linear Response Approximation, and these responses are averaged over mutations to calculate a sensitivity matrix **S** that quantifies the mutation-response patterns.

Mutations as forces

Point mutations can be modelled by forcing the contacts of the mutated site (*Echave, 2008*). Let j be the site to mutate, $C(j)$ be the set of contacts of j, and jl the contact between j and l. Then, a mutation is modelled by applying a force

$$\mathbf{f}(j) = \sum_{jl \in C(j)} \mathbf{f}(jl) \tag{6}$$

where $\mathbf{f}(jl)$ is the force applied to contact jl. Let $f(jl)$ be a scalar and $\mathbf{e}_{jl}$ a unit vector directed from j to l. Then, $\mathbf{f}(jl)$ consists of a force $f(jl)\mathbf{e}_{jl}$ applied to l, plus a reaction force $-f(jl)\mathbf{e}_{jl}$ applied to j, and no force applied to other sites.

A random mutation at site j is modelled by picking independent random numbers $f(jl)$ and building $\mathbf{f}(jl)$ and $\mathbf{f}(j)$ (Eq. (6)). Following previous work (*Echave, 2008*; *Echave & Fernández, 2010*; *Marcos & Echave, 2020*), I use

$$f(jl) \sim N(0, \sigma^2) \tag{7}$$

Thus, the contact forces are picked from independent identical normal distributions.

Sensitivity matrix, S

What is the effect on a site i of mutating a site j? Consider a random mutation at site j, represented by a force $\mathbf{f}(j)$. Then, from (5), the structural deformation due to this mutation is given by

$$\Delta \mathbf{r}^0(j) = \frac{\mathbf{C}}{k_B T} \mathbf{f}(j) \tag{8}$$

$\Delta \mathbf{r}^0(j)$ can be written:

$$\Delta \mathbf{r}^0(j) = \begin{pmatrix} \Delta \mathbf{r}_1^0(j) \\ \vdots \\ \Delta \mathbf{r}_N^0(j) \end{pmatrix} \tag{9}$$

where $\Delta \mathbf{r}^0{}_i(j)$ is the 3×1 column vector that contains the change in Cartesian coordinates of site i caused by mutation $\mathbf{f}(j)$ applied to site j. Therefore, the magnitude of the effect of the mutation on the structure of site i may be quantified by the Euclidean norm $||\Delta \mathbf{r}_i^0(j)^2||$.

The *sensitivity matrix* **S** is the matrix with elements

$$S_{ij} = \left\langle ||\Delta \mathbf{r}_i^0(j)||^2 \right\rangle \tag{10}$$

where i is the response site, j the mutated site, and $\langle\cdots\rangle$ stands for averaging over mutations. S_{ij} represents the structural response of site i averaged over mutations at site j. Mutation-response scanning is the calculation of the sensitivity matrix $\mathbf{S}$ defined by 10.

Simulation-based mutation-response scanning

The sensitivity matrix $\mathbf{S}$ can be obtained using the simulation-based Mutation-Response Scanning method, `sMRS`. Given a protein's pdb file, this numerical method proceeds as follows.

1. **Set parameters.** Set parameters k and R_0 of the ANM model, parameter σ used to generate forces (Eq. (7)), and a desired number of mutations to apply to each site, M.
2. **Calculate the covariance matrix.** Read protein coordinates from the pdb file, for all pairs of sites calculate $C_\alpha - C_\alpha$ distances, compare them with R_0 to define contacts, then calculate the elastic network's matrix $\mathbf{K}$ using (2) and (3). Finally, invert this matrix to calculate $\mathbf{C}$ using (4).
3. **Generate mutational forces.** For each site j, generate $\mu = 1 \cdots M$ mutational force vectors $\mathbf{f}(j, \mu)$ using (6) and (7).
4. **Calculate mutational deformations.** For each mutational force $\mathbf{f}(j,\mu)$, calculate the resulting response $\Delta\mathbf{r}_i^0(j, \mu)$.
5. **Calculate the sensitivity matrix.** Average the deformations $\Delta\mathbf{r}_i^0(j, \mu)$ over mutations μ to obtain element S_{ij} of the sensitivity matrix $\mathbf{S}$, according to (10).

Analytical formula for the sensitivity matrix

In this section, I derive an analytical formula that allows the direct calculation of the sensitivity matrix, $\mathbf{S}$, without performing simulations.

The first step is to consider the deformation caused by forcing a single contact. Let $\mathbf{f}(jl)$ be a force applied along contact jl, composed by a force $f(jl)\mathbf{e}_{jl}$ applied to l and a reaction force $-f(jl)\mathbf{e}_{jl}$ applied to j. Replacing $\mathbf{f}(jl)$ into (5) and using (9), leads to

$$\Delta\mathbf{r}_i^0(jl) = \left(\mathbf{C}_{il} - \mathbf{C}_{ij}\right)\mathbf{e}_{jl}f(jl) \tag{11}$$

where $\Delta\mathbf{r}_{i}^0(jl)$ is the structural shift of site i caused by $\mathbf{f}(jl)$ and $\mathbf{C}_{xy}$ is the 3×3 block of $\mathbf{C}$ corresponding to the covariance between sites x and y.

Second, the deformation resulting from mutating a site is the sum of the deformations caused by forcing its contacts. From (6), (8), and (9), it follows that

$$\Delta\mathbf{r}_i^0(j) = \sum_{jl\in C(j)} \Delta\mathbf{r}_i^0(jl) \tag{12}$$

where $\Delta\mathbf{r}_i^0(j)$ is the shift of i due to mutating j and the sum runs over all contacts of j. Replacing (11) into (12), leads to

$$\Delta\mathbf{r}_i^0(j) = \sum_{jl\in C(j)} \left(\mathbf{C}_{il} - \mathbf{C}_{ij}\right)\mathbf{e}_{jl}f(jl) \tag{13}$$

Finally, an analytical formula for the direct calculation of the sensitivity matrix may be derived. Replacing (13) into (10), leads to

$$
\begin{aligned}
S_{ij} &\equiv \langle \| \Delta \mathbf{r}_i^0(j) \|^2 \rangle \\
&= \sum_{jk \in C(j)} \sum_{jl \in C(j)} \Delta \mathbf{r}_i(jk)^T \Delta \mathbf{r}_i(jl) \\
&= \sum_{jk \in C(j)} \sum_{jl \in C(j)} \langle f(jk) f(jl) \rangle \mathbf{e}_{jk}^T (\mathbf{C}_{ik} - \mathbf{C}_{ij})^T (\mathbf{C}_{il} - \mathbf{C}_{ij}) \mathbf{e}_{jl}
\end{aligned}
\tag{14}
$$

where $\langle \cdots \rangle$ stands for averaging over mutations at j. Since $f(jl) \sim N(0, \sigma^2)$ are independent identically distributed random variables ("Mutations as forces"), it follows that

$$
\langle f(jk) f(jl) \rangle = \sigma^2 \delta_{jk,jl}
\tag{15}
$$

where δ_{xy} is the Kronecker delta, which is 1 for $x = y$ and 0 otherwise. Therefore, replacing (15) into (14), leads to

$$
S_{ij} = \sigma^2 \sum_{jl \in C(j)} \mathbf{e}_{jl}^T (\mathbf{C}_{il} - \mathbf{C}_{ij})^T (\mathbf{C}_{il} - \mathbf{C}_{ij}) \mathbf{e}_{jl}
\tag{16}
$$

This equation allows the calculation of the sensitivity matrix.

Analytical mutation-response scanning

The analytical Mutation-Response Scanning method, `aMRS` calculates the sensitivity matrix **S** using the analytical formula (16). Given a protein's pdb file, this method proceeds as follows.

1. **Set parameters.** Set the parameters k and R_0 of the ANM model, and the parameter σ that defines the distribution of forces (Eq. (7)).
2. **Calculate the covariance matrix.** Read protein coordinates from the pdb file, for all pairs of sites calculate $C_\alpha - C_\alpha$ distances, compare them with R_0 to define contacts, then calculate the elastic network's matrix **K** using (2) and (3). Finally, invert this matrix to calculate **C** using (4).
3. **Calculate the sensitivity matrix.** Calculate the elements S_{ij} of the sensitivity matrix **S** using (16).

Double mutation-response scanning

The aim of Double Mutation-Response Scanning (DMRS) is to analyse how protein structure responds to pairs of point mutations. Just as for the MRS methods described above, the DMRS methods that I consider in this section model mutations using forces and calculate structural responses using the Linear Response Approximation. These responses are used to calculate a compensation matrix **D** that quantifies the degree of structural compensation between pairs of mutations.

Compensation matrix

In this subsection, I define the compensation matrix that DMRS aims to calculate. Let $\Delta \mathbf{r}^0(i\mu)$ be the structural response to a mutation μ at site i, and $\Delta \mathbf{r}^0(j\nu)$ be the structural

response to a mutation v at j. The deformation due to introducing both mutations is given by

$$\Delta \mathbf{r}^0(i\mu, jv) = \Delta \mathbf{r}^0(i\mu) + \Delta \mathbf{r}^0(jv) \tag{17}$$

and the magnitude of this deformation is given by

$$\|\Delta \mathbf{r}^0(i\mu, jv)\|^2 = \|\Delta \mathbf{r}^0(i\mu)\|^2 + \|\Delta \mathbf{r}^0(jv)\|^2 + 2\Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv) \tag{18}$$

The first two terms are positive, but the third term may be positive or negative. When the third term is negative, the mutations will compensate each other. Given a first mutation $i\mu$, the maximum compensation due to a second mutation at j is obtained when $\Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv)$ is minimum. Therefore, the degree of compensation may be quantified by $\min_v \Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv)$. For mutations modelled as forces, this is equal to minus the maximum, because if a force maximizes the dot-product, the opposite force, which is as likely, minimizes it. Therefore, to keeps things positive, it is convenient to define the compensating power of j by $\max_v [\Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv)]^2$. With the help of this equation, I define a compensation matrix, $\mathbf{D}$, with elements D_{ij} given by

$$D_{ij} = \left\langle \max_v [\Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv)]^2 \right\rangle_\mu^{\frac{1}{2}} \tag{19}$$

where $\langle \cdots \rangle_\mu$ is the average over μ. D_{ij} is a positive number that quantifies the degree to which mutating j can compensate the structural effect of mutating i.

Forces for double mutation-response scanning

The choice of forces used to model mutations in "Mutations as forces" is not appropriate for calculating the compensation matrix because the maximum involved is ill defined. The value of $\Delta \mathbf{r}^0(i\mu)^T \Delta \mathbf{r}^0(jv)$ is proportional to the lengths of force vectors $\mathbf{f}(i\mu)$ and $\mathbf{f}(j\mu)$. Defined as described in "Mutations as forces", the lengths of these vectors may become arbitrarily large, making the maximum in (19) infinite. To fix this, I apply the additional constraint

$$\|\mathbf{f}(x)\|^2 = \sigma^2 CN(x) \tag{20}$$

where σ^2 is the parameter used to define contact forces (see Eq. (7)) and $CN(x)$ is the number of contacts of site x. In practice, this is achieved by picking the forces as before, then renormalizing them. The norm of these forces is finite and the maximum of (19) is well defined.

Simulation-based double mutation-response scanning

The compensation matrix may be obtained using the method simulation-based Double Mutation-Response Scanning, sDMRS, which proceeds as follows.

1. **Set parameters.** Set parameters k and R_0 of the ANM model, parameter σ used to generate forces (Eq. (7)), and a desired number of mutations to apply to each site, M.

2. **Calculate the covariance matrix.** Read protein coordinates from the pdb file, for all pairs of sites calculate $C_\alpha - C_\alpha$ distances, compare them with R_0 to define contacts, then calculate the elastic network's matrix $\mathbf{K}$ using (2) and (3). Finally, invert this matrix to calculate $\mathbf{C}$ using (4).

3. **Generate mutational forces.** For each site i, generate $\mu = 1 \cdots M$ mutational force vectors $\mathbf{f}(i\mu)$ using (6), (7), and (20).

4. **Calculate mutational deformations.** For each mutational force $\mathbf{f}(i\mu)$, calculate the resulting response $\Delta\mathbf{r}^0(i\mu)$.

5. **Calculate the compensation matrix.** For each pair $(i\mu, jv)$, calculate $\Delta\mathbf{r}^0(i\mu)^T\Delta\mathbf{r}^0(jv)$, maximize over v, and average over μ to obtain the elements of the compensation matrix $\mathbf{D}$, according to (19).

Analytical formula for the compensation matrix

In this section, I derive an analytical formula that allows the direct calculation of the compensation matrix, $\mathbf{D}$, without performing simulations.

The first step is to consider the overlap between two deformations, $\Delta\mathbf{r}^0(i)^T\Delta\mathbf{r}^0(j)$. Consider two mutations, at sites i and j, represented by forces $\mathbf{f}(i)$ and $\mathbf{f}(j)$, respectively. From (6) and (8), it follows that

$$
\begin{aligned}
\Delta\mathbf{r}^0(i) &= \sum_{ik \in C(i)} (\mathbf{C}_k - \mathbf{C}_i)\mathbf{e}_{ik}f(ik) \\
\Delta\mathbf{r}^0(j) &= \sum_{jl \in C(j)} (\mathbf{C}_l - \mathbf{C}_j)\mathbf{e}_{jl}f(jl)
\end{aligned}
\tag{21}
$$

where $\Delta\mathbf{r}^0(x)$ is the protein's deformation due to mutating site x, $\mathbf{C}_x$ is the $3N \times 3$ block of $\mathbf{C}$ with the 3 columns corresponding to site x, and $f(xy)$ is the scalar force applied to contact xy. From (21), the overlap between two deformations is given by

$$
\Delta\mathbf{r}^0(i)^T\Delta\mathbf{r}^0(j) = \sum_{ik \in C(i)} \sum_{jl \in C(j)} f(ik)f(jl)\mathbf{e}_{ik}^T(\mathbf{C}_k - \mathbf{C}_i)^T(\mathbf{C}_l - \mathbf{C}_j)\mathbf{e}_{jl}
\tag{22}
$$

For simplicity of notation, it is convenient to rewrite this equation in matrix form:

$$
\Delta\mathbf{r}^0(i)^T\Delta\mathbf{r}^0(j) = \mathbf{f}(i)^T\mathbf{A}_{ij}\mathbf{f}(j)
\tag{23}
$$

where $\mathbf{f}(i)$ is a column vector whose elements are the $CN(i)$ contact forces $f(ik)$, $\mathbf{f}(j)$ is the column vector with $CN(j)$ elements $f(jl)$, and $\mathbf{A}_{ij}$ is a matrix of size $CN(i) \times CN(j)$ with elements

$$
A_{ik,jl} \equiv \mathbf{e}_{ik}^T(\mathbf{C}_k - \mathbf{C}_i)^T(\mathbf{C}_l - \mathbf{C}_j)\mathbf{e}_{jl}
\tag{24}
$$

At this point it is easy to derive a formula for the compensation matrix. The maximum of $\left[\Delta\mathbf{r}^0(i)^T\Delta\mathbf{r}^0(j)\right]^2$, subject to the constraint $\mathbf{f}(j)^2 = \sigma^2 CN(j)$ (Eq. (20)) can be shown to be

$$
\max\left[\Delta\mathbf{r}^0(i)^T\Delta\mathbf{r}^0(j)\right]^2 = CN(j)\mathbf{f}(i)^T\mathbf{A}_{ij}\mathbf{A}_{ij}^T\mathbf{f}(i)
\tag{25}
$$

Then, replacing (25) into (19), and using (15), leads to:

$$D_{ij} = \sigma^2 \sqrt{\mathrm{CN}(j) \mathrm{Tr} \mathbf{A}_{ij} \mathbf{A}_{ij}^T} \tag{26}$$

where Tr is the trace operator. This equation allows the calculation of the compensation matrix.

Analytical double mutation-response scanning

The analytical Double Mutation-Response Scanning method, `aDMRS`, calculates the compensation matrix $\mathbf{D}$ using the analytical formula (26). Given a protein's pdb file, this method proceeds as follows.

1. **Set parameters.** Set the parameters k and R_0 of the ANM model, and the parameter σ that defines the distribution of forces (Eq. (7)).
2. **Calculate the covariance matrix.** Read protein coordinates from the pdb file, for all pairs of sites calculate $C_\alpha - C_\alpha$ distances, compare them with R_0 to define contacts, then calculate the elastic network's matrix $\mathbf{K}$ using (2) and (3). Finally, invert this matrix to calculate $\mathbf{C}$ using (4).
3. **Calculate the compensation matrix.** Calculate the elements D_{ij} of the compensation matrix $\mathbf{D}$ using (26).

Implementation

In the present work, `sMRS` (Simulation-based Mutation-Response Scanning), `aMRS` (Analytical Mutation-Response Scanning), `sDMRS` (Simulation-based Double Mutation-Response Scanning), and `aDMRS` (Analytical Double Mutation-Response Scanning) were implemented using the R language. As much as possible, the code was optimised by using the linear algebra functions of the BLAS and LAPACK packages. For implementation details see available code.

Parameters

The parameter values used in the present paper are $R_0 = 12.5$ Å, $k = 1/\text{Å}^2$, and $\sigma = 0.3/\text{Å}$. With the chosen R_0 value, previous work found good agreement between predicted and empirical structural deformations *Marcos & Echave (2020)*. Regarding k, energy units are arbitrarily chosen so that $k = 1/\text{Å}^2$. The precise values of k and σ do not affect the present results because they have a mere scaling effect on the sensitivity matrix and the compensation matrix (It can easily be proved that both matrices are proportional to $\frac{\sigma^2}{k^2}$).

Dataset

Table 1 summarises the dataset used to assess the methods developed in this work. The structure files for the calculations were obtained from the Protein Data Bank for d2l8ma and d2acya, and from the Homstrad database for the other proteins (*Stebbings & Mizuguchi, 2004*). I use the 8 Homstrad proteins because mutation-response simulations were tested against empirical data for these proteins in a recent study (*Marcos & Echave, 2020*). I added the other two proteins, with which I am familiar from other studies, to

complete the dataset: d2acya to have a second representative of the alpha & beta SCOP structural class and 2l8ma to add a large protein to the dataset.

RESULTS

Mutation-response scanning

This section assesses the analytic Mutation-Response Scanning method (aMRS) by comparison with the simulation-based Mutation-Response Scanning method (sMRS). These methods were described in detail in "Methods". Briefly, for a given protein, an sMRS simulation consists in subjecting each of the protein sites j to M mutations, calculating the resulting structural deformation of each site i, and averaging these deformations over mutations to obtain the elements S_{ij} of a sensitivity matrix **S** (see "Simulation-based Mutation-Response Scanning"). The analytical method, aMRS, calculates **S** using the closed analytical expression Eq. (16), avoiding the need of simulations (see "Analytical Mutation-Response Scanning"). Methods are compared on the proteins of Table 1.

sMRS *converges rapidly towards* aMRS

I compare aMRS with sMRS for the proteins of Table 1. The point of this work is to assess whether the analytical method is faster than the simulation method. However, since the calculations performed with the simulation method depend on the number of mutations per site, M, before addressing computational cost, I consider the convergence of sMRS calculations.

   Theoretically, sMRS and aMRS are equivalent ways of calculating the sensitivity matrix **S**. Specifically, in the limit of an infinite number of mutations per site, $M \rightarrow \infty$, the sMRS **S** should converge towards the aMRS **S**. To study this convergence, Fig. 1, compares simulated and analytical matrices for the example case of Phospholipase A2 (SCOP id d1jiaa) (Similar figures for the other proteins studied can be found in Supplemental_info.pdf). For the d1jiaa example, sMRS converges rapidly towards the aMRS matrix as M increases (Fig. 1C), so that the sMRS matrix calculated with $M = 200$ is very similar to the aMRS matrix (Fig. 1A and Fig. 1B).

   For the other proteins the results are similar. Thus, for all cases the sMRS matrix converges rapidly towards the aMRS matrix (see grey lines of Fig. 1C). For $M = 200$, the correlation coefficient between sMRS and aMRS matrices is 1.00 for all proteins (Table 2). Thus, the sMRS sensitivity matrix converges rapidly with increasing M, so that with $M = O(10^2)$ it is very similar to the aMRS matrix.

   To further assess convergence, I consider sMRS and aMRS profiles. Site-dependent profiles are obtained by averaging the sensitivity matrix over rows or columns. Averaging over rows leads to an *influence profile*, with elements $S_j \equiv 1/N \sum_i^N S_{ij}$ that measure the average influence of mutating j. Averaging over columns leads to a *sensitivity profile*, with elements $S_i \equiv 1/N \sum_j^N S_{ij}$ that measure the sensitivity of site i with respect to mutations elsewhere.

   Figure 2 compares sMRS and aMRS profiles for Phospholipase A2 (d1jiaa) (Similar figures for the other proteins studied can be found in Supplemental_info.pdf). Comparing influence profiles, we see that sMRS with $M = 200$ and aMRS profiles are very similar

domain	family	class	N
d1lcka1	SH3 domain	All beta	54
d1ntxa	Snake venom toxins	Small	60
d1fxla2	Canonical RNA-binding domain	Alfa & beta	82
d1bxva	Plastocianine/Azurin-like	All beta	91
d2acya	Acyl-phosphatase-like	Alpha & beta	98
d1jiaa	Vertebrate Phospholipase A2	All alpha	122
d1hmta	Fatty acid binding protein-like	All beta	131
d1a4fb	Globines	All alpha	146
d1mcta	Eukaryotic proteases	All beta	223
d2l8ma	Cytochrome P450	All alpha	405

Table 1 Protein data set.

Note:
Columns show, in order, protein domain id, family, and structural class according to the SCOP classification (*Murzin et al., 1995*), and protein length N.

Sensitivity matrix, d1jiaa
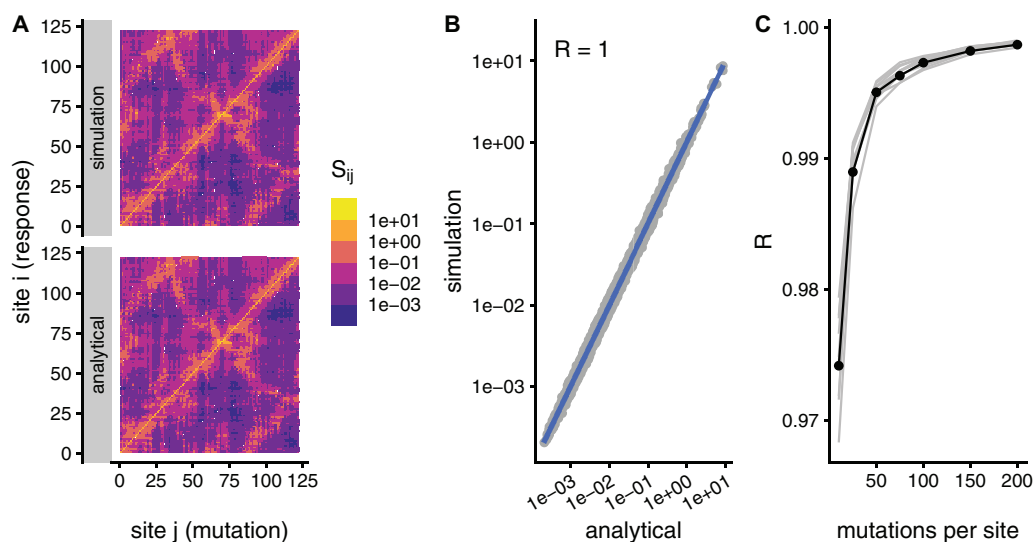
Figure 1 Comparison between sMRS and aMRS sensitivity matrices. Results shown for Phospholipase A2 (d1jiaa). The sensitivity matrix S has elements S_{ij} that measure the structural shift of site i averaged over mutations at site j. sMRS is a simulation-based Mutation Response Scanning method that calculates S by averaging over simulated point mutations. aMRS is an analytical method that calculates S using a closed formula. (A) sMRS response matrix obtained by averaging over 200 mutations (simulation) compared with the aMRS matrix (analytical). (B) Scatterplot of the sMRS vs. aMRS matrix elements of A. (C) Convergence of sMRS with increasing number of mutations per site. In C the d1jiaa case is shown with black lines and points, and the other 9 proteins studied are shown with grey lines. Matrix elements Si j are normalised so that their average is 1. Logarithmic scale is used in A and B and R is the Pearson correlation coefficient between the log-transformed sMRS and aMRS matrices.

(Fig. 2A and Fig. 2B) and that sMRS influence profiles converge rapidly towards the corresponding aMRS profiles as M increases (Fig. 2C). Similarly, the sensitivity profile estimated by sMRS with $M = 200$ is also very similar to its aMRS counterpart (Figs. 2D and

Table 2 aMRS vs. sMRS summary.

protein	N	t_{sMRS}	t_{aMRS}	R	R_i	R_j
d1lcka1	54	6.26	0.03	1.00	1.00	0.99
d1ntxa	60	7.18	0.05	1.00	1.00	1.00
d1fxla2	82	11.22	0.07	1.00	1.00	1.00
d1bxva	91	12.44	0.07	1.00	1.00	0.97
d2acya	98	18.08	0.07	1.00	1.00	0.98
d1jiaa	122	18.77	0.12	1.00	1.00	0.99
d1hmta	131	21.16	0.11	1.00	1.00	0.99
d1a4fb	146	26.02	0.18	1.00	1.00	0.99
d1mcta	223	54.08	0.37	1.00	1.00	0.99
d2l8ma	405	180.91	1.47	1.00	1.00	0.99

Note:

N: protein length; t_{sMRS}: CPU time of sMRS in seconds; t_{aMRS}: CPU time of aMRS in seconds. Convergence measures at $M = 200$ mutations per site: R: correlation coefficient between sMRS and aMRS sensitivity matrices; R_i: correlation between sensitivity profiles; R_j: correlation between influence profiles.

2E) and the sMRS profile converges rapidly towards the aMRS profile as M increases (Fig. 2F).

Similar results are found for the other proteins studied. The convergence of influence profiles (grey lines of Fig. 2C) is somewhat slower than that of sensitivity profiles (grey lines of Fig. 2F), but in both cases there is good convergence. For $M = 200$, Pearson's correlation between sMRS and aMRS influence profiles is in the range $0.97 \leq R \leq 1.00$ and the correlation between sensitivity profiles is 1.00 for all proteins (Table 2). In summary, sMRS influence and sensitivity profiles converge rapidly, so that with $M = O(10^2)$ they are very similar to their aMRS counterparts.

aMRS *is much faster than* sMRS

The purpose of this paper is to develop a faster mutation-response scanning method. To see whether aMRS is indeed faster than sMRS, Fig. 3 compares their computational cost. An sMRS calculation using a typical number of $M = 200$ mutations per site is much slower than an aMRS calculation (Fig. 3A). The computational cost, as measured by CPU time, scales with protein length as $N^{1.5}$ for both sMRS and aMRS. As a result, t_{sMRS} increases linearly with t_{aMRS} with a slope that is the speedup of aMRS vs. sMRS; For the M = 200 case, this speedup is $t_{\text{sDMRS}}/t_{\text{aDMRS}} \approx 126$ (Fig. 3B). Further, the speedup increases linearly with M: $t_{\text{sMRS}}/t_{\text{aMRS}} \propto M$ (Fig. 3C). Thus, the analytical method provides a speedup of the order of the number of mutations per site, which is typically in the hundreds. In a word, aMRS is much faster than sMRS.

Double mutation-response scanning

This section assesses the analytical Double Mutation-Response Scanning method (aDMRS) by comparison with the simulation-based Double Mutation-Response Scanning method (sDMRS). These methods are alternative ways of calculating a compensation matrix **D**. This matrix is composed by elements D_{ij} that measure the degree to which mutating site j may compensate the structural deformation due to a first mutation at site i (Eq. (19)).

Influence and sensitivity profiles, d1jiaa
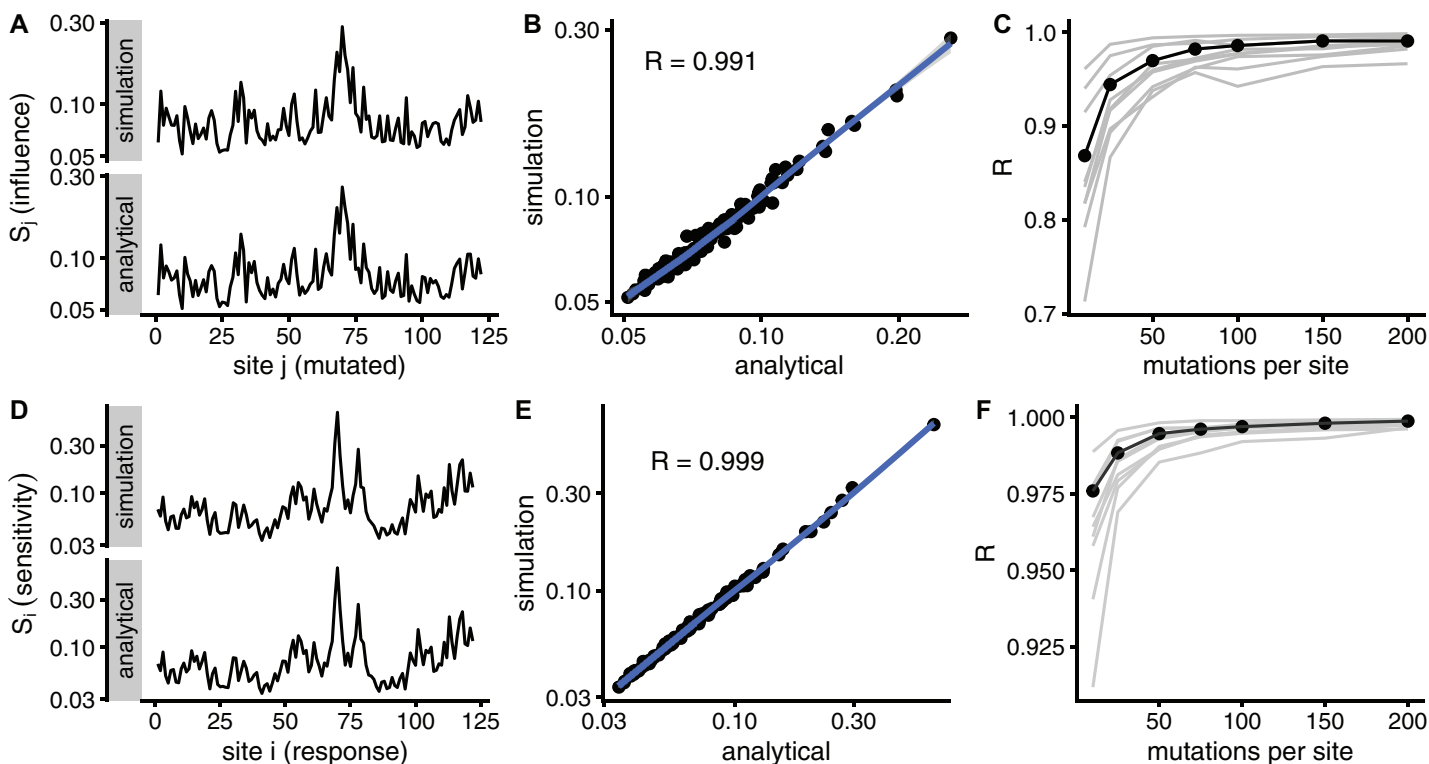
Figure 2 Comparison of sMRS and aMRS marginal profiles. Results shown for Phospholipase A2 (d1jiaa). The influence profile is the average of the sensitivity matrix over rows; element S_j measures the average influence of mutations at site j. The sensitivity profile is the average of the response matrix over columns; element S_i measures the average sensitivity of site i. (A) Sj profiles obtained with sMRS using 200 mutations per site (simulation) and aMRS (analytical); (B) scatter plot of the sMRS vs. aMRS S_j values of A; (C) convergence of the sMRS S_j profile towards the aMRS profile. (D) Si profiles obtained with sMRS using 200 mutations per site (simulation) and aMRS (analytical); (E) scatter plot of the sMRS vs. aMRS S_i values of D; (F) convergence of the sMRS S_i profiles towards the aMRS profile. In C and F, the d1jiaa case is shown with black lines and points, and the other 9 proteins studied are shown using grey lines. Profiles were calculated using the normalised matrix (matrix average is 1). Profile elements are shown in logarithmic scale and R is the Pearson correlation coefficient between log-transformed sMRS and aMRS profiles.

The simulation method, sDMRS, obtains this matrix numerically scanning over pairs of simulated mutations (see "Simulation-based Double Mutation-Response Scanning"). The analytical method, aDMRS, calculates the compensation values using a closed formula (Eq. (26)), avoiding the use of simulations (see "Analytical Double Mutation-Response Scanning").

sDMRS *converges slowly towards* aDMRS

I compare aDMRS with aDMRS for the proteins of Table 1. As in "Mutation-Response Scanning", before addressing computational cost, I consider the convergence of the simulation method with increasing M.

In principle, the simulation and analytical methods are equivalent. The compensation matrix **D** calculated with sDMRS with $M \to \infty$ will be identical to the aDMRS matrix. However, in practice the sDMRS matrix depends on M. Figure 4 compares simulated and analytical compensation matrices for the example case of Phospholipase A2 (SCOP id
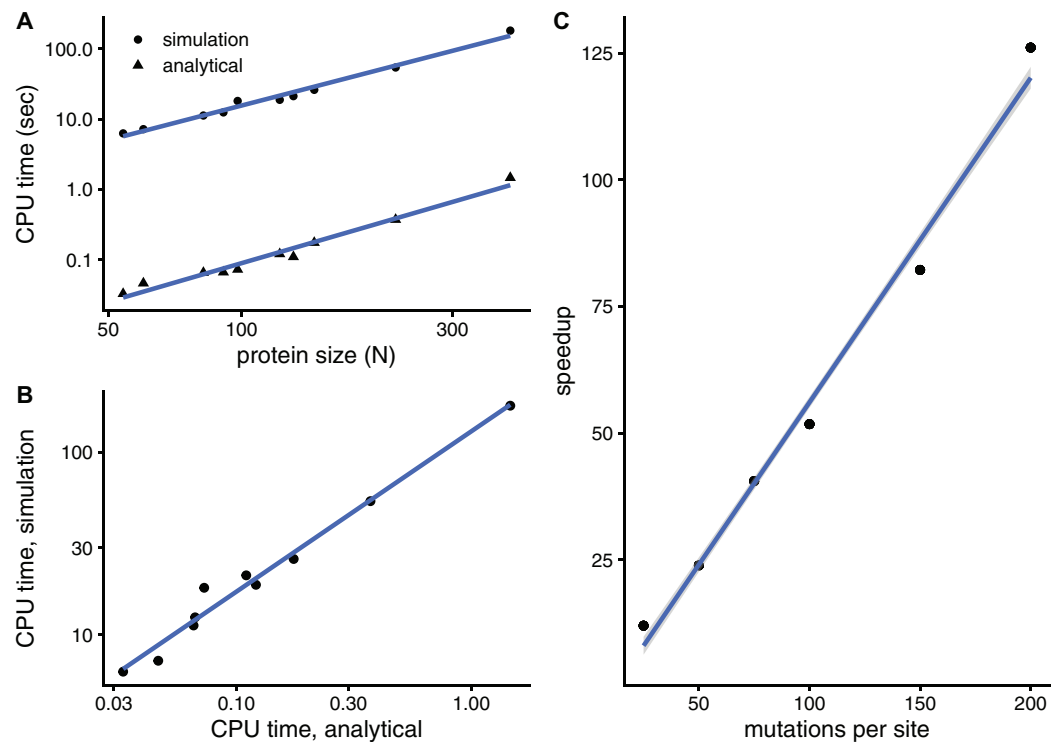
Figure 3 The analytical mutation-response scanning method (aMRS) is much faster than the simulation method (sMRS). (A) CPU time vs. protein size the for sMRS with 200 mutations per site (simulation) and for aMRS (analytical). Time is shown in logarithmic scale. From the slope of the linear fits it follows that both times scale with $N^{1.5}$ (N is the number of sites, each point is one protein). (B) The CPU time of the simulation method increases linearly with the CPU time of the analytical method, with a speedup of 126: $t_{sMRS} = 126 \times t_{aMRS}$. (C) The speedup, t_{sMRS}/t_{aMRS} obtained as shown in B, increases linearly with the number of mutations per site. Calculations were performed on the proteins of Table 1 using the methods implemented in R, with base LAPACK and the optimised AtlasBLAS libraries for matrix operations, on an early-2018 MacBook Pro notebook (processor i7-8850H).

d1jiaa) (Similar figures for the other proteins studied can be found in Supplemental_info. pdf). First, note that the compensation matrix obtained with sDMRS with $M = 200$ looks similar to the aDMRS matrix (Fig. 4A). More quantitatively, a scatter plot of sDMRS vs. aDMRS matrix elements shows good correlation, but there is a visible scattering of points around the linear fit (Fig. 4C). The similarity between sDMRS and aDMRS matrices can be measured by the correlation coefficient, which in this case is $R = 0.95$. Figure 4C shows that as M increases, the sDMRS matrix converges rapidly at first towards the aDMRS matrix, but convergence slows down with further increases of M. Thus, for Phospholipase A2, sDMRS with $O(10^2)$ mutations per site produces a compensation matrix that is in good agreement with, but not identical to, the aDMRS matrix.

A similar situation is found for the other proteins of the dataset. Convergence quickly slows down as M increases (see grey lines of Fig. 4C). For $M = 200$, the correlation between sDMRS and aDMRS matrices falls within the range $0.87 \leq R \leq 0.97$ (Table 3). Thus, the sDMRS compensation matrix converges slowly towards the aDMRS matrix, so that for $M = O(10^2)$ the simulated matrix is in moderate to good agreement with the analytical

Compensation matrix, d1jiaa
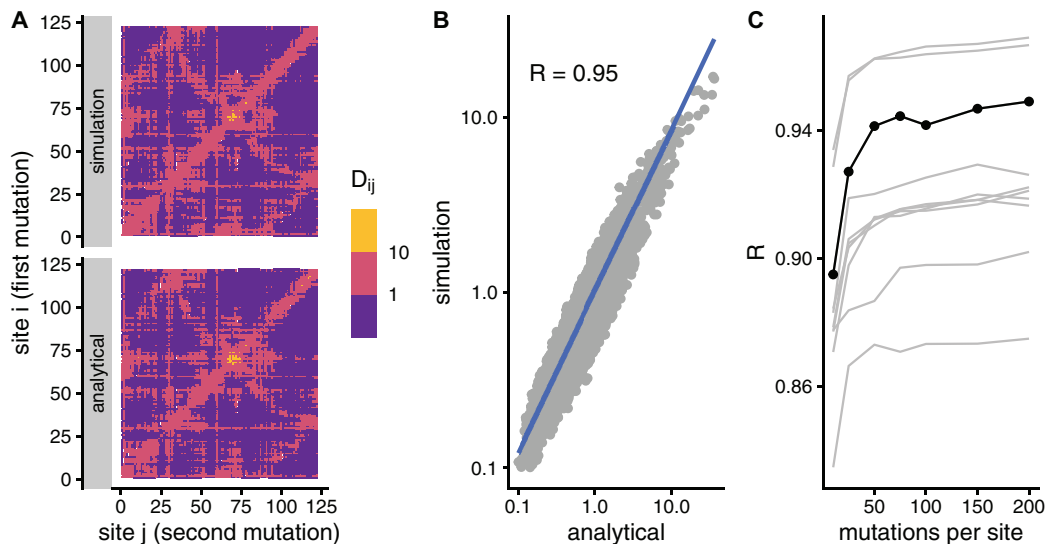
Figure 4 Comparison of sDMRS and aDMRS compensation matrices. Results shown for Phospholipase A2 (d1jiaa). The compensation matrix D has elements D_{ij} that measure the maximum compensation of the structural deformation due to a mutation at site i afforded by a second mutation at j. sDMRS is a simulation-based Double Mutation Response Scanning method that calculates D by maximizing the structural compensation over pairs of simulated mutations. aDMRS is an analytical method that calculates D using a closed formula. (A) sDMRS compensation matrix obtained using 200 mutations per site (simulation) compared with the aDMRS matrix (analytical). (B) Scatterplot of the sDMRS vs. aDMRS matrix elements of A. (C) Convergence of the sDMRS matrix towards the aDMRS matrix with increasing number of mutations per site. In C the d1jiaa case is shown with black lines and points, and the other 9 proteins studied are shown with grey lines. D_{ij} are normalised so that their average is 1, logarithmic scales are used in A and B, and R is Pearson's correlation coefficient between log-transformed sDMRS and aDMRS matrix elements.

Table 3 aDMRS vs. sDMRS summary.

protein	N	t_{sDMRS}	t_{aDMRS}	R	R_i	R_j
d1lcka1	54	22.24	0.21	0.87	0.76	0.78
d1ntxa	60	27.70	0.17	0.97	0.97	0.99
d1fxla2	82	59.58	0.43	0.93	0.89	0.94
d1bxva	91	77.46	0.60	0.92	0.55	0.69
d2acya	98	116.38	0.76	0.90	0.56	0.71
d1jiaa	122	167.89	1.24	0.95	0.85	0.94
d1hmta	131	203.65	1.30	0.97	0.77	0.93
d1a4fb	146	274.29	2.13	0.92	0.70	0.83
d1mcta	223	1,034.36	11.95	0.92	0.64	0.74
d2l8ma	405	12,995.91	56.53	0.92	0.61	0.77

Note:
 N: protein length; t_{sDMRS}: CPU time of sDMRS in seconds; t_{aDMRS}: CPU time of aDMRS in seconds. Convergence measures at $M = 200$ mutations per site: R: correlation coefficient between sDMRS and aDMRS compensation matrices **D**;
 R_i: correlation between D_i profiles; R_j: correlation between D_j profiles.

Compensation profiles, d1jiaa
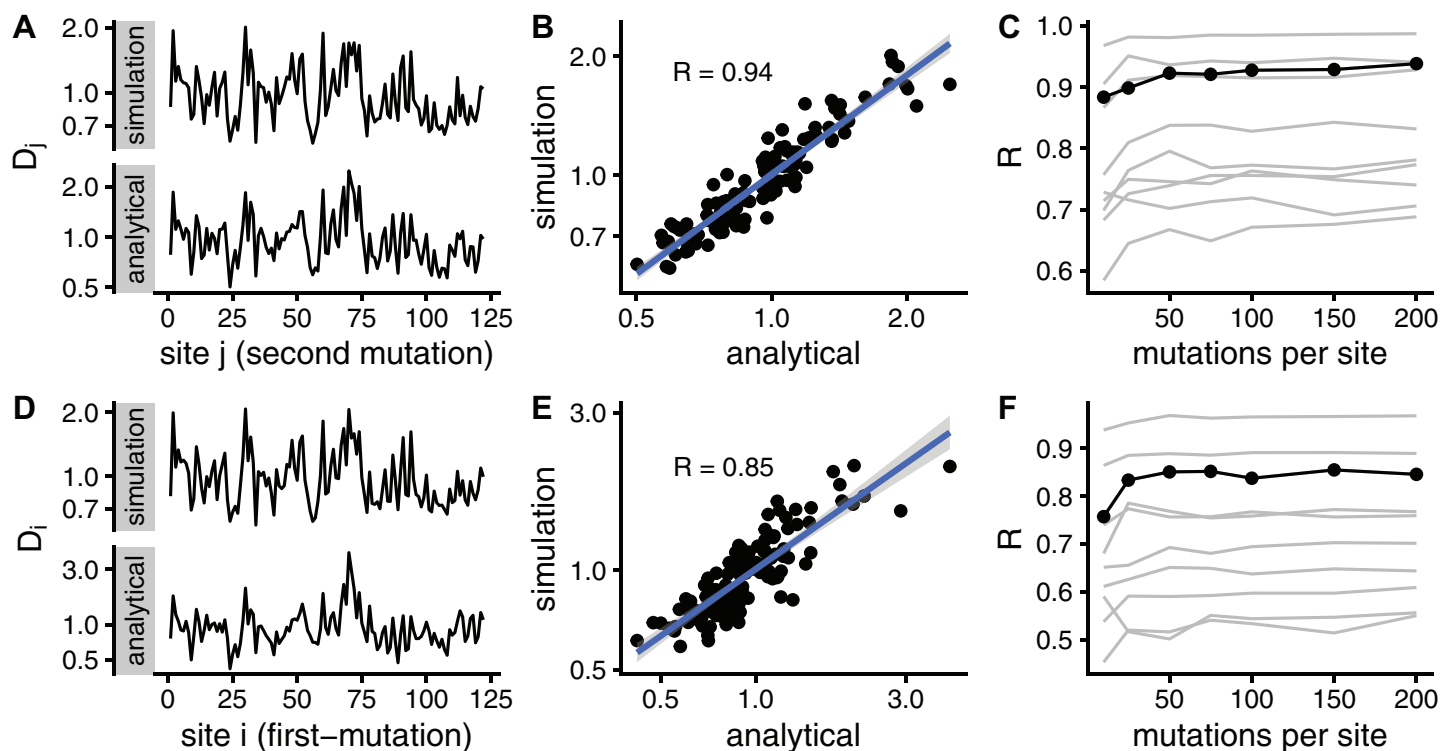
Figure 5 Comparison of sDMRS **and** aDMRS **marginal profiles. Results shown for Phospholipase A2 (d1jiaa). Two marginal profiles are considered.** The D_j profile is the average of the compensation matrix over rows; element D_j measures the ability of j to compensate mutations at other sites. The D_i profile is the average of the compensation matrix over columns; element D_i measures the degree to which a mutation at i can be compensated by mutations elsewhere. (A) sDMRS D_j profile obtained using 200 mutations per site (simulation) and aDMRS D_j profile (analytical); (B) scatter plot of the sDMRS vs. aDMRS D_j values of A; (C) convergence of the sDMRS D_j profile towards the aDMRS profile. (D) sDMRS D_i profile obtained using 200 mutations per site (simulation) and aDMRS D_i profile (analytical); (E) scatter plot of the sDMRS vs. aDMRS D_i values of D; (F) convergence of the sDMRS D_i profile towards the aDMRS profile. In C and F, the d1jiaa case is shown with black lines and points, and the other 9 proteins studied are shown with grey lines. Profiles were calculated with normalised matrices (matrix average is 1), they are in logarithmic scale, and R is the Pearson correlation coefficient between the log-transformed sDMRS and aDMRS profiles.

matrix. The degree of convergence is not clearly related to protein properties such as structural class or protein size, thus convergence should be tested whenever the simulation method is used.

I further assess convergence by considering site-dependent compensation profiles. Averaging **D** over rows, I obtain a D_j profile that measures the average compensation power of sites j. Averaging over columns, I obtain a D_i profile that measures how likely to be compensated mutations at i are. Figure 5 compares sDMRS and aDMRS profiles for Phospholipase A2 (d1jiaa). The $M = 200$ sDMRS profiles are visually similar to aDMRS profiles (Fig. 5A and Fig. 5D). The similarity is not very high, however: points are quite scattered around the linear fit in sDMRS vs. aDMRS plots (Fig. 5B and Fig. 5E). The convergence of sDMRS profiles towards their aDMRS counterparts is very slow (Fig. 5C and Fig. 5F).

Similar results are found for the other proteins studied. Profiles generally improve very slowly with increasing M (see grey lines of Fig. 5C and Fig. 5F). For $M = 200$, the
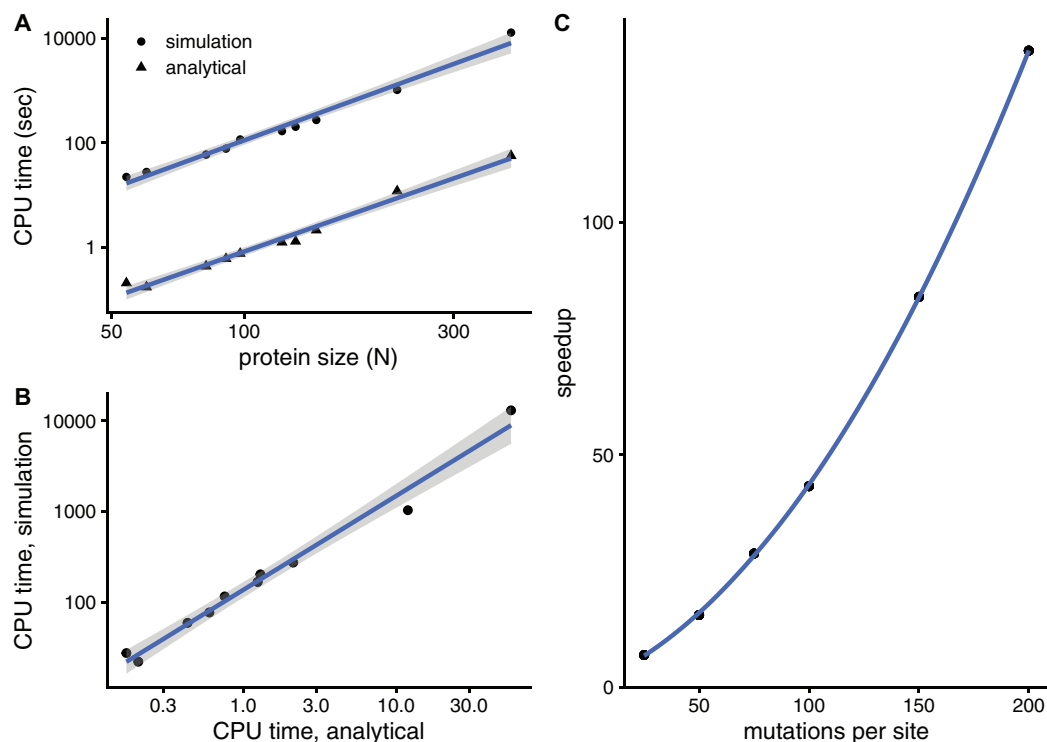
Figure 6 The analytical double mutation-response scanning method (aDMRS) is much faster than the simulation method (aDMRS). (A) CPU time vs. protein size for sDMRS with 200 mutations per site (simulation) and for aDMRS (analytical). Time is shown in logarithmic scale. From the slope of the linear fits it follows that both CPU times scale with N^3 (N is the number of sites, each point is one protein). (B) The CPU time of the simulation method increases linearly with the CPU time of the analytical method, with a speedup of 137: $t_{sDMRS} = 137 \times t_{aDMRS}$. (C) The speedup, t_{sDMRS}/t_{aDMRS}, increases non-linearly with the number of mutations per site M, tending towards $O(M^2)$ for large M. Calculations were performed on the proteins of Table 1 using the methods implemented in R, with base LAPACK and the optimised AtlasBLAS libraries for matrix operations, on an early-2018 MacBook Pro notebook (processor i7-8850H).

correlation coefficient between sDMRS and aDMRS D_i profiles falls in the range $0.55 \leq R \leq 0.97$ and between D_j profiles it falls in the range $0.69 \leq R \leq 0.99$ (Table 3). In summary, sDMRS profiles converge very slowly with increasing M, so that for $M = O(10^2)$, they are often poorly converged. In addition, There are no obvious determinants of convergence: R is not clearly determined by either protein size or structural class. Therefore, whenever the simulation method is used, convergence should be tested.

aDMRS *is much faster than* sDMRS

To see whether aDMRS is faster than aDMRS, Fig. 6 compares their computational cost. sDMRS with $M = 200$ mutations per site is much slower than aDMRS (Fig. 6A). The computational cost, as measured by CPU time, scales with protein length as N^3 for both sDMRS and aDMRS. As a result, t_{sDMRS} increases linearly with t_{aDMRS} with a slope that is the speedup of aDMRS vs. sDMRS. For the $M = 200$ case, $t_{sDMRS}/t_{aDMRS} \approx 137$ (Fig. 6B). The speedup increases non-linearly with M (Fig. 6C). This dependence can be understood from the sDMRS procedure schematised in "Simulation-based Double Mutation-Response

Scanning". The cost of generating the mutations (steps 3 and 4) increases linearly with M, while performing the average and maximization needed to calculate the compensation matrix (steps 5) scales as M^2. Therefore, for large M the analytical method provides a speedup of $O(M^2)$, making aDMRS much faster than sDMRS.

DISCUSSION

I have derived, implemented, and assessed two mutation-response scanning methods, aMRS and aDMRS, which are analytical alternatives to the simulation methods sMRS and sDMRS, respectively. All methods were implemented using R with optimized BLAS and LAPACK libraries. None of the methods posed major implementation difficulties.

The methods were assessed on a dataset of 10 proteins of varying lengths. First, I consider the convergence of simulation methods. In the limit if infinite mutations per site (M), simulation and analytical methods should give the same results. In practice, the degree of convergence of the simulation methods depends on M. sMRS converges rapidly towards aMRS, so that with a typical $M = O(10^2)$ the sDMRS sensitivity matrix and its marginal profiles are almost identical to those calculated with aMRS (Fig. 1C, Fig. 2C, Fig. 2F, Table 2). On the other hand, sDMRS converges slowly, so that even with $M = O(10^2)$ sDMRS convergence is not guaranteed (Fig. 4C, Fig. 5C, Fig. 5F, Table 3). sDMRS converges more slowly than sMRS because it is more difficult to find extreme values (calculation of the compensation matrix involves maximization over pairs of mutations) than averages (sensitivity matrix elements are averages over mutations). In general, when using simulation-based methods convergence should always be assessed. In contrast, since the analytical methods do not depend on M, there is no need to study convergence, and possible convergence issues are altogether avoided.

Beyond convergence, since the purpose of this work was to develop faster methods, the key finding is that the analytical methods are much faster than the simulation methods. For a typical case of $M = 200$ mutations per site, aMRS is 126 × faster than sMRS and aDMRS is 137 × faster than sDMRS. While the computational cost of sMRS is relatively modest and increases rather slowly in proportion to $N^{1.5} M$, sDMRS is much more computationally expensive and its cost rises steeply in proportion to $N^3 M^2$. The speedup of analytical methods is of $O(M)$ for single-mutation scans and $O(M^2)$ for double-mutation scans. This speedup may be most important for large proteins. For instance, for the 405-sites-long Cytochrome P450, an sMRS calculation takes 3 CPU min vs. 1.5 s of the alternative aMRS calculation (Table 2). On the other hand, an sDMRS calculation takes 3.6 h vs. 1 min of the alternative aDMRS calculation (Table 3). Therefore, there is a large speedup for both single and double mutation-response scans, that may be most useful for the later case.

To further compare the mutation-response scanning methods considered here, I discuss some of their main limitations. All methods are based on the Linear Response Approximation formula $\Delta \mathbf{r}^0 = \mathbf{Cf}$. Therefore, the main limitations are the validity of LRA, the quality of $\mathbf{C}$, and how well mutations can be modelled by the force $\mathbf{f}$. Regarding the first limitation, LRA will be valid if both perturbations ($\mathbf{f}$) and their responses ($\Delta \mathbf{r}^0$) are small. Thus LRA should be valid for most mutations, failing only in the rare cases in which

specific mutations induce very large conformational changes. Second, calculating **C** with a simple elastic network model, as done here, might impose additional limitations. However, this could be alleviated by calculating **C** using more sophisticated means, such as MD simulations, if necessary. More fundamentally, the main limitation is the very assumption that **C** characterizes the conformational ensemble, which will be the case for proteins with a single native structure, but may fail for proteins that have two or more stable conformations. The final limitation depends on whether mutations can be adequately modelled using forces (**f**). While it is possible that this fails for the prediction of specific mutations, mutations-as-forces models have been proved successful in many previous studies that depend on summary statistics such averages or maxima (*Echave, 2008*; *Echave & Fernández, 2010*; *Tiberti et al., 2018*; *Marcos & Echave, 2020*). For the present work, it should be noted that the limitations mentioned are common to the simulation methods and their analytical alternatives. The analytical approach adds no limitation to the list.

Given that limitations exist, it is worthwhile to discuss why this work has not validated the methods by comparison with empirical data. The main reason is that the aim of this work is not to develop mutation-response methods in better agreement with experiment, but to develop faster methods. This is why the assessment was performed by comparing between simulation and analytical approaches, rather than validating such approaches against empirical data. Validating mutation-response scanning itself is beyond the scope of this work. A second reason is that taking the validity of mutation-response scanning as a given is reasonable. For 8 of the proteins of Table 1, the mutation-response model of the present paper has been recently validated by comparison with empirical structural sensitivity profiles *Marcos & Echave (2020)*. More generally, the validity of perturbation-response methods follows from their extensive successful use in a variety of applications for at least 15 years, as mentioned in "Introduction".

The main conclusion of this work is that the analytical methods should be chosen over the simulation methods because they are faster and, in addition, they have no convergence issues. Therefore, the analytical methods should be useful for a wide range of potential applications, such as predicting evolutionary divergence of protein structures (*Echave & Fernández, 2010*; *Marcos & Echave, 2020*), detecting and interpreting pathological mutations (*Nevin Gerek, Kumar & Banu Ozkan, 2013*; *Raimondi et al., 2018*; *Verkhivker, 2019*), and detecting compensating mutations and rescue sites (*Tiberti et al., 2018*). The speedup afforded by the analytical methods would be especially helpful for treating otherwise intractable large proteins, protein complexes, and large protein databases.

To finish, I mention two possible lines of further development. A first line is to derive analytical expressions for the deformations caused by external forces applied to single sites, as in Perturbation-Response Scanning (PRS) (*Atilgan & Atilgan, 2009*; *General et al., 2014*) and Double Force Scanning (DFS) (*Tiberti et al., 2018*). This will be useful for applications related to ligand-binding induced deformations (*Atilgan et al., 2010*; *General et al., 2014*). Beyond deformations, a second line of development is to derive analytical alternatives to simulation-based methods that calculate effects of mutations on protein motions (*Hamacher, 2008*; *Zheng & Tekpinar, 2009*; *Zheng & Thirumalai, 2009*;

Echave, 2012; *Hamacher, 2008*). This would be important for studies of the role of protein dynamics in function and evolution (*Echave, 2012*; *Micheletti, 2013*; *Ponzoni & Bahar, 2018*; *Zhang et al., 2019*; *Zhang & Su, 2019*; *Wingert et al., 2021*).

Funding
JE is a researcher of CONICET. This work was supported by Consejo Nacional de Investigaciones Científicas y Técnicas (grant number PIP 112 201501 00385 CO) and by Agencia Nacional de Promoción Científica y Tecnológica (grant number PICT-2016-4209).There was no additional external funding received for this study. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures
The following grant information was disclosed by the authors:
Consejo Nacional de Investigaciones Científicas y Técnicas: PIP 112 201501 00385 CO.
Agencia Nacional de Promoción Científica y Tecnológica: PICT-2016-4209.

Competing Interests
The author declares that he has no competing interests.

Author Contributions
- Julian Echave conceived and designed the experiments, performed the experiments, analyzed the data, prepared figures and/or tables, authored or reviewed drafts of the paper, and approved the final draft.

Data Availability
   The data and code are available at Zenodo. Data: Echave, Julian. (2020). Fast and exact single and double mutation-response scanning of proteins: data and results (Version 1.0.0) [Data set]. Zenodo. DOI 10.5281/zenodo.4122623.
   Code: Echave, Julian. (2020, October 23). Fast and exact single and double mutation-response scanning of proteins: code (Version 1.0.0). Zenodo. DOI 10.5281/zenodo.4123149.

Supplemental Information
Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj.11330#supplemental-information.

REFERENCES

Alfayate A, Caceres CR, Dos Santos HGH, Bastolla U. 2019. Predicted dynamical couplings of protein residues characterize catalysis, transport and allostery. *Bioinformatics* **35(23)**:4971–4978 DOI 10.1093/bioinformatics/btz301.

Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I. 2001. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophysical Journal* **80(1)**:505–515 DOI 10.1016/S0006-3495(01)76033-X.

Atilgan C, Atilgan AAR. 2009. Perturbation-response scanning reveals ligand entry-exit mechanisms of ferric binding protein. *PLOS Computational Biology* **5(10)**:e1000544 DOI 10.1371/journal.pcbi.1000544.

Atilgan C, Gerek ZN, Ozkan SB, Atilgan AR. 2010. Manipulation of conformational change in proteins by single-residue perturbations. *Biophysical Journal* **99(3)**:933–943 DOI 10.1016/j.bpj.2010.05.020.

Echave J. 2008. Evolutionary divergence of protein structure: the linearly forced elastic network model. *Chemical Physics Letters* **457(4–6)**:413–416 DOI 10.1016/j.cplett.2008.04.042.

Echave J. 2012. Why are the low-energy protein normal modes evolutionarily conserved? *Pure and Applied Chemistry* **84(9)**:1931–1937 DOI 10.1351/PAC-CON-12-02-15.

Echave J, Fernández FM. 2010. A perturbative view of protein structural variation. *Proteins: Structure, Function, and Bioinformatics* **78(1)**:173–180 DOI 10.1002/prot.22553.

Fowler DM, Fields S. 2014. Deep mutational scanning: a new style of protein science. *Nature Methods* **11(8)**:801–807 DOI 10.1038/nmeth.3027.

General IJ, Liu Y, Blackburn ME, Mao W, Gierasch LM, Bahar I. 2014. ATPase subdomain IA is a mediator of interdomain allostery in Hsp70 molecular chaperones. *PLOS Computational Biology* **10(5)**:e1003624 DOI 10.1371/journal.pcbi.1003624.

Hamacher K. 2008. Relating sequence evolution of HIV1-protease to its underlying molecular mechanics. *Gene* **422(1–2)**:30–36 DOI 10.1016/j.gene.2008.06.007.

Ikeguchi M, Ueno J, Sato M, Kidera A. 2005. Protein structural change upon ligand binding: linear response theory. *Physical Review Letters* **94(7)**:1–4 DOI 10.1103/PhysRevLett.94.078102.

Jalalypour F, Sensoy O, Atilgan C, Atilgan C. 2020. Perturb-scan-pull: a novel method facilitating conformational transitions in proteins. *Journal of Chemical Theory and Computation* **16(6)**:3842–3855 DOI 10.1021/acs.jctc.9b01222.

Lake PT, Davidson RB, Klem H, Hocky GM, McCullagh M. 2020. Residue-level allostery propagates through the effective coarse-grained hessian. *Journal of Chemical Theory and Computation* **16(5)**:3385–3395 DOI 10.1021/acs.jctc.9b01149.

Livesey BJ, Marsh JA. 2020. Using deep mutational scanning to benchmark variant effect predictors and identify disease mutations. *Molecular Systems Biology* **16(7)**:1–12 DOI 10.15252/msb.20199380.

Marcos ML, Echave J. 2020. The variation among sites of protein structure divergence is shaped by mutation and scaled by selection. *Current Research in Structural Biology* **2(3)**:156–163 DOI 10.1016/j.crstbi.2020.08.002.

Micheletti C. 2013. Comparing proteins by their internal dynamics: exploring structure-function relationships beyond static structural alignments. *Physics of Life Reviews* **10(1)**:1–26 DOI 10.1016/j.plrev.2012.10.009.

Murzin AG, Brenner SE, Hubbard T, Chothia C. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology* **247**:536–540.

Nevin Gerek Z, Kumar S, Banu Ozkan S. 2013. Structural dynamics flexibility informs function and evolution at a proteome scale. *Evolutionary Applications* **6(3)**:423–433 DOI 10.1111/eva.12052.

Ponzoni L, Bahar I. 2018. Structural dynamics is a determinant of the functional significance of missense variants. *Proceedings of the National Academy of Sciences of the United States of America* **115(16)**:4164–4169 DOI 10.1073/pnas.1715896115.

Raimondi D, Orlando G, Tabaro F, Lenaerts T, Rooman M, Moreau Y, Vranken WF. 2018. Large-scale in-silico statistical mutagenesis analysis sheds light on the deleteriousness landscape of the human proteome. *Scientific Reports* **8(1)**:1–11 DOI 10.1038/s41598-018-34959-7.

Stebbings LA, Mizuguchi K. 2004. HOMSTRAD: recent developments of the homologous protein structure alignment database. *Nucleic Acids Research* **32(90001)**:203D–207 DOI 10.1093/nar/gkh027.

Tamura K, Hayashi S. 2015. Linear response path following: a molecular dynamics method to simulate global conformational changes of protein upon ligand binding. *Journal of Chemical Theory and Computation* **11(7)**:2900–2917 DOI 10.1021/acs.jctc.5b00120.

Tiberti M, Pandini A, Fraternali F, Fornili A. 2018. In silico identification of rescue sites by double force scanning. *Bioinformatics* **34(2)**:207–214 DOI 10.1093/bioinformatics/btx515.

Verkhivker GM. 2019. Biophysical simulations and structure-based modeling of residue interaction networks in the tumor suppressor proteins reveal functional role of cancer mutation hotspots in molecular communication. *Biochimica et Biophysica Acta (BBA)—General Subjects* **1863(1)**:210–225 DOI 10.1016/j.bbagen.2018.10.009.

Wingert B, Krieger J, Li H, Bahar I. 2021. Adaptability and specificity: how do proteins balance opposing needs to achieve function? *Current Opinion in Structural Biology* **67**:25–32 DOI 10.1016/j.sbi.2020.08.009.

Yilmaz LS, Atilgan AR. 2000. Identifying the adaptive mechanism in globular proteins: fluctuations in densely packed regions manipulate flexible parts. *The Journal of Chemical Physics* **113(10)**:4454–4464 DOI 10.1063/1.1288512.

Zhang PF, Su JG. 2019. Identification of key sites controlling protein functional motions by using elastic network model combined with internal coordinates. *The Journal of Chemical Physics* **151(4)**:045101 DOI 10.1063/1.5098542.

Zhang S, Li H, Krieger JM, Bahar I, Ozkan B. 2019. Shared signature dynamics tempered by local fluctuations enables fold adaptability and specificity. *Molecular Biology and Evolution* **36(9)**:2053–2068 DOI 10.1093/molbev/msz102.

Zheng W, Brooks BR. 2005. Normal-modes-based prediction of protein conformational changes guided by distance constraints. *Biophysical Journal* **88(5)**:3109–3117 DOI 10.1529/biophysj.104.058453.

Zheng W, Tekpinar M. 2009. Large-scale evaluation of dynamically important residues in proteins predicted by the perturbation analysis of a coarse-grained elastic model. *BMC Structural Biology* **9(1)**:45 DOI 10.1186/1472-6807-9-45.

Zheng W, Thirumalai D. 2009. Coupling between normal modes drives protein conformational dynamics: illustrations using allosteric transitions in myosin II. *Biophysical Journal* **96(6)**:2128–2137 DOI 10.1016/j.bpj.2008.12.3897.