Chapter 9

Applications of Normal Mode Analysis Methods in Computational Protein Design

Vincent Frappier, Matthieu Chartier, and Rafael Najmanovich

Abstract

Recent advances in coarse-grained normal mode analysis methods make possible the large-scale prediction of the effect of mutations on protein stability and dynamics as well as the generation of biologically relevant conformational ensembles. Given the interplay between flexibility and enzymatic activity, the combined analysis of stability and dynamics using the Elastic Network Contact Model (ENCoM) method has ample applications in protein engineering in industrial and medical applications such as in computational antibody design. Here, we present a detailed tutorial on how to perform such calculations using ENCoM.

Key words Normal mode analysis, Protein stability, Protein dynamics, Mutations, Vibrational entropy, Protein engineering

1 Introduction

Protein engineering aims at modulating the physico-chemical and biological properties of proteins through chemical modifications for industrial and medical applications. Such modifications include derivatizing surface residues and the introduction of mutations. Industrial applications often require mutations that confer increased efficiency in conditions drastically different than physiological as well as improved resistance to denaturation [1]. In a visionary article in 1983, Kevin Ulmer proposed that the integration of experimental approaches in protein chemistry, X-ray crystallography, and computer modeling held the key to understand and engineer protein structure and function [2]. Over 30 years later, much progress has been made but we are far from truly understanding protein function and structure to the point where we can engineer de novo functions. Traditionally, protein engineering involved structure-guided design through site-directed mutagenesis. While this approach is still used [3, 4], new methodologies such as directed evolution are commonly used today. Directed evolution is an experimental approach mimicking biological evolution where

Ilan Samish (ed.), *Computational Protein Design*, Methods in Molecular Biology, vol. 1529,
DOI 10.1007/978-1-4939-6637-0_9,

a large number of random mutants are produced and evolutionary pressure is applied in which successive rounds of selection are used to favor the emergence of desired phenotypes [5]. In that respect and depending on the goal, promiscuity in terms of binding or catalysis often simplifies the engineering task [6]. Otherwise, directed evolution can be sensitive to local minima of the fitness landscape [7, 8]. The late physicist Richard Feynman stated "what I cannot create, I do not understand." Directed evolution shows that it is possible to create new proteins without full understanding. However, in the spirit of Ulmer, the true potential of protein engineering will be achieved once we understand enough of the principles underlying protein structure and function to perform ab initio protein design.

Computational approaches have been used to identify mutations that change protein affinity [9], function [10], and stability [11]. However, most computational methods that focus on the impact of mutations on protein stability are biased toward predicting destabilizing mutations. This bias comes at times as an artifact of machine learning, but it can also be caused by the inherent difficulty of modeling stabilizing mutations. Therefore, most computational methods currently available fail to correctly predict stabilizing mutations [12, 13]. Another important point to consider is that changes in thermodynamic stability may have a detrimental effect on enzymatic activity [14–19]. A striking example comes from the comparison of mesophilic enzymes with their more stable thermophilic counterparts that exhibit lower enzyme efficiency at room temperatures [20]. This loss of efficiency is often associated with a rigidification of the structure [21, 22]. More generally, dynamics affects molecular recognition [9, 23–26] and catalytic rates [27, 28]. It is especially true for antibodies [29] where a rigidification of the complementarity determining region (CDR) is observed during the maturation process [30] and crucial to obtain high affinity specific molecules [31]. Allosteric mutations that improve binding affinity [32] in therapeutic antibodies highlight the importance of assessing the impact of mutations on protein dynamics. Finally, describing a protein as the conformational ensemble rather than a single structure has been shown to improve the prediction of the effect of mutations [33, 34] and improved the outcome of protein design protocols [35].

The evaluation of dynamic properties of proteins in a high-throughput context is not a trivial task. Experimental procedures (NMR or crystallographic b-factors) can be time-consuming and despite tremendous advances in molecular dynamic simulations, the ability to assess the effect of a mutation on dynamic properties of proteins is still computationally demanding, particularly for the long timescales associated with protein function [36]. Thus, evaluating several hundred mutants would seem unrealistic without specialized hardware. Normal mode analysis (NMA) provides an alternative.

It is a computational approach that predicts vibrational frequencies and movements of a system around an equilibrium state using a harmonic potential. The fundamentals of NMA have been extensively reviewed [37, 38] and classically is applied on all atoms of the structure with a molecular dynamics force field after initial minimization. Pioneering work by Tirion [39] showed that it is possible to reproduce the slow dynamics of proteins with a single-parameter potential by considering the structure as already in its equilibrium conformation and building a mass-spring system, removing the requirement for minimization. Tama et al. [40] showed that it is possible to replace all atoms of a residue by a single mass generally centered at the position of the alpha carbon, drastically reducing computational time. The speed of such coarse-grained NMA methods made possible their use in many applications to explore conformational space in small molecule docking [41, 42], to predict conformational changes [43] and in structural refinement [44, 45]. However, most coarse-grained methods do not account for the nature of amino acids by using spring constants that are independent of residue type. We recently introduced a coarse-grained NMA method called ENCoM [46], which uses a potential based on STeM [47] considering bond stretching, angle bending, dihedral rotation, and long-range interactions. Crucially, ENCoM adds an additional factor to the long-range interactions using the surface area in contact and the type of heavy atoms in contact. Thus, unlike other coarse-grained NMA methods, ENCoM calculations are affected by the specific amino acid nature of the protein in addition to its structure. Compared to the Anisotropic Network Model (ANM), one of the most used coarse-grained NMA methods [48], ENCoM shows an increased predictive power for conformational change between crystal structures of bound and unbound enzymes with an average increase in squared overlap of 28 % for 117 coupled movements and 60 % for 236 cases of coupled loop movements.

With ENCoM, we also introduced a novel application for coarse-grained NMA methods in the prediction of the effect of mutations on protein stability and dynamic properties. Predicted vibrational entropy differences (ΔS_{vib}) upon mutation were analyzed for 303 manually curated mutations [49] and compared to several existing methods, notably FoldX3.0 (beta 3.0) [50], Rosetta [51], DMutant [52], and PoPMusic [49]. Although not the overall best predictive method, ENCoM proved to be the most self-consistent and least biased. ENCoM and DMutant gave the best predictive power on the subset of 45 stabilizing mutations versus other methods that predicted as good or worse than a random model. Classic coarse-grained NMA models predicted every mutation as neutral and did not have any predictive power. The combination of ENCoM with enthalpy-based methods such as Rosetta and FoldX was synergistically beneficial [53]. As a proof of concept for the prediction of the effect of mutations on function,

ENCoM predicted the effect of the G121V mutation on *E. coli* DHFR consistent with S^2 differences NMR results [54]. Despite having a modest effect on protein stability (0.77 kcal/mol [55]) and being 15 Å away from the binding site, this mutation disrupts enzyme efficiency by 200-fold through allosteric effects. More recently, ENCoM was used to show that thermophile proteins are on average more rigid than their mesophile counterpart and used ΔS_{vib} to guide the selection of mutations observed between such proteins with potential uses in protein engineering [22].

In the following sections, we demonstrate how to use ENCoM to predict the effect of mutations on thermal stability and dynamics as well as to generate conformational ensembles (Fig. 1). The ability to perform large-scale combined predictions of the effect of mutations on stability and dynamics offers great possibilities in protein engineering. Likewise, the generation of biologically realistic conformational ensembles has ample applications in protein engineering and beyond.

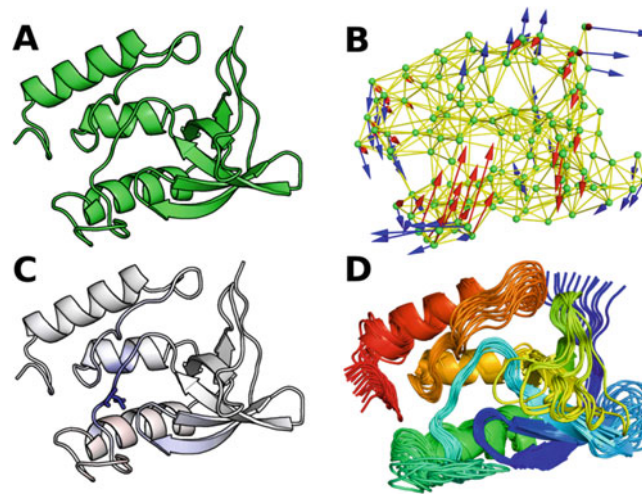

Fig. 1 Uses of ENCoM in protein engineering. The wild-type nuclease from Staphylococcus aureus (1EY0) used in the text is shown in (**a**). The protein structure is represented as an elastic network model using ENCoM algorithm (**b**), where amino acids are represented by masses (*green spheres*) and interactions by springs (*yellow sticks*). The Eigenvectors representing the seventh and tenth modes are shown in *red* and *blue* respectively. The mutation T41I (shown as stick in **c**) increases the thermal stability and rigidifies the protein in the regions identified in *blue* (**c**). A conformational ensemble of 11 conformations of the wild-type nuclease generated using the seventh and tenth modes are shown in (**d)**

2 Materials

For this tutorial it will be necessary to have some basic knowledge of command line environments and to install software (*see* **Note 1**). At the moment ENCoM does not work under the Windows operating system. Thus, for the tutorial below it is necessary to use a Unix-based operating system (Linux or Mac OS). Please make sure your system has up-to-date versions of Python and Perl.

The ENCoM Source code can be found at http://bcb.med.usherbrooke.ca/encom or through GitHub at https://github.com/NRGlab/ENCoM. Code can be compiled by the following instructions in the Readme file (*see* **Note 2**). ENCoM is used for the prediction of the effect of mutations and to generate conformational ensembles. Precompiled executables of FoldX3 can be found at: http://foldx.crg.es (*see* **Note 3**). FoldX3 is used exclusively for the prediction of the effect of mutations. Instructions to download and install Modeller can be found at https://salilab.org/modeller/download_installation.html. PyMOL is used for molecular visualizations. Instructions for installation on different operating systems can be found at http://www.pymolwiki.org/index.php/Category:Installation. Alternatively, the PyMOL source code can be found at: http://sourceforge.net/projects/pymol (*see* **Note 4**). All scripts required for the protocols used below can be found at http://bcb.med.usherbrooke.ca/encom.

3 Methods

The evaluation of the effect of mutations on protein thermodynamic stability is achieved by a linear combination of the predictions of ENCoM and FoldX. The prediction of the effect of mutations on protein dynamic on the other hand uses ENCoM exclusively. ENCoM is also used to generate ensembles of realistic protein conformations. The following protocols can be carried out in standard computers and do not require any specialized hardware. Execution times can vary from a few minutes to a few hours depending on the type of hardware used, the size of the protein, and the number of mutations to evaluate or conformations to generate. The entire protocol can also be automatically executed through the ENCoM Server [53] at http://bcb.med.usherbrooke.ca/encom. The advantage of running oneself the protocols is to overcome restrictions that are in place in the ENCoM Server such as the possibility to model and predict the effect of double (or more) mutants, the manner in which conformations are modeled using Modeller, and to explore combinations of modes that generate larger conformational ensembles than allowed in the webserver. Results obtained through the ENCoM Server interface can

serve to validate results obtained using the protocols below as the user learns how to use ENCoM.

We will be using the structure of the *Staphylococcus aureus* Thermonuclease (PDB ID 1EY0) as an example. However, any protein structure or model can be used (*see* **Note 5**).

During the protocol, we will be using software that can be installed in different directories depending on the computer. The FoldX3 installation folder will be referred to as *FoldX/*, ENCoM installation folder will be referred to as *ENCoM/*, and the perl and python scripts will be referred to as *script/*. The user should make sure to recognize what are the appropriate directories in their installation and replace the names accordingly. Text in italic following the > symbols represent command lines that are to be entered in a terminal.

3.1 Preparing Working Environment

In order to run ENCoM, is it better to create a work directory within which we will place the PDB formatted file containing the coordinates of the protein and prepare it:

1. Create a folder named *work* in which you will be working and change the working directory:

   ```
   > mkdir work
   > cd work
   ```

2. Download the 1EY0 structure from the PDB website using this address http://www.rcsb.org/pdb/files/1EY0.pdb and name it *1ey0_nc.pdb*; alternatively, use the command line below:

   ```
   > curl http://www.rcsb.org/pdb/files/1EY0.pdb > 1ey0_nc.pdb
   ```

3. Clean the PDB file by removing heteroatoms, water molecules, alternative conformations, and hydrogen atoms, changing negative residue numbers or residues with non-numeric characters, removing multiple models and adding a chain identifier Z if none is present using this command (*see* **Note 6**). The cleaned structure is now called *1ey0.pdb*.

   ```
   > perl script/clean_pdb.pl 1ey0_nc.pdb 1ey0.pdb
   ```

3.2 FoldX3 Thermal Stability Predictions

Thermal stability predictions involve a linear combination of FoldX3 predictions and ENCoM. ENCoM. As noted above, users must download FoldX3 and install it first. Once this is done follow the steps below:

1. In order to preprocess the protein structure we start with the following command

   ```
   > echo 1ey0.pdb > list.txt
   ```

2. Copy the *rotabase.txt* file found within the FoldX3 software into the working directory:

```
> cp FoldX/rotabase.txt ./
```

3. Launch FoldX3 repair function. This will generate a file named *RepairPDB_1ey0.pdb*.

```
> FoldX/foldx3b6 -runfile ./script/repair.txt
```

4. Write this filename in a list using

```
> echo RepairPDB_1ey0.pdb > list.txt
```

5. Open the file named *individual_list.txt* using any plain text editor (in the following command line we use nano) and write mutations that are to be evaluated using the following nomenclature: One letter code wild-type residue, chain, position in the structure sequence, and one letter code mutated residues, followed by a semicolon. For example, to mutate threonine 41 to an isoleucine in the 1EY0 structure, write *TA41I*;. For this protocol, please write in the *individual_list.txt* file on different lines the two following mutants: *TA41I;* and *DA21K;* (*see* **Note 7**).

```
> nano individual_list.txt
```

6. Launch the FoldX3 mutation function. The file *Dif_BuildModel_RepairPDB_1ey0.fxout* created in the working directory will have the difference in folding energy between WT and mutated forms (*see* **Note 8**).

```
> FoldX/foldx3b6 -runfile script/run.txt.
```

3.3 Effect of Mutations on Protein Stability and Dynamics

The ENCoM predictions can then be calculated as follows:

1. Generate the structure of the T41I and D21K mutants in chain A with the following command lines, where 1ey0 represents the filename, 41 or 21 the positions to mutate, ILE or LYS the new residues at these positions in chain A. The resulting modeled mutant structures will be in files *1ey0ILE41A.pdb* and *1ey0LYS21A.pdb*. In the command line below, the last two arguments represent the input PDB file containing the wild-type coordinates and the filename for the mutant coordinates respectively.

```
> python script/mutate_model.py 1ey0 41 ILE A 1ey0.pdb
1ey0ILE41A.pdb
> python script/mutate_model.py 1ey0 21 LYS A 1ey0.pdb
1ey0LYS21A.pdb
```

2. Calculate the normal modes and mode amplitudes for the wild-type and mutant structures generated in the previous step using the following command. The *.cov* files represent the entropy for each residue and the *.eigen* files contain the eigenvalues (mode frequencies) and eigenvectors (normal modes) of the different vibrational modes. These files will be used to compare dynamics between structures (*see* **Note 9**).

```
> ./ENCoM/bin/build_encom -i 1ey0.pdb -cov wt.cov -o
wt.eigen
> ./ENCoM/bin/build_encom -i 1ey0ILE41A.pdb -cov
TA41I.cov -o TA41I.eigen
> ./ENCoM/bin/build_encom -i 1ey0LYS21A.pdb -cov
DA21K.cov -o DA21K.eigen
```

3. The following command will use the files produced above to calculate the differences in dynamics between each mutant and the wild type, as well as the predicted $\Delta\Delta G$ for each mutation. The predicted $\Delta\Delta G$ is a linear combination of ENCoM and FoldX calculated earlier (*see* **Note 10**). The order of *.cov* files for the *-mutl* argument must be the same that the one in *individual_list.txt*.

```
> perl script/compare_cov.pl -FoldX Dif_BuildModel_-
RepairPDB_1ey0.fxout -wt wt.cov -mutl TA41I.cov DA21K.
cov.
```

4. The command script will generate a PyMOL session script called *Diff.pml* that colors every amino acid in function of ΔS for residue in each mutant, where blue represents a rigidification of the structure and red a gain in flexibility (*see* **Note 11**). It can be viewed using:

```
> pymol Diff.pml
```

3.4 Generation of Conformational Ensembles

In addition to the prediction of the effect of mutations on stability and dynamics, ENCoM can be used to generate conformational ensembles:

1. The following script generates multiple conformations using ENCoM. In the case below, we are using the wild type and use the eigenvectors previously calculated in Subheading 3.3, **step 2** (file *wt.eigen*). The same could be done for a mutant, using the appropriate mutant structure and calculated eigenvectors. The file *all_conformations.pdb* contains all the exhaustively generated models using the 10th and the 12th slowest vibrational modes (parameter *–ml*) with a maximum RMSD distortion of 2 Å (parameter *–md)* and a minimum RMSD distortion of 1 Å (parameter *–step*) per mode. Remember that the first six modes represent rotations and translations; thus, the smallest value for any argument passed via *–ml* should be 7, representing the slowest, most global mode of movement.

```
> ENCoM/bin/build_grid_rmsd -i 1ey0.pdb -ieig wt.eigen
-md 2 -step 1 -p all_conformations.pdb -ml 10 12
```

2. Each individual mode can be viewed using the motion function. For example, the mode 10 can be given by

```
> ENCoM/bin/motion -i 1ey0.pdb -m 10 -ieig wt.eigen -
p motion_10.pdb
```

3. Cartesian space NMA methods such as ENCoM generated conformations that are linear combinations of movements (translations of atomic coordinates) along different modes. Thus, the structures generated do not respect bond angles and distances. Conformations represent distorted physically unrealistic structures. Modeller is used to rebuild physically realistic structures using each distorted NMA structure as a template. The rebuilt model will be found in the folder called *models*. This is done with the command below.

```
> perl script/rebuild.pl -i all_conformations.pdb -
script script/rebuild.py
```

4 Notes

1. All software employed in the protocols are free at least for nonprofit users. ENCoM is free for everyone and distributed under the GNU General Public License.
2. Users need to have the GNU GSL library installed, more information can be found at http://www.gnu.org/software/gsl/.
3. FoldX is developed and maintained by the research group of Dr. Luis Serrano at the GRG. Users need to make an account and accept a yearly-renewable Licence. FoldX needs to be downloaded anew every year to work with the newly renewed license.
4. Homebrew installation is recommended for Mac OS, particularly for Mac OS 10.10 Yosemite. Binary distributions are recommended for Linux.
5. Experimentally determined protein structures can be found on the PDB depository (http://www.rcsb.org/). If the desired structure is not available, servers such as I-Tasser or Robetta can be used to generate homology models. It is important to note that PDB X-ray structures represent the asymmetric unit that may or may not correspond to the biological unit (quaternary structure). Users can download experimentally verified or predicted biological units from any of the PDB depositories.
6. Alternatively, you can manually curate your PDB file by analyzing the structure in PyMOL, making modifications and saving the modified structure or by editing the file directly in a text editor.
7. Multiple mutations can be specified by separating them with a comma in the same line, i.e., *TA41I,DA21K*; will evaluate a double mutant whereas if these two mutations appear in individual lines, two single mutants will be predicted.

8. This is a relative score representing the ΔΔG of folding; negative values are associated with stabilizing mutations.
9. The first six modes are rotation and translation modes. They should not be considered.
10. Energy is calculated as previously done [22, 46, 53] with higher values corresponding to more rigid structures.
11. The colors are scaled by the maximum absolute difference or three times the standard deviation, whichever is smaller.

Acknowledgments

R.J.N. is part of PROTEO (the Québec network for research on protein function, structure and engineering), and GRASP (Groupe de Recherche Axé sur la Structure des Protéines). The authors would like to thank Dr. Luis Serrano for giving his permission to use FoldX within the ENCoM server.

Funding: V.F. is the recipient of a Ph.D. fellowship from the Fonds de Recherche du Québec—Nature et Technologies (FRQ-NT); M.C. is the recipient of a Ph.D. fellowship from the Natural Sciences and Engineering Research Council of Canada (NSERC). NSERC Discovery Grant RGPIN-2014-05766.

References

1. Bommarius AS, Blum JK, Abrahamson MJ (2011) Status of protein engineering for biocatalysts: how to design an industrially useful biocatalyst. Curr Opin Chem Biol 15:194–200
2. Ulmer KM (1983) Protein engineering. Science 219:666–671
3. Ott K-H, Kwagh J-G, Stockton GW, Sidorov V, Kakefuda G (1996) Rational molecular design and genetic engineering of herbicide resistant crops by structure modeling and site-directed mutagenesis of acetohydroxyacid synthase. J Mol Biol 263:359–368
4. Diskin R, Scheid JF, Marcovecchio PM, West AP, Klein F, Gao H, Gnanapragasam PNP, Abadir A, Seaman MS, Nussenzweig MC, Bjorkman PJ (2011) Increasing the potency and breadth of an HIV antibody by using structure-based rational design. Science 334:1289–1293
5. Socha RD, Tokuriki N (2013) Modulating protein stability – directed evolution strategies for improved protein function. FEBS J 280:5582–5595
6. Khersonsky O, Roodveldt C, Tawfik D (2006) Enzyme promiscuity: evolutionary and mechanistic aspects. Curr Opin Chem Biol 10:498–508
7. Carlson JC, Badran AH, Guggiana-Nilo DA, Liu DR (2014) Negative selection and stringency modulation in phage-assisted continuous evolution. Nat Chem Biol 10:216–222
8. Dickinson BC, Leconte AM, Allen B, Esvelt KM, Liu DR (2013) Experimental interrogation of the path dependence and stochasticity of protein evolution using phage-assisted continuous evolution. Proc Natl Acad Sci U S A 110:9007–9012
9. Tinberg CE, Khare SD, Dou J, Doyle L, Nelson JW, Schena A, Jankowski W, Kalodimos CG, Johnsson K, Stoddard BL, Baker D (2013) Computational design of ligand-binding proteins with high affinity and selectivity. Nature 501:212–216
10. Jiang L, Althoff EA, Clemente FR, Doyle L, Röthlisberger D, Zanghellini A, Gallaher JL, Betker JL, Tanaka F, Barbas CF, Hilvert D, Houk KN, Stoddard BL, Baker D (2008) De novo computational design of retro-aldol enzymes. Science 319:1387–1391

11. Zhang S-B, Wu Z-L (2011) Identification of amino acid residues responsible for increased thermostability of feruloyl esterase A from Aspergillus niger using the PoPMuSiC algorithm. Bioresour Technol 102:2093–2096
12. Thiltgen G, Goldstein RA (2012) Assessing predictors of changes in protein stability upon mutation using self-consistency. PLoS One 7, e46084
13. Kepp KP (2014) Computing stability effects of mutations in human superoxide dismutase 1. J Phys Chem B 118:1799–1812
14. Teilum K, Olsen JG, Kragelund BB (2011) Protein stability, flexibility and function. Biochim Biophys Acta 1818:969–976
15. Poelwijk FJ, Kiviet DJ, Weinreich DM, Tans SJ (2007) Empirical fitness landscapes reveal accessible evolutionary paths. Nature 445:383–386
16. van den Burg B, Eijsink VGH (2002) Selection of mutations for increased protein stability. Curr Opin Biotechnol 13:333–337
17. Bloom JD, Meyer MM, Meinhold P, Otey CR, MacMillan D, Arnold FH (2005) Evolving strategies for enzyme engineering. Curr Opin Struct Biol 15:447–452
18. Shoichet BK, Baase WA, Kuroki R, Matthews BW (1995) A relationship between protein stability and protein function. Proc Natl Acad Sci U S A 92:452–456
19. Giver L, Gershenson A, Freskgard PO, Arnold FH (1998) Directed evolution of a thermostable esterase. Proc Natl Acad Sci U S A 95:12809–12813
20. Ruller R, Deliberto L, Ferreira TL, Ward RJ (2007) Thermostable variants of the recombinant xylanase a from Bacillus subtilis produced by directed evolution show reduced heat capacity changes. Proteins 70:1280–1293
21. Wolf-Watz M, Thai V, Henzler-Wildman K, Hadjipavlou G, Eisenmesser EZ, Kern D (2004) Linkage between dynamics and catalysis in a thermophilic-mesophilic enzyme pair. Nat Struct Mol Biol 11:945–949
22. Frappier V, Najmanovich RJ (2015) Vibrational entropy differences between mesophile and thermophile proteins and their use in protein engineering. Protein Sci 24:474–483
23. Jiménez-Osés G, Osuna S, Gao X, Sawaya MR, Gilson L, Collier SJ, Huisman GW, Yeates TO, Tang Y, Houk KN (2014) The role of distant mutations and allosteric regulation on LovD active site dynamics. Nat Chem Biol 10:431–436
24. Motlagh HN, Wrabl JO, Li J, Hilser VJ (2014) The ensemble nature of allostery. Nature 508:331–339
25. Gaudreault F, Chartier M, Najmanovich RJ (2012) Side-chain rotamer changes upon ligand binding: common, crucial, correlate with entropy and rearrange hydrogen bonding. Bioinformatics 28:i423–i430
26. Gaudreault F, Najmanovich RJ (2015) FlexAID: revisiting docking on non-native-complex structures. J Chem Inf Model
27. van den Bedem H, Bhabha G, Yang K, Wright PE, Fraser JS (2013) Automated identification of functional dynamic contact networks from X-ray crystallography. Nat Methods 10:896–902
28. Doucet N (2011) Can enzyme engineering benefit from the modulation of protein motions? Lessons learned from NMR relaxation dispersion experiments. Protein Pept Lett 18:336–343
29. Elvin JG, Couston RG, van der Walle CF (2013) Therapeutic antibodies: market considerations, disease targets and bioprocessing. Int J Pharm 440:83–98
30. Zimmermann J, Zimmermann J, Oakman EL, Oakman EL, Thorpe IF, Thorpe IF, Shi X, Shi X, Abbyad P, Abbyad P, Brooks CL, Brooks CL, Boxer SG, Boxer SG, Romesberg FE, Romesberg FE (2006) Antibody evolution constrains conformational heterogeneity by tailoring protein dynamics. Proc Natl Acad Sci U S A 103:13722–13727
31. Thielges MC, Zimmermann J, Yu W, Oda M, Romesberg FE (2008) Exploring the energy landscape of antibody–antigen complexes: protein dynamics, flexibility, and molecular recognition. Biochemistry 47:7237–7247
32. Boder ET, Midelfort KS, Wittrup KD (2000) Directed evolution of antibody fragments with monovalent femtomolar antigen-binding affinity. Proc Natl Acad Sci U S A 97:10701–10705
33. Smith CA, Kortemme T (2008) Backrub-like backbone simulation recapitulates natural protein conformational variability and improves mutant side-chain prediction. J Mol Biol 380:742–756
34. Kellogg EH, Leaver-Fay A, Baker D (2010) Role of conformational sampling in computing mutation-induced changes in protein structure and stability. Proteins 79:830–838
35. Davey JA, Chica RA (2013) Improving the accuracy of protein stability predictions with multistate design using a variety of backbone ensembles. Proteins 82:771–784
36. LeVine MV, Weinstein H (2014) NbIT – a new information theory-based analysis of allosteric mechanisms reveals residues that underlie function in the leucine transporter LeuT. PLoS Comput Biol 10, e1003603

37. Mahajan S, Sanejouand Y-H (2015) On the relationship between low-frequency normal modes and the large-scale conformational changes of proteins. Arch Biochem Biophys 567:59–65
38. Fuglebakk E, Tiwari SP, Reuter N (2015) Comparing the intrinsic dynamics of multiple protein structures using elastic network models. Biochim Biophys Acta 1850:911–922
39. Tirion M (1996) Large amplitude elastic motions in proteins from a single-parameter. Atom Anal Phys Rev Lett 77:1905–1908
40. Tama F, Gadea FX, Marques O, Sanejouand YH (2000) Building-block approach for determining low-frequency normal modes of macromolecules. Proteins 41:1–7
41. Abagyan R, Rueda M, Bottegoni G (2009) Consistent improvement of cross-docking results using binding site ensembles generated with elastic network normal modes. J Chem Inf Model 49:716–725
42. Park S-J, Kufareva I, Abagyan R (2010) Improved docking, screening and selectivity prediction for small molecule nuclear receptor modulators using conformational ensembles. J Comput Aided Mol Des 24:459–471
43. Alexandrov V, Lehnert U, Echols N, Milburn D, Engelman D, Gerstein M (2005) Normal modes for predicting protein motions: a comprehensive database assessment and associated Web tool. Protein Sci 14:633–643
44. Schröder GF, Brunger AT, Levitt M (2007) Combining efficient conformational sampling with a deformable elastic network model facilitates structure refinement at low resolution. Structure 15:1630–1641
45. Tama F, Valle M, Frank J, Brooks C (2003) Dynamic reorganization of the functionally active ribosome explored by normal mode analysis and cryo-electron microscopy. Proc Natl Acad Sci U S A 100:9319–9323
46. Frappier V, Najmanovich RJ (2014) A coarse-grained elastic network atom contact model and its use in the simulation of protein dynamics and the prediction of the effect of mutations. PLoS Comput Biol 10:e1003569
47. Lin T-L, Song G (2010) Generalized spring tensor models for protein fluctuation dynamics and conformation changes. BMC Struct Biol 10(Suppl 1):3
48. Atilgan AR, Durell SR, Jernigan RL, Demirel MC, Keskin O, Bahar I (2001) Anisotropy of fluctuation dynamics of proteins with an elastic network model. Biophys J 80:505–515
49. Dehouck Y, Grosfils A, Folch B, Gilis D, Bogaerts P, Rooman M (2009) Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: PoPMuSiC-2.0. Bioinformatics 25:2537–2543
50. Schymkowitz J, Borg J, Stricher F, Nys R, Rousseau F, Serrano L (2005) The FoldX web server: an online force field. Nucleic Acids Res 33:W382–W388
51. Leaver-Fay A, Tyka M, Lewis SM, Lange OF, Thompson J, Jacak R, Kaufman K, Renfrew PD, Smith CA, Sheffler W, Davis IW, Cooper S, Treuille A, Mandell DJ, Richter F, Ban Y-EA, Fleishman SJ, Corn JE, Kim DE, Lyskov S, Berrondo M, Mentzer S, Popović Z, Havranek JJ, Karanicolas J, Das R, Meiler J, Kortemme T, Gray JJ, Kuhlman B, Baker D, Bradley P (2011) ROSETTA3: an object-oriented software suite for the simulation and design of macromolecules. Methods Enzymol 487:545–574
52. Zhou H, Zhou Y (2002) Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. Protein Sci 11:2714–2726
53. Frappier V, Chartier M, Najmanovich RJ (2015) ENCoM server: exploring protein conformational space and the effect of mutations on protein function and stability. Nucleic Acids Res
54. Boehr DD, Schnell JR, McElheny D, Bae S-H, Duggan BM, Benkovic SJ, Dyson HJ, Wright PE (2013) A distal mutation perturbs dynamic amino acid networks in dihydrofolate reductase. Biochemistry 52:4605–4619
55. Gekko K, Yamagami K, Kunori Y, Ichihara S, Kodama M, Iwakura M (1993) Effects of point mutation in a flexible loop on the stability and enzymatic function of Escherichia coli dihydrofolate reductase. J Biochem 113:74–80