A beginner's guide to molecular dynamics simulations and the identification of cross-correlation networks for enzyme engineering

Haoran Yu, Paul A. Dalby*

Department of Biochemical Engineering, University College London, London, United Kingdom
*Corresponding author: e-mail address: p.dalby@ucl.ac.uk

Contents

Abstract

The functional properties of proteins are decided not only by their relatively rigid overall structures, but even more importantly, by their dynamic properties. In a protein, some regions of structure exhibit highly correlated or anti-correlated motions with others, some are highly dynamic but uncorrelated, while other regions are relatively static. The residues with correlated or anti-correlated motions can form a so-called dynamic

Methods in Enzymology
ISSN 0076-6879
https://doi.org/10.1016/bs.mie.2020.04.020

© 2020 Elsevier Inc.
All rights reserved.

cross-correlation network, through which information can be transmitted. Such networks have been shown to be critical to allosteric transitions, and ligand binding, and have also been shown to be able to mediate epistatic interactions between mutations. As a result, they are likely to play a significant role in the development of new enzyme engineering strategies. In this chapter, protocols are provided for the assessment of dynamic cross-correlation networks, and for their application in protein engineering. Transketolase from *E. coli* is used as a model and the software GROMACS is applied for carrying out MD simulations to generate trajectories containing structural ensembles. The trajectory is then used for a dynamic cross correlation analysis using the R package, Bio3D. A matrix of all atom-wise cross-correlation coefficients is finally obtained, which can be displayed in a graphical representation termed a dynamical cross-correlation matrix.

1. Introduction

It has long been understood that proteins are densely folded and also dynamic (DuBay, Bowman, & Geissler, 2015) with motions that can range from small-scale atomic fluctuations around an average structure, to large-scale reorganizations of the molecular topology (Henzler-Wildman & Kern, 2007). The functional properties of proteins are also determined not only by their relatively rigid structures, but even more importantly, by their dynamic properties (Yang et al., 2014). These functions may include regulation of the passage of ions across the cell membrane, chaperoning of protein folding, transduction of signals, transcription and translation of DNA, catalysis of chemical reactions, and many other important cellular functions (Mazal & Haran, 2019).

Dynamics of proteins are characterized not only by the timescale of the fluctuations but also by the amplitude and directionality of the fluctuations (Yang et al., 2014). Inspection of the conformational ensemble arising due to the dynamic nature of proteins, has indicated that some regions exhibit highly correlated or anti-correlated motions, reflecting motions in the same or opposite direction, respectively. These correlated residues can form a so-called dynamic cross-correlation network, through which information can be transmitted to link, for example, the binding of a molecule at one site on the protein, to a change in local structure elsewhere in the protein (Doshi, Holliday, Eisenmesser, & Hamelberg, 2016). In this way, dynamic cross-correlation networks have been shown to be critical in mediating allosteric transitions (Clarkson, Gilmore, Edgell, & Lee, 2006; Selvaratnam, Chowdhury, VanSchouwen, & Melacini, 2011) and ligand binding (Zhuravleva et al., 2007).

Mutations located away from the active sites of enzymes have often been found to drastically alter the catalytic activity of enzymes, including for RNase A, GH1 beta-glucosidase, HIV-1 protease, phosphoglucose isomerase, cytochrome P450 and amide hydrolase (Gagne et al., 2015; Sharir-Ivry & Xia, 2018; Souza, Ikegami, Arantes, & Marana, 2018; Subramanian et al., 2019; Wilding, Hong, Spence, Buckle, & Jackson, 2019; Wrenbeck, Azouz, & Whitehead, 2017; Zhang, Liu, Lewis, & Wei, 2011). Several studies have indicated that these mutations are involved in dynamically correlated motions with their respective active sites (Estabrook et al., 2005; Gagne et al., 2015; Tyukhtenko et al., 2018). The effect of a single-site amino acid mutation on protein structure and function is also often dependent on the presence of other mutations either in structurally neighboring sites, or also at more distant sites (Reetz, 2013). Such effects can therefore lead to epistasis between mutations. For example, a mutation is neutral to the protein if it has little effect on the protein fitness. However, the same mutation in conjunction with mutations at other sites could have a positive or a negative effect on the fitness landscape (Naganathan, 2019). We previously found that long-range epistasis could also be mediated through dynamics cross-correlation networks (Yu & Dalby, 2018a, 2018b), and so could also be useful for developing enzyme engineering strategies.

Transketolase (TK), a thiamine diphosphate-dependent (ThDP) enzyme, catalyzes the reversible transfer of a C2-ketol unit from D-xylulose-5-phosphate to either D-ribose-5-phosphate or D-erythrose-4-phosphate in living cells (Draths et al., 1992; Sprenger, Schorken, Sprenger, & Sahm, 1995). TK is a homodimer of two 70- to 74-kDa monomers, each composed of a pyrophosphate (PP)-binding domain, pyrimidine (Pyr)-binding domain, and a C-terminal domain. Using *E. coli* TK as a model, we explored the epistatic interactions between a set of stabilizing mutations from across two different domains within the protein structure, and found that not all pairwise effects between distant mutations were additive (Yu & Dalby, 2018a). Molecular-dynamics simulations and a pairwise cross-correlation analysis revealed that mutations influenced the dynamics of their local environment, but also in some cases the dynamics of regions much more distant in the structure. This effect was found to mediate epistatic interactions between distant mutations (Yu & Dalby, 2018a). Based on this mechanism, we proposed that the regions outside the active site, whose dynamics were highly correlated to flexible active sites, could be used as mutation targets to stabilize the protein (Yu & Dalby, 2018b). This strategy was then successfully applied to counteract the activity–stability trade-off in a transketolase variant that had been previously engineered to accept a new substrate (Yu & Dalby, 2018b).

ARTICLE IN PRESS

Enzyme active-sites have been used as mutation targets for engineering enzyme functions including their activity, substrate scope, stereospecificity, regioselectivity, and enantioselectivity. Therefore, dynamic cross-correlation networks, in which some of the correlated sites fall within the active site, offer potential targets for selectively modifying these enzyme properties in future protein engineering strategies.

The dynamic correlation between residues can be determined through molecular dynamics (MD) simulations (Herzberg & Moult, 1991), or using NMR perturbation methods (Palmer, 2014). NMR studies provide insight into the conformational dynamics of proteins over a wide-range of time-scales. By introducing perturbations such as ligand binding or amino-acid substitutions, and analyzing correlated changes in NMR parameters such as chemical shift, the potential networks of correlated residues can be determined (O'Rourke, Gorman, & Boehr, 2016). MD simulation is an approach that is also widely used for investigating the dynamics of proteins. Simulations have become more accurate with improved force fields, and also better representation of the solvent. Using MD techniques, researchers can obtain not only the ultimate details concerning individual particle motions as a function of time, but also the global molecular motions of proteins that are difficult to access experimentally. An MD simulation trajectory contains a description of the dynamical motions of all atoms, and the extent of the dynamic correlation can be quantified by calculating the covariance between the fluctuations of two atoms (Estabrook et al., 2005; Ichiye & Karplus, 1991). The covariance $c(i,j)$ is calculated by:

$$c(i,j) = < \Delta \mathbf{r}_i \cdot \Delta \mathbf{r}_j >$$

in which $\Delta \mathbf{r}_i$ is the displacement vector of atom i and the angle brackets denote an ensemble average. The cross-correlation coefficient, or normalized covariance, is calculated by:

$$C(i,j) = \frac{c(i,j)}{[c(i,i)c(i,j)]^{1/2}} = \frac{< \Delta \mathbf{r}_i \cdot \Delta \mathbf{r}_j >}{< \Delta \mathbf{r}_i{}^2 >^{1/2} < \Delta \mathbf{r}_j{}^2 >^{1/2}}$$

The positive value of $C(i,j)$ implies positively correlated movement whereby the two atoms moved in the same direction, whereas negative value indicates anti-correlated movements whereby the atoms moved in the opposite direction. Completely correlated motions, $C(i,j) = 1$, or completely anticorrelated motions, $C(i,j) = -1$, means the motions have the same phase and period. If two atoms have fluctuations of the same period

ARTICLE IN PRESS

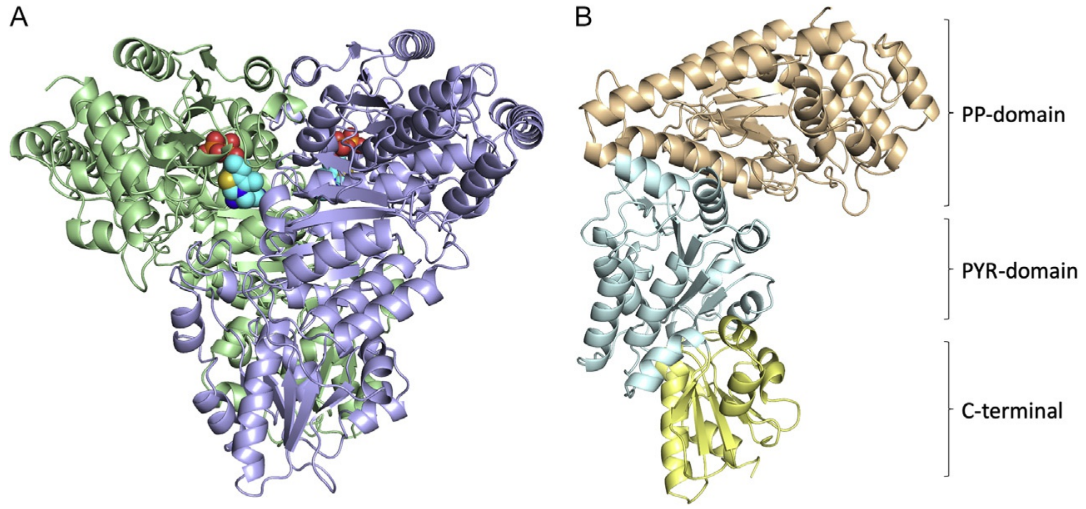

Fig. 1 Crystal structure of *E. coli* transketolase (PDB 1QGD). (A) TK structure showing the dimer. Two chains are shown by different colors (green/blue). ThDP is shown in the style of spheres. (B) TK structure showing only the monomer. The three different domains, PP-domain, PYR-domain and C-terminal, are colored orange, cyan, and yellow, respectively.

and phase but the displacements are oriented at an angle of 90°, then $C(i,j) = 0$. A contour plot of the matrix $C(i,j)$ constitutes a dynamical cross-correlation matrix, in which atomic motions with strong correlations will have correspondingly large off-diagonal cross peaks (Swaminathan, Harte, & Beveridge, 1991).

In this chapter, we provide the protocols as we have used them, to assess the dynamic cross-correlation network (Fig. 2), and to inform potential protein engineering strategies. TK from *E. coli* is used as a model and its crystal structure 1QGD.pdb will be used as the input (Fig. 1) (Littlechild et al., 1995). The software GROMACS (Lindahl, Hess, & van der Spoel, 2001) will be used to perform an MD simulation of TK and to generate a trajectory containing a large number of structural ensembles. The trajectory is then used to perform a dynamic cross-correlation analysis using the R package, Bio3D (Grant, Rodrigues, ElSawy, McCammon, & Caves, 2006). This package will return a matrix of all atom-wise cross-correlations whose elements, C_{ij}, can be displayed in a graphical representation termed a dynamical cross-correlation matrix (DCCM) (Fig. 2).

2. Molecular dynamics simulations

MD simulations use principles of classical mechanics to predict the motion of particle-based systems. MD simulations can be carried out in

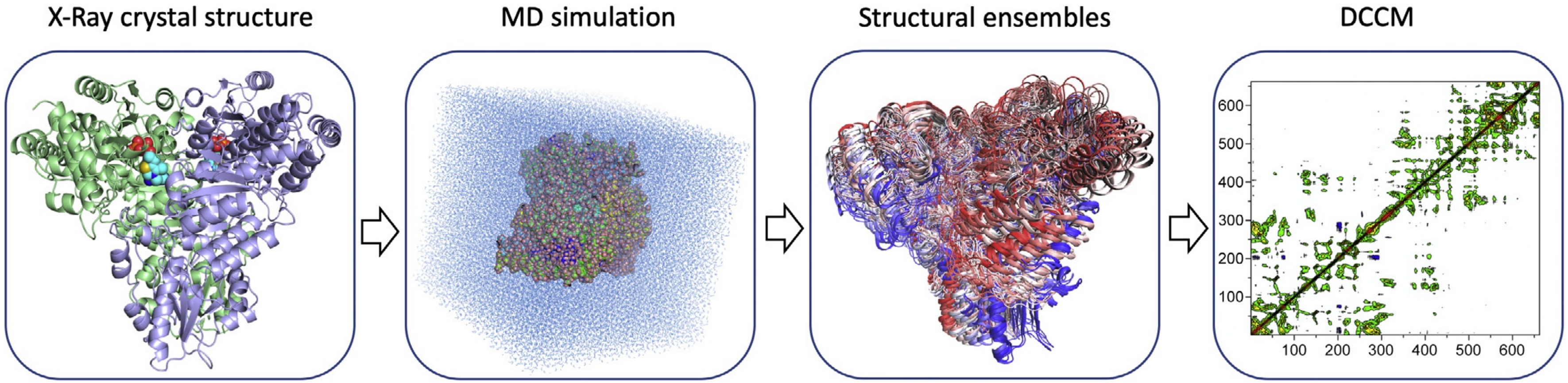

Fig. 2 Overview of methodology for calculating dynamic cross-correlation map.

ARTICLE IN PRESS

software packages such as GROMACS (Lindahl et al., 2001), AMBER (Case et al., 2005), NAMD (Phillips et al., 2005), CHARMM (Brooks et al., 2009), LAMMPS (Plimpton, 1995) and DESMON (Bowers et al., 2006). GROMACS is a fast, free, popular and user-friendly package to perform molecular dynamics. It has been continually upgraded and maintained since its initial release in 1991. There are detailed documents and tutorials for assisting learning to use GROMACS in the website http://www.gromacs.org/. Here, the GROMACS V5.0 is used to investigate the dynamics of transketolase at the temperature of 300 K.

2.1 Preparation of the system

2.1.1 Obtaining a PDB structure

The basic ingredient to start MD simulations is a protein structure coordinate file in PDB format that can be downloaded from the RCSB website (https://www.rcsb.org/). When the protein structure of interest is not available, a homology model as an initial starting structure may be built by a variety of software tools including Modeller (Sali & Blundell, 1993), I-Tasser (Roy, Kucukural, & Zhang, 2010), Discovery Studio (Dassault Systèmes BIOVIA, 2016), or the Pepbuild web server (Singh, 2004). The Protein Databank may also contain several structures for the same protein. In this case, all potential structures should be checked for quality and completeness to select the most suitable one for MD simulation. Once a PDB structure is downloaded, it could be checked and re-formatted using a structure viewing program such as VMD, Chimera, or PyMOL, and plain text editors such as nano, vi, emacs, Sublime Text2 (Linux/Mac) or Notepad (Windows). The PDB structures should be checked for whether the structure resolution is high enough (Note 1), whether there are any missing atoms or residues in the structure (Note 2), and whether the ligands or water molecules are present (Note 3). We used the 3-D structure with PDB code 1QGD for the MD simulation of TK from *E. coli* (Fig. 1). TK is a homodimer and each subunit is 70–74 kDa. The cofactor thiamine diphosphate molecule binds at the interface between the two subunits, which makes each monomer of TK consist of three domains, the PP-binding domain, the Pyr-binding domain and the C-terminal domain (Fig. 1). PDB structure (1QGD) has a resolution of 1.9 Å and no missing atoms/residues are found in this structure. The ligands and crystal waters are removed before starting the MD simulations.

2.1.2 Creating a topology from the PDB file

After getting the input PDB structure ready, we can execute the first GROMACS module, **pdb2gmx** to convert the PDB file into a GROMACS specific molecular geometry file format (.gro), the topology file (.top) and position restraint file (.itp). For the simulation of TK, we worked with the OPLS-AA force field (Note 4), the SPC/E water model (Note 5), and accepted the default choices for all residue protonation states (Note 6), termini, disulfide bridges, etc. The command to use is then:

```
gmx pdb2gmx -f 1QGD.pdb -o 1QGD.gro -water spce
```

pdb2gmx will process the PDB structure and prompt the user to select a force field:

```
    Select the Force Field:
    From '/usr/local/gromacs/share/gromacs/top':
     1: AMBER03 protein, nucleic AMBER94 (Duan et al., J. Comp. Chem.
24, 1999-2012, 2003)
     2: AMBER94 force field (Cornell et al., JACS 117, 5179-5197, 1995)
     3: AMBER96 protein, nucleic AMBER94 (Kollman et al., Acc. Chem.
Res. 29, 461-469, 1996)
     4: AMBER99 protein, nucleic AMBER94 (Wang et al., J. Comp. Chem.
21, 1049-1074, 2000)
     5: AMBER99SB protein, nucleic AMBER94 (Hornak et al., Proteins 65,
712-725, 2006)
     6: AMBER99SB-ILDN protein, nucleic AMBER94 (Lindorff-Larsen
et al., Proteins 78, 1950-58, 2010)
     7: AMBERGS force field (Garcia & Sanbonmatsu, PNAS 99, 2782-2787, 2002)
     8: CHARMM27 all-atom force field (CHARM22 plus CMAP for proteins)
     9: GROMOS96 43a1 force field
    10: GROMOS96 43a2 force field (improved alkane dihedrals)
    11: GROMOS96 45a3 force field (Schuler JCC 2001 22 1205)
    12: GROMOS96 53a5 force field (JCC 2004 vol 25 pag 1656)
    13: GROMOS96 53a6 force field (JCC 2004 vol 25 pag 1656)
    14: GROMOS96 54a7 force field (Eur. Biophys. J. (2011), 40, 843-856,
DOI: 10.1007/s00249-011-0700-9)
    15: OPLS-AA/L all-atom force field (2001 aminoacid dihedrals)
```

Since we use the all-atom OPLS-AA force field, type 15 at the command prompt, followed by "Enter." This command will produce three files: 1QGD.gro, topol.top and posre.itp. 1QGD.gro is a GROMACS-formatted

ARTICLE IN PRESS

structure file that contains the coordinates of all the atoms defined within the force field (Note 7). The missing hydrogen atoms are added by the **pdb2gmx** by default. The topol.top file is the system topology that contains the molecular description such as molecular parameters, bonding, force field and charges. The posre.itp file contains information used to restrain the positions of heavy atoms, which will be used in the step of position-restraint equilibration.

2.1.3 Creating a simulation box

In a box with rigid walls, the atoms on the protein surface are closer to the box walls and will experience different forces than the other atoms. Therefore, periodic boundary conditions (PBC) are applied to avoid this edge effect on the surface atoms. The existence of PBC means that any atom that leaves a simulation box through one boundary of the cell, will reappear on the opposite side. For this purpose, a box (unit cell) is defined, and that unit cell is surrounded by infinitely replicated, periodic images of itself. The module to define the box dimensions is **editconf** and the command to generate a box is then:

```
gmx editconf -f 1QGD.gro -o 1QGD_box.gro -c -d 1.0 -bt cubic
```

The above command centers the protein in the box (**-c**), and places it at least 1.0 nm from the box edge (**-d 1.0**) (Note 8). The box type is defined as a cube (**-bt cubic**) (Note 9). The new conformation is written to the file **1QGD_box.gro**.

2.1.4 Adding solvent water

To mimic the physiological environment for a protein, the box around the protein needs to be solvated. This is performed by using a small pre-equilibrated system of water coordinates that is repeated over the box, with overlapping water molecules removed. To solvate the protein, the **solvate** module will be used, which adds the required number of water molecules around the protein based upon the box type that is specified in the **editconf** step.

```
gmx solvate -cp 1QGD_box.gro -cs spc216.gro -o 1QGD_solv.gro -p topol.top
```

The **solvate** module solvates a protein configuration in a bath of solvent molecules. The box specified in the protein coordinate file (**-cp**) is used,

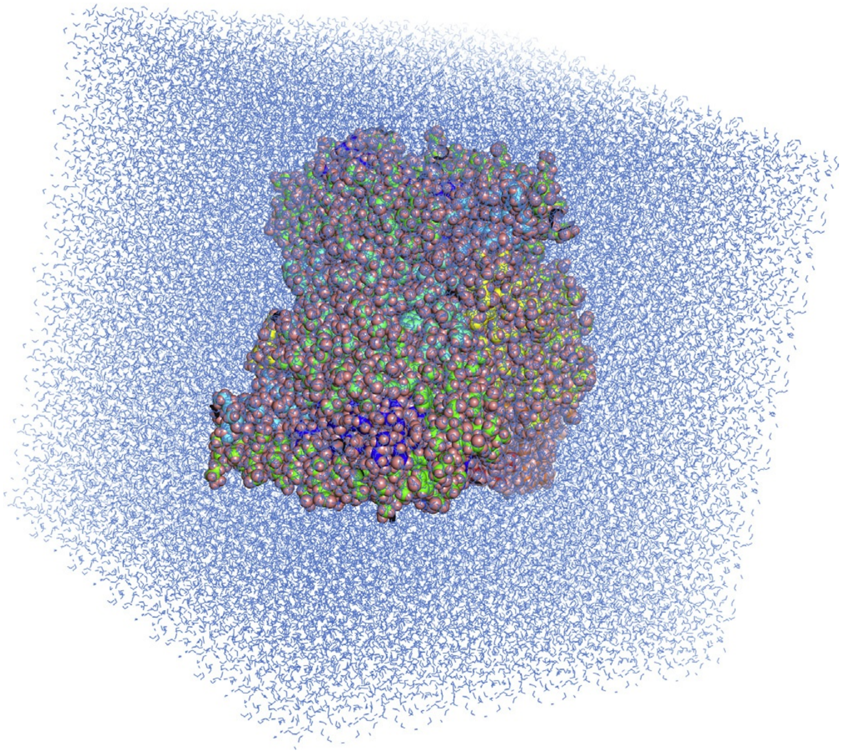

Fig. 3 Simulation box solvated with water. The TK structure is in a cubic simulation box of dimensions 12.4 × 12.4 × 12.4 nm, solvated with water.

which is the output of the previous **editconf** step, and the configuration of the solvent (**-cs**) is part of the standard GROMACS installation. The spc216.gro is a generic equilibrated three-point solvent model which can be used as the solvent configuration for SPC, SPC/E, or TIP3P water models. Water coordinates are taken from an SPC water system and the **–p** flag adds the new water topology to the topology (.top) file. The generated configuration will be written to 1QGD_solv.gro containing the coordinates of both the protein and the water molecules (Fig. 3).

2.1.5 Neutralizing the system

In the step of the **pdb2gmx**, GROMACS took the protonation state of an amino acid free in solvent at pH 7 as the default. For most of the cases, a charged protein is now in the solvated system. The output of **pdb2gmx** showed the charge of the protein processed, which was −36 e for 1QGD, and this should be neutralized prior to simulations (Note 10). For that, counter ions should be added according to the total charge of the system by using the **genion** module. What **genion** does is to read through the topology and replace water molecules with the ions that the user specifies.

ARTICLE IN PRESS

The input file for **genion** module has an extension of .tpr that is also called as gromacs pre-processor file. This file is produced by the GROMACS **grompp** module, GROMACS pre-processor. What **grompp** does is to assemble simulation parameters (.mdp), topology (.top) and coordinates (.gro) into a single run input (.tpr) file. The file with extension .mdp is a molecular dynamics parameter file that contains a series of settings for running control, energy minimization, output control, neighbor searching, electrostatics, VdW, temperature coupling, pressure coupling and so forth. The .mdp file used at this step can contain any legitimate combination of parameters. The em1.mdp used for energy-minimization of section 2.2 will be used here for generating .tpr file, the required input for module **genion**, as .mdp file for energy minimization is very basic and does not involve any complicated parameter combinations.

em1.mdp

```
cpp=/usr/bin/cpp
define=-DFLEX_SPC
constraints=none
integrator=steep
nsteps=5000
emtol=1000
emstep=0.01
nstcomm=1
ns_type=grid
rlist=1
coulomb_type=PME
rcoulomb=1.0
rvdw=1.0
Tcoupl=no
Pcoupl=no
gen_vel=no
```

Once the .mdp file is prepared, then execute the command:

```
gmx grompp -f em1.mdp -c 1QGD_solv.gro -p topol.top -o ions.tpr
```

The output is the atomic-level description of the simulation system in the binary file **ions.tpr** that contains all the parameters for all of the atoms in the system. This file is then passed into the **genion** module by the command:

ARTICLE IN PRESS

```
gmx genion -s ions.tpr -o 1QGD_solv_ions.gro -p topol.top -pname NA -np 36
```

When prompted, choose group 13 "SOL" for embedding ions. In the **genion** command, the .tpr file is provided as the input (**-s**), a .gro file is generated as output (**-o**), and the topology (**-p**) is updated to reflect the removal of water molecules and addition of ions. The positive and negative ion names are defined by flags **-pname** and **-nname**, respectively. For the structure of TK, sodium ions were added to neutralize the negative charges and the number of the positive ions added was 36 defined by the flag **-np**.

2.2 Energy minimization

During the setup of the simulation system, an unnatural stress might be introduced, for example by placing two atoms accidentally too close to each other. This might result in large forces acting on the particles that would blow up the system in the simulation. Energy minimization (EM) is commonly used to remove these steric clashes or inappropriate geometry and refine low-resolution experimental structures. GROMACS supports different minimization algorithms and the most commonly used are steepest descent and conjugate gradient. The steepest descent algorithm is the quickest in removing the largest strains in the system but converges slowly when close to a minimum. Conjugate gradient is slower than steepest descent in the early stages of the minimization, but becomes more efficient closer to the energy minimum. It is common to do an initial energy minimization using the efficient steepest descent method and a further minimization with a more sophisticated method such as the conjugate gradient algorithm.

The process for EM is much like the addition of ions. We are once again going to use **grompp** to assemble the structure, topology, and simulation parameters into a binary input file (.tpr). Instead of passing the .tpr file to **genion**, we will run the energy minimization through the GROMACS MD engine, **mdrun** (Note 11). The **grompp** command is firstly used to process and prepare the input file necessary for steepest descent energy minimization run.

```
gmx grompp -f em1.mdp -c 1QGD_solv_ions.gro -p topol.top -o em1.tpr
```

Once the .tpr file is ready, submit the job for energy minimization by following command: **gmx mdrun -v -deffnm em1**

Here, the **-v** flag makes **mdrun** create verbose output and print the computation progress to the screen at every step. The **-deffnm** flag defines a default filename prefix that is used for all input and output files. The corresponding file endings will be appended for the different input and output files, so they do not each need to be specified individually. In this step, the following files will be output. The file em1.log contains detailed information on several energy terms throughout the simulation and gives a performance analysis at the end. The file em1.gro contains the coordinates of the final configuration of the system. The file em1.trr contains coordinates and forces for each output frame. The em1.edr file contains information on different energy terms and, if applicable for the simulation, information on the temperature, pressure, volume and some other quantities during the simulation.

EM with conjugate gradient method is carried out after steepest descent method by the following commands:

```
gmx grompp -f em2.mdp -c em1.gro -p topol.top -o em2.tpr
gmx mdrun_d -v -deffnm em2
```

For a minimization using the conjugate gradient method, GROMACS compiled in double precision should be used with the module of **mdrun_d**. The .mdp file used at this step is:

em2.mdp

```
cpp = /usr/bin/cpp
define = -DFLEX_SPC
constraints = none
integrator = cg
nsteps = 5000
emtol = 100
emstep = 0.01
nstcgsteep = 1000
nstcomm = 1
ns_type = grid
rlist = 1
coulomb_type = PME
rcoulomb = 1.0
rvdw = 1.0
Tcoupl = no.
```

ARTICLE IN PRESS

```
Pcoupl = no
gen_vel = no
```

2.3 Position-restraint equilibration

After energy minimization, a reasonable starting structure is obtained for simulations. To begin real dynamics simulations, the solvent and ions around the protein should also be equilibrated and brought to the temperature at which we wish to simulate. In this step, the positions of protein atoms are restrained and the water molecules are allowed to flow.

Equilibration is often conducted in two steps. The system is first brought to the correct temperature based on kinetic energies, and then the correct pressure is applied to the system until it reaches the proper density. The first step is conducted under an NVT ensemble where the number of particles, volume, and temperature are constant. In NVT, the temperature of the system should reach a plateau at the desired value. The second step is conducted under an NPT ensemble where the number of particles, pressure, and temperature are constant, which most closely resembles experimental conditions. In this step, the pressure is stabilized and thus also the density of the system.

2.3.1 NVT equilibration

The posre.itp file generated by **pdb2gmx** is used for position-restraint equilibration, which applies a position restraining force on the heavy atoms of the protein. This allows to equilibrate the solvent around the protein, without the added variable of structural changes in the protein. Typically, 50–100 ps is sufficient (Note 12) for position-restraint equilibration. For the NVT ensemble, the temperature is controlled by V-rescale, the reference temperature is 300 K and the pressure coupling is turned off. The detailed settings used are:

nvt.mdp

```
title = OPLS TK NVT equilibration
define = -DPOSRES
integrator = md
nsteps = 25000
dt = 0.002
nstxout = 100
nstvout = 100
```

```
nstenergy = 100
nstlog = 100
continuation = no
constraint_algorithm = lincs
constraints = all-bonds
lincs_iter = 1
lincs_order = 4
ns_type = grid
nstlist = 5
rlist = 1.0
rcoulomb = 1.0
rvdw = 1.0
coulombtype = PME
pme_order = 4
fourierspacing = 0.16
tcoupl = V-rescale
tc-grps = Protein Non-Protein
tau_t = 0.1   0.1
ref_t = 300   300
pcoupl = no
pbc = xyz
DispCorr = EnerPres
gen_vel = yes
gen_temp = 300
gen_seed = −1
```

The **grompp** and **mdrun** modules are used just as we did at the EM step:

```
gmx grompp -f nvt.mdp -c em2.gro -p topol.top -o nvt.tpr
gmx mdrun -deffnm nvt
```

2.3.2 NPT equilibration

The .mdp file for 50-ps NPT equilibration is not significantly different from the parameter file used for NVT equilibration. The pressure coupling should be added and controlled using Parrinello-Rahman barostat. Since this is continuing the simulation from the NVT equilibration phase, the continuation = yes should be set. Velocities are also read from the previous trajectory, so the velocity generation should be set off by gen_vel = no. The detailed .mdp file for NPT equilibration is as follows:

npt.mdp

```
title=TK NPT equilibration
define=-DPOSRES
integrator=md
nsteps=25000
dt=0.002
nstxout=100
nstvout=100
nstenergy=100
nstlog=100
continuation=yes
constraint_algorithm=lincs
constraints=all-bonds
lincs_iter=1
lincs_order=4
ns_type=grid
nstlist=5
rlist=1.0
rcoulomb=1.0
rvdw=1.0
coulombtype=PME
pme_order=4
fourierspacing=0.16
tcoupl=V-rescale
tc-grps=Protein Non-Protein
tau_t=0.1    0.1
ref_t=300       300
pcoupl=Parrinello-Rahman
pcoupltype=isotropic
tau_p=2.0
ref_p=1.0
compressibility=4.5e-5
refcoord_scaling=com
pbc=xyz
DispCorr=EnerPres
gen_vel        = no
```

The modules **grompp** and **mdrun** will be called just as we did for NVT equilibration.

```
gmx grompp -f npt.mdp -c nvt.gro -t nvt.cpt -p topol.top -o npt.tpr
gmx mdrun -deffnm npt
```

2.4 Production simulation

After completion of the two equilibration phases, the system is well-equilibrated at the desired temperature and pressure, and ready for running production simulation for data collection. The parameter .mdp file needs to be adjusted accordingly. For production run, the position restraints are turned off. Simulations are also carried under an NPT ensemble but the simulation time is significantly longer than the NPT equilibration process (Note 13). For the study of TK, we run a 30-ns MD simulation at 300 K with the following parameter file.

md.mdp

```
title = OPLS TK MD
integrator = md
nsteps = 15000000
dt = 0.002
nstxout = 1000
nstvout = 1000
nstxtcout = 1000
nstenergy = 1000
nstlog = 1000
continuation = yes
constraint_algorithm = lincs
constraints = all-bonds
lincs_iter = 1
lincs_order = 4
ns_type = grid
nstlist = 5
rlist = 1.0
rcoulomb = 1.0
rvdw = 1.0
coulombtype = PME.
pme_order = 4
fourierspacing = 0.16
tcoupl = V-rescale
```

```
tc-grps = Protein Non-Protein
tau_t = 0.1    0.1
ref_t = 300       300
pcoupl = Parrinello-Rahman
pcoupltype = isotropic
tau_p = 2.0
ref_p = 1.0
compressibility = 4.5e-5
pbc = xyz
DispCorr = EnerPres
gen_vel = no
```

The modules **grompp** and **mdrun** will be applied just as we did before (note14).

```
gmx grompp –f md.mdp -c npt.gro -t npt.cpt -p topol.top -o md.tpr
gmx mdrun -deffnm md
```

In general, the final **mdrun** is performed on computer cluster or super computer (Note 14). GROMACS can run in parallel on multiple cores of a single workstation using its built-in thread-MPI. Assuming a standard MPI installation with mpirun tool, launch a simulation with 32 processes using the command:

```
mpirun -np 32 gmx mdrun_mpi -deffnm md
```

2.5 Quality assurance of the simulation

Regarding the quality of the simulation, it is necessary to perform checks to test for the convergence of thermodynamic parameters, such as temperature, pressure, energy, volume, density, and root mean square deviation (RMSD) (Note 15). If any of the thermodynamic parameters has not converged sufficiently, it is necessary to extend the required simulation steps (Note 16). The widely used method for checking the stability of the protein is calculating the RMSD of backbone atoms with respect to the initial structure. During the simulation, the protein diffuses through the unit cell, and may appear "broken" or may "jump" across to the other side of the box. The tool of **trjconv** is a post-processing tool to strip out coordinates, correct for periodicity, or manually alter the trajectory. To account for the periodicity in the system, issue the following command.

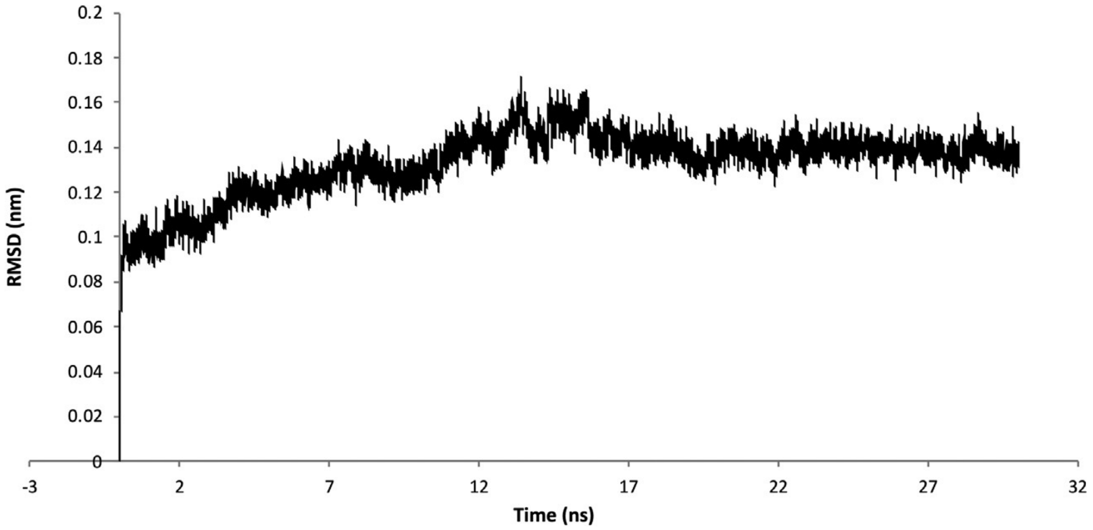

Fig. 4 Backbone RMSD versus simulation time for TK at 300K.

gmx trjconv -s md.tpr -f md.xtc -o mdnoPBC.xtc -pbc mol -center

When prompted, select 1 ("Protein") as the group to be centered and 0 ("System") for output. The input file **md.xtc** is one of the output files of module **mdrun,** which is a light-weight trajectory containing only coordinates. This corrected trajectory **mdnoPBC.xtc** will then be used for further analysis. GROMACS has a built-in tool for RMSD calculations called rms. To use rms, issue this command:

gmx g_rms -s md.tpr -f mdnoPBC.xtc -o rmsd.xvg -tu ns

When prompted, select 4 ("backbone") for both the least-squares fit and the group for RMSD calculation. The **-tu** flag will output the results in terms of ns, even though the trajectory was written in ps. The output plot shows the RMSD values relative to the initial structure in the minimized and equilibrated system (Fig. 4).

2.6 Generation of interesting ensembles

The dynamics cross correlation between atoms within the protein is calculated by analyzing the trajectories of the simulation system. Selecting representative structures from the MD trajectory frames for analysis is critical in the utilization of large structural ensembles. The choice of the selection criteria may vary depending on the goal of the subsequent analysis. The representative structure could be selected based on the RMSD, conformational energy, or certain known conformational changes which may be crucial for protein function. In our protocol, the trajectories of the MD production run for each system were monitored based on the RMSD relative to the initial structure. The backbone RMSD remained at around 0.13 nm and that structure became relatively stable within 15 ns of a 30 ns molecular simulation performed at 300 K (Fig. 4). Based on the results of RMSD, the last 10 ns of the trajectory was used for the calculation of DCCM. In our simulation process, we used an integration time step of 2 fs and the simulated trajectories were saved every 2 ps. The structures are sampled every 10 ps in the 20–30 ns and thus, each system consists of a conformational ensemble of 1001 structures for the production run of the last 10 ns (Fig. 5). The following command is used to get the interesting trajectory.

```
gmx trjconv –f mdnoPBC.xtc –b 20000 –e 30000 –o last10ns.xtc
–skip 5
```

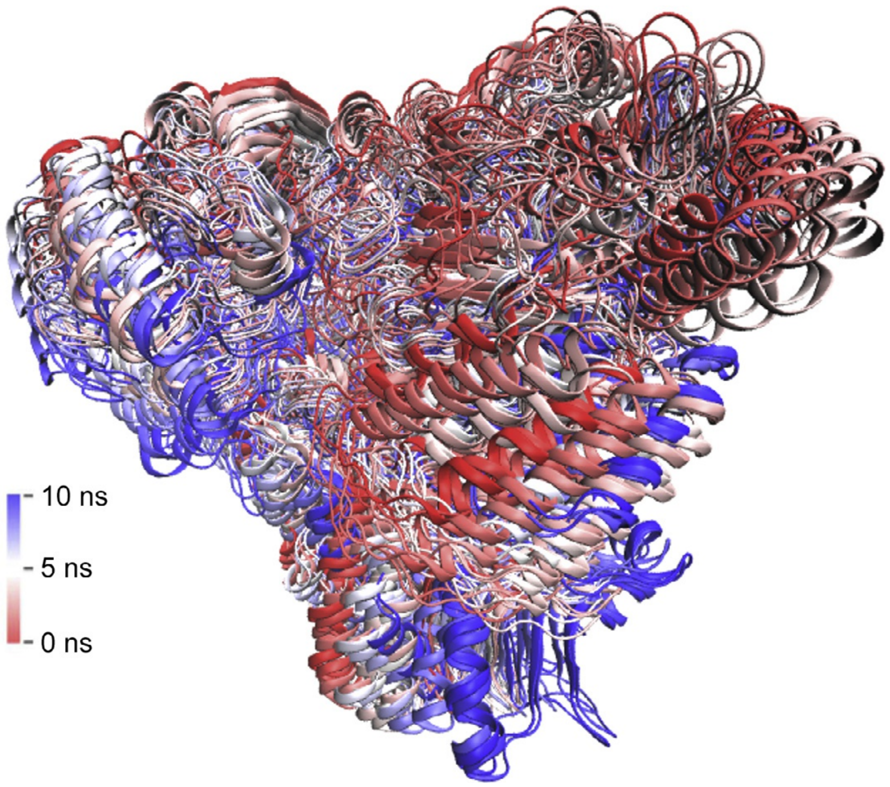

Fig. 5 The structural ensembles generated by MD simulations. For better visibility, only 20 frames within last 10 ns of one MD simulation are displayed. Frames at the start of the simulation are in red, whereas frames at the end are in blue. The pictures were generated by VMD graphics system.

ARTICLE IN PRESS

Here, the **-b 2000** flag indicates that the time (ps) of fist frame to read from the trajectory is 20 ns, **-e 3000** flag indicates that the time (ps) of last frame to read from the trajectory is 30 ns, and **-skip 5** flag indicates that only write the 5th frame to **last10ns.xtc**.

2.7 Notes

1. The resolution of the structure can be found on the RCSB web page or entries listed along with REMARK 2 in a .pdb file, which can be viewed using plain text editors. The quality of the initial structure is a major factor in the reliability of the simulation results. For example, a 2-Å resolution crystal structure would provide a much better starting point than either a 4-Å resolution crystal structure or homology model (Dror, Jensen, Borhani, & Shaw, 2010). The structure with higher resolution should be used as the initial structure for simulation given that no significant missing residues or atoms found in the .pdb file.
2. The missing residues and atoms of a structure could be found in the entries listed along with REMARK 465 and REMARK 470, respectively, in the .pdb file. Terminal regions may be absent, and may not present a problem for dynamics. However, incomplete internal sequences or any amino acid residues that have missing atoms will cause the GROMACS program to fail. These missing atoms and residues must be modeled in using other software packages such DeepView (Guex & Peitsch, 1997) and Modeller (Sali & Blundell, 1993). DeepView (http://www.expasy.ch/ spdbv/) will automatically replace any missing side chains. However, it might mark those rebuilt side chains with a strange control character that can only be removed manually using a text editor. Modeller (https://salilab.org/modeller/wiki/Missing%20residues) can be used to fill in the missing residues by treating the original structure as a template and building a comparative model using the full sequence.
3. The GROMACS software only recognizes the atoms defined by the force field, generally including proteins, nucleic acids, and a very finite number of cofactors, like NAD(H) and ATP. If the ligand is not defined by the force field, it won't be recognized by GROMACS. The ligand can be removed if the ligand binding is not interesting to the researchers. However, if the ligand is required during the simulation, the ligand topology needs to be prepared separately using external tools and then added in the main topology file. It is also advised to remove

the external water molecules that are present in the PDB file to decomplicate the topology. However, if the water molecule serves a structural or catalytic role, they should be retained in the PDB file. In order to remove the unwanted ligands and water molecules, the HETATM lines corresponding to them in the .pdb file can be directly deleted using the plain text editors.

4. A force field is a special case of energy function describing the dependence of the system energy on the coordinates of its particles. It consists of an analytical form of the interatomic potential energy and a set of parameters entering into this form. There are several widely applicable fields including CHARMM (MacKerell et al., 1998), AMBER (Cornell et al., 1996), GROMOS (Oostenbrink, Villa, Mark, & van Gunsteren, 2004), OPLS (Jorgensen, Maxwell, & TiradoRives, 1996) and COMPASS (Sun, 1998). One of the most critical considerations when selecting a force-field for large molecules simulation is whether to employ an all-atom or a united-atom variant. In all-atom (or explicit hydrogen) force-fields, every atom in the molecule is explicitly treated and parameterized. By the contrary, united-atom force-fields do not explicitly treat non-polar hydrogen atoms, but rather group them with carbon atoms and assign parameters to the resultant CH, CH2, or CH3 "pseudo-atoms." However, it is difficult to compare the performance of the existing general force fields, as the result will depend strongly on the system and properties simulated. OPLS (Optimized Potentials for Liquid Simulations) was originally developed to simulate small molecules (Jorgensen, 1981) and an all-atom version (OPLS-AA) was developed later for simulating proteins, with bond stretching and angle bending parameters adopted from AMBER and CHARMM (Jorgensen et al., 1996). OPLS-AA is one of the most widely used biomolecular force fields (Guvench & MacKerell, 2008) and could perform better than other force fields when ligands are to be included (Robertson, Tirado-Rives, & Jorgensen, 2016). For simpler folded protein systems, CHARMM force fields including CHARM27 and CHARM22* can provide more reasonably accurate description of the native states of proteins (Lindorff-Larsen et al., 2012). Users can also add new force field files in the GROMACS directory, and then choose to use them in the pdb2gmx step.

5. Water is a highly polarizable molecule and has a great ability to form hydrogen bonds, both as a hydrogen bond donor and acceptor. For these reasons, water is undoubtedly the most important solvent in biological processes. A large number of water models have been proposed since the first MC simulation of Barker and Watts (1969). Water models can be divided into three types: rigid models, flexible models, and polarizable models (Guimaraes, Barreiro, de Oliveira, & de Alencastro, 2004). The simple, rigid and non-polarizable three-site models, including TIP3P, TIP4P, TIP5P, SPC and SPC/E, are the most commonly used water models in simulations of proteins (Vega & Abascal, 2011). The SPC/E model was originally introduced by Berendsen, Grigera, and Straatsma (1987) and remains a popular choice for use in molecular simulations. The model consists of point charges located at the nuclear positions of the hydrogen and oxygen atoms. In addition to the intermolecular electrostatic interactions, oxygen atoms interact through a Lennard-Jones potential. The hydrogen-oxygen bond distance is fixed at 1 Å and the hydrogen-oxygen-hydrogen bond angle is fixed at 109.5°.

6. GROMACS takes the protonation state of an amino-acid free in solvent at pH 7 as the default. Lys and Arg are protonated, while the Asp and Glu are unprotonated. For His, the proton could be either on ND1, on NE2 or on both and the selections are done automatically based on optimal hydrogen-bonding conformations.
7. The .gro file is not mandatory for running simulations with GROMACS. GROMACS can handle many different file formats. If you prefer to use, for instance, the PDB format, all you need to do is to specify an appropriate file name with .pdb extension as your output.
8. The distance to the edge of the box is an important parameter. Since we will be using periodic boundary conditions, the minimum image convention must be satisfied. Specifying a solute-box distance of 1.0 nm means that there are at least 2.0 nm between any two periodic images of a protein. This distance is sufficient for any cut-off schemes commonly used in simulations.
9. There are also other box types available such as triclinic, dodecahedron, and octahedron. Cubic is a rectangular box with all sides equal, whereas dodecahedron represents a rhombic dodecahedron, and octahedron is a truncated version of octahedron. Fewer solvent molecules are required to fill the box given a minimum distance between macromolecular images. The rhombic dodecahedron and the truncated octahedron are closer to being a sphere than a cube is, and are therefore better suited to the study of an approximately spherical macromolecule in solution.

10. Since the system has non-zero total charge, counter-ions are required for the system to neutralize the charge. If the system charge is not very close to an integer after pdb2gmx, there may be a problem with the topology. The topology file should be checked by looking at the right-hand comment column of the atom listing, which lists the cumulative charge. This should be an integer after every charged residue. If the charge is already close to an integer, then the difference is caused by rounding errors and is not a major problem.

11. There are two factors to determine if energy minimization is successful or not. The first factor is that the potential energy, E_{pot} which should be negative on the order of 10^5–10^6, depending on the system size and number of water molecules. The second important feature is that the maximum force, F_{max} should be no greater than 1000 kJ mol^{-1} nm^{-1}. If the energy minimization process arrives at a reasonable E_{pot} with an F_{max} greater than 1000 kJ mol^{-1} nm^{-1}, the system may not be stable enough for simulation. In this case, the minimization parameters should be adjusted accordingly. The potential energy, E_{pot} can be extracted from the em1.edr file that contains all of the energy terms that GROMACS collects during energy minimization, by the GROMACS **energy** module: gmx energy -f em1.edr -o potential.xvg. At the command prompt, type "10" to select Potential, "0" to terminate the input.

12. For a small protein, 50–100 ps (25,000–50,000 steps) should be more than enough for the water to equilibrate around the protein. However, for a large simulation system such as a membrane system, the water and slow lipid motions can require several nanoseconds of relaxation. The only way to know for certain is to check the potential energy, and the equilibration should be extended until it has converged.

13. Errors in molecular simulation arise from two factors: inaccuracy in the models and insufficient sampling (Grossfield & Zuckerman, 2009). Simulation time is an important factor determining the adequate sampling. For decent sampling the simulation should be at least 10 times longer than the interesting phenomena, which unfortunately sometimes conflicts with reality and available computer resources. But as a rule of thumb, longer simulations are much better than shorter MD runs. Even then, there is still a need to confirm that the system has converged to an experimentally relevant ensemble.

14. Considering the huge computer resources needed in the production simulation, the final mdrun is generally performed on a computer

ARTICLE IN PRESS

cluster or super computer. The remote servers can be connected via ssh clients such as putty from windows and the file transfers can be performed by using tools such as FileZilla and WinSCP tool.

15. When MD simulations are performed, various quantities are calculated and written to output files. The thermodynamic measures can be used as an indication of simulation quality. For example, the temperature in a simulation is expected to be fluctuating about a constant value, within a small range, over time. If this is not the case, then the quality of the simulation is suspicious. The quality-indicating measures suggested to be considered include temperature, pressure, potential energy, kinetic energy, number density, volume, cell dimensions, and specific heat capacity (Murdock et al., 2006). Apart from the thermodynamic parameters, the root mean square deviation (RMSD) should be used to examine the convergence of a simulation. The RMSD provides a measure of conformational stability and for a converging simulation, RMSD is expected to increase and then start to plateau, when calculated with respect to the initial configuration in the trajectory. The radius of gyration can be also calculated to give a measure of how the atoms of a protein are distributed around their center of mass. For converged trajectories, the radius of gyration time-series of a protein should also reach a plateau.

16. If the simulation has terminated but not completed due to queue limits or power failure, the simulation can be continued by using the command:

```
gmx mdrun –s md.tpr –cpi md.cpt –append
```

where **md.cpt** file is a checkpoint file. If the simulation has completed and there is a need for extension, the following commands can be used:

```
gmx covert-tpr –s md.tpr –extend timetoextendby –o next.tpr
gmx mdrun –s next.tpr –cp md.cpt –append
```

3. Dynamic cross-correlation network

Bio3D will be used to calculate the dynamics cross correlation network of TK (Grant et al., 2006; Skjaerven, Yao, Scarabelli, & Grant, 2014). A GROMACS analysis tool **g_covar** can also be used to calculate the dynamics covariance matrix. However, a further post processing of the results from **g_covar** is needed to obtain a dynamical cross-correlation matrix. Bio3D is an R package that provides user-friendly interactive tools for the analysis of biomolecular structure, sequence and simulation data. This package returns a matrix of all atom-wise cross-correlations whose elements, correlation coefficients C_{ij}, can be displayed in a graphical representation termed a dynamical cross-correlation matrix.

3.1 Preparing trajectory file

Since the trajectory files including .xtc file and .trr file from GROMACS are not supported by Bio3D, these files should be converted to DCD format, that are single precision binary FORTRAN files and accepted by the Bio3D. The trajectory files produced by CHARMM/NAMD are also in DCD format. Here, a tool called **CatDCD** (http://www.ks.uiuc.edu/Development/MDTools/catdcd/) is used to transfer .xtc file to .dcd file. **CatDCD** functions much like the Unix "cat" command. It can concatenate DCD files into a single DCD file, write only selected atoms to the final DCD file and read/write any of the structure/trajectory formats that are supported by VMD (https://www.ks.uiuc.edu/Research/vmd/). **CatDCD** is now built as part of the VMD Plugin. To use **CatDCD**, just copy it over to the home directories and use this command in the command prompt:

```
./catdcd –o last10ns.dcd –xtc last10ns.xtc
```

where the **-o** flag indicates output file, **-xtc** flag indicates the file format of the input file last10ns.xtc obtained from the GROMACS production simulation trajectory.

3.2 Loading Bio3D package

Bio3D is an R package and can be installed on all platforms (Mac, Linux and PC) following the official tutorials (http://thegrantlab.org/bio3d/tutorials/installing-bio3d). To get started, start R first by typing **R** at the command prompt or, on Windows, double clicking on the R icon. After that, the Bio3D package can be loaded by using the command at the R console prompt:

```
>library(bio3d)
```

3.3 Reading a trajectory file

The trajectory file last10ns.dcd is now ready to be input into the Bio3D package by the following commands:

```
> dcdfile <- "/path/to/dcd/file/last10ns.dcd"
> dcd <- read.dcd(dcdfile)
```

The file last10ns.dcd is first assigned to the object **dcdfile**. The function **read.dcd()** then processes the input trajectory file and returns its output to the new object **dcd.** This can be checked by the following command:

```
>print(dcd)
```

The output of this command is:

```
Total Frames#: 1001
Total XYZs#: 60138, (Atoms#: 20046)
 [1] 33.52 65.21 30.84 <...> 65.04 41.75 78.91 [60198138]
+ attr: Matrix DIM = 1001 x 60138
```

This indicates that there are 1001 frames in total in the trajectory, and that there are 20,046 atoms and 60,138 coordinates in each frame.

3.4 Reading the starting PDB file

A PDB file needs to be read to determine atom correspondence by the following commands:

```
>pdbfile <- "/path/to/pdb/file/1QGD.pdb"
> pdb <- read.pdb(pdbfile)
```

The simulation initial structure **1QGD.pdb** is first assigned to the object **pdbfile**. The function **read.pdb()** then processes the PDB structure and returns its output to the new object **pdb.** This object can be checked by the following command:

```
> print(pdb)
```

```
Call: read.pdb(file = pdbfile)

   Total Models#: 1
     Total Atoms#: 20046, XYZs#: 60138 Chains#: 2 (values: A B)
     Protein Atoms#: 20046 (residues/Calpha atoms#: 1324)
```

ARTICLE IN PRESS

```
    Nucleic acid Atoms#: 0 (residues/phosphate atoms#: 0)
    Non-protein/nucleic Atoms#: 0 (residues: 0)
    Non-protein/nucleic resid values: [ none ]
  Protein sequence:
    SSRKELANAIRALSMDAVQKAKSGHPGAPMGMADIAEVLWRDFLKHNPQNPSWADRDRFV
    LSNGHGSMLIYSLLHLTGYDLPMEELKNFRQLHSKTPGHPEVGKTAGVETTTGPLGQGIA
    NAVGMAIAEKTLAAQFNRPGHDIVDHYTYAFMGDGCMMEGISHEVCSLAGTLKLGKLIAF
    YDDNGISIDGHVEGWFTDDTAMRFEAYGWHVIRDIDGHDAASIKR...<cut>...KELL
+ attr: atom, xyz, calpha, call
```

The output indicates the information about the PDB structure including atom number, chain number, non-protein atoms and protein sequence.

3.5 Superposing trajectory frames

In this step, the frames in the trajectory will be superposed. The interesting atoms will be first identified by using the function **atom.select()**.

```
> ca.inds <- atom.select(pdb, elety = “CA”)
```

Here, all the C-alpha atoms are selected. The returned object **ca.inds** is a list containing atom and xyz numeric indices that we can now use to superpose all frames of the trajectory on the selected indices. For this, the **fit.xyz()** function will be used.

```
> xyz <- fit.xyz(fixed = pdb$xyz, mobile = dcd, fixed.inds = ca.inds$xyz, mobile.inds = ca.inds$xyz)
```

The above command performs the actual superposition and stores the new coordinates in the matrix object **xyz**.

3.6 Running cross-correlation analysis

The function **dccm()** will be used to calculate the cross-correlation matrices and this function will return a matrix of all atom-wise cross-correlations. The command is:

```
> cij <- dccm(xyz[, ca.inds$xyz])
```

The function **dccm()** calculates the DCCM using the above object **xyz** as the input and returns its output to the new object **cij.** The DCCM can be visualized by using the command (Fig. 6):

```
> plot(cij)
```

ARTICLE IN PRESS

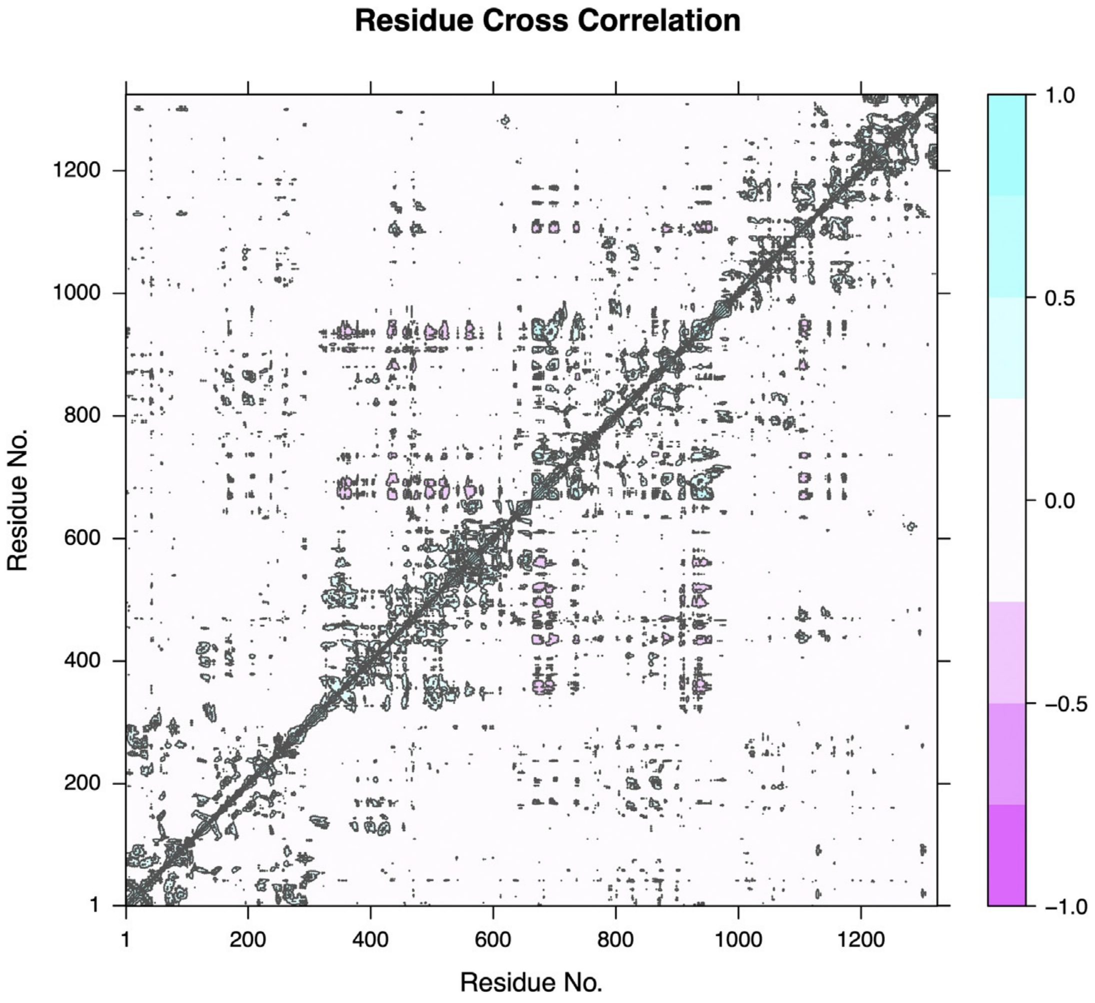

Fig. 6 Dynamic cross-correlation matrix (DCCM) for C$^{\alpha}$ atom pairs calculated with dccm() function and plotted with plot() function.

3.7 Generating a dynamical cross-correlation matrix

The residue number 1324 in the DCCM calculated by Bio3D represents the addition of the two chains (Fig. 6). Since TK is a homo-dimer, the correlation coefficients from the two chains can be averaged to investigate the dynamics correlations within the same monomer. The matrix can be calculated using **R** and the final matrix data can be saved to a .csv file that can be opened using Microsoft Excel using the following commands.

Execute following command to assign the rows 1–662 and columns 1–622 of matrix **cij** to a new object **ChainA**:

```
> ChainA <- cij[1:662,1:662]
```

Execute following command to assign the rows 663–1324 and columns 663–1324 of matrix **cij** to a new object **ChainB**:

```
> ChainB <- cij[663:1324,663:1324]
```

Execute the following command to average the correlation coefficients of two chains and return the result to a new object **average**.

```
> average <- monomerA/2+monomerB/2
```

This matrix **average** can then be written to a **.csv** file in the directory you want it to be.

```
> write.csv(average, file="/path/to/where/you/want/it/to/
be/average.csv")
```

The average.csv can be opened using Microsoft Excel and the matrix data can be imported to software such as OriginPro9.0 for better visualization (Fig. 7).

With the DCCM (Fig. 7), it is easy to identify the residues maintaining dynamic cross correlation with the critical residues for protein engineering. These critical residues could be active sites that are hot-spots for engineering enzyme functions, flexible sites that are hot-spots for engineering protein stability through rigidification, access tunnel sites that are hot-spots for

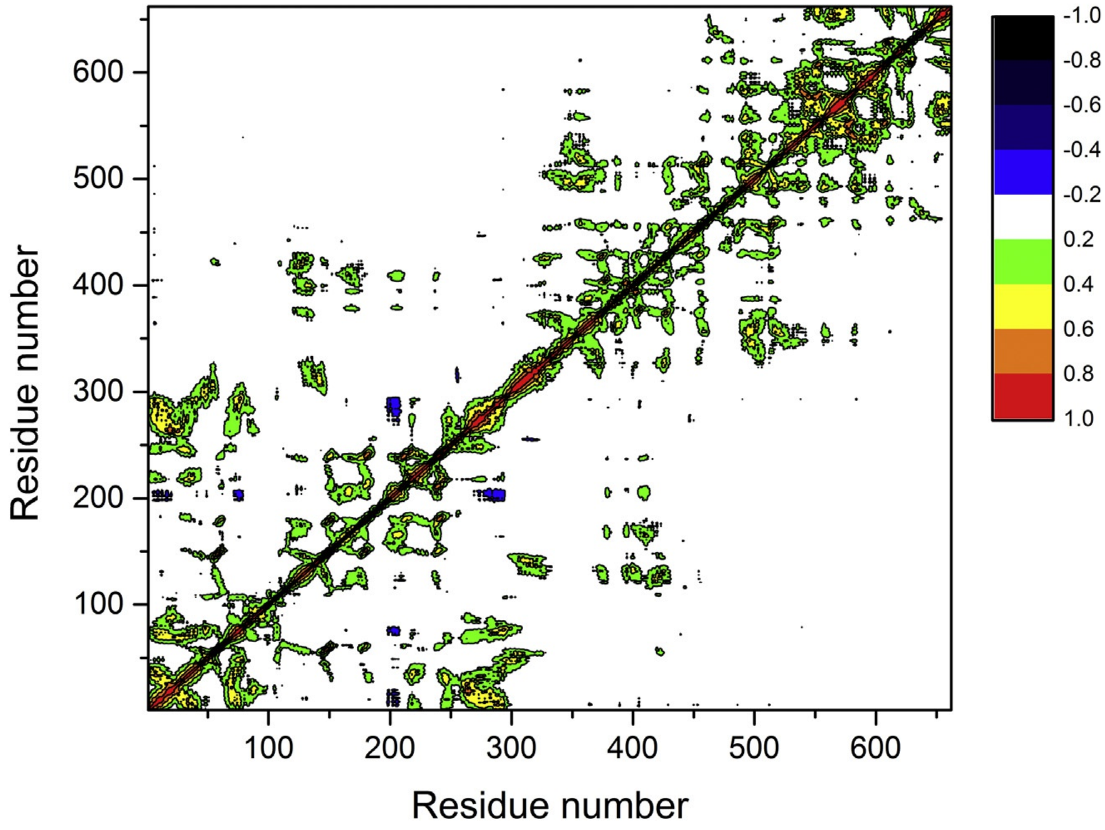

Fig. 7 Dynamics cross-correlation map for the C^{α} atom pairs within monomers of TK. Correlation coefficient (C_{ij}) was shown as different colors. C_{ij} with values from 0 to 1 represents positive correlations, whereas C_{ij} with values from −1 to 0 represents negative correlations.

engineering substrate specificity, or interface sites that are hot-spots for the engineering of multimeric enzymes. Both these hotspots and the residues having dynamic correlation with them should be considered when developing future protein-engineering strategies. We have used DCCM to identify mutations that counteracted the activity–stability trade-off in a transketolase variant 3M containing mutations S385Y/D469T/R520Q (Yu & Dalby, 2018b). This variant was previously engineered to be active on three benzaldehyde derivatives, in contrast to WT TK, which was active only on non-aromatic aldehydes. However, the variant achieved new functions at the cost of a significant trade-off in thermal stability due to the destabilization of active sites located at the dimer interface. Modifying the flexible active sites for restoring stability risks losing activity due to the activity–stability trade-off. We hence used the DCCM to select regions outside the active site, whose dynamics were highly correlated to flexible active sites, as the targets for stabilizing mutations, and successfully obtained the variants showing enhanced stability but with no loss in activity (Yu & Dalby, 2018b).

4. Summary and conclusion

The protein consists of a series of correlation networks formed by interacting amino-acid residues. Because of these amino-acid networks, the effect that a single-site amino-acid has on the protein structure and function is related not only to the type of amino acid at that site but also to its neighboring amino acids and even amino acids at more distant sites. Some of these networks are even more important than others as they directly influence the protein's function and structure, and will hence be critical for developing protein engineering strategies. The dynamic cross-correlation network is undoubtedly one of these important networks. With the protocols described above, we hope to contribute to developing protein engineering strategies based on targeting residues within dynamic cross-correlation networks that influence particular protein properties. In addition, this protocol will definitely assist explorations of the mechanisms of long-range dynamic correlation.

Acknowledgments

We thank the Chinese Scholarship Council (stipend to H. Y.), and EPSRC EP/N025105/1 & EP/P006485/1 (consumables), for financial support. We thank Dr. Cheng Zhang for his advice in the MD simulations protocol writing.

Reference

Barker, J. A., & Watts, R. O. (1969). Structure of water; a Monte Carlo calculation. *Chemical Physics Letters*, *3*(3), 144–145. https://doi.org/10.1016/0009-2614(69)80119-3.

Berendsen, H. J. C., Grigera, J. R., & Straatsma, T. P. (1987). The missing term in effective pair potentials. *Journal of Physical Chemistry*, *91*(24), 6269–6271. https://doi.org/10.1021/j100308a038.

Bowers, K. J., Chow, D. E., Xu, H., Dror, R. O., Eastwood, M. P., Gregersen, B. A., et al. (2006). Scalable algorithms for molecular dynamics simulations on commodity clusters. In *Paper presented at the SC'06: Proceedings of the 2006 ACM/IEEE conference on supercomputing*.

Brooks, B. R., Brooks, C. L., 3rd, Mackerell, A. D., Jr., Nilsson, L., Petrella, R. J., Roux, B., et al. (2009). CHARMM: The biomolecular simulation program. *Journal of Computational Chemistry*, *30*(10), 1545–1614. https://doi.org/10.1002/jcc.21287.

Case, D. A., Cheatham, T. E., 3rd, Darden, T., Gohlke, H., Luo, R., Merz, K. M., Jr., et al. (2005). The Amber biomolecular simulation programs. *Journal of Computational Chemistry*, *26*(16), 1668–1688. https://doi.org/10.1002/jcc.20290.

Clarkson, M. W., Gilmore, S. A., Edgell, M. H., & Lee, A. L. (2006). Dynamic coupling and allosteric behavior in a nonallosteric protein. *Biochemistry*, *45*(25), 7693–7699. https://doi.org/10.1021/bi060652l.

Cornell, W. D., Cieplak, P., Bayly, C. I., Gould, I. R., Merz, K. M., Ferguson, D. M., et al. (1996). A second generation force field for the simulation of proteins, nucleic acids, and organic molecules (vol 117, pg 5179, 1995). *Journal of the American Chemical Society*, *118*(9), 2309. https://doi.org/10.1021/ja955032e.

Dassault Systèmes BIOVIA. (2016). *Discovery studio modeling environment, release 2017*. San Diego: Dassault Systèmes.

Doshi, U., Holliday, M. J., Eisenmesser, E. Z., & Hamelberg, D. (2016). Dynamical network of residue-residue contacts reveals coupled allosteric effects in recognition, catalysis, and mutation. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(17), 4735–4740. https://doi.org/10.1073/pnas.1523573113.

Draths, K., Pompliano, D., Conley, D., Frost, J., Berry, A., Disbrow, G., et al. (1992). Biocatalytic synthesis of aromatics from D-glucose: The role of transketolase. *Journal of the American Chemical Society*, *114*(10), 3956–3962.

Dror, R. O., Jensen, M. O., Borhani, D. W., & Shaw, D. E. (2010). Exploring atomic resolution physiology on a femtosecond to millisecond timescale using molecular dynamics simulations. *Journal of General Physiology*, *135*(6), 555–562. https://doi.org/10.1085/jgp.200910373.

DuBay, K. H., Bowman, G. R., & Geissler, P. L. (2015). Fluctuations within folded proteins: Implications for thermodynamic and allosteric regulation. *Accounts of Chemical Research*, *48*(4), 1098–1105. https://doi.org/10.1021/ar500351b.

Estabrook, R. A., Luo, J., Purdy, M. M., Sharma, V., Weakliem, P., Bruice, T. C., et al. (2005). Statistical coevolution analysis and molecular dynamics: Identification of amino acid pairs essential for catalysis. *Proceedings of the National Academy of Sciences of the United States of America*, *102*(4), 994–999. https://doi.org/10.1073/pnas.0409128102.

Gagne, D., French, R. L., Narayanan, C., Simonovic, M., Agarwal, P. K., & Doucet, N. (2015). Perturbation of the conformational dynamics of an active-site loop alters enzyme activity. *Structure*, *23*(12), 2256–2266. https://doi.org/10.1016/j.str.2015.10.011.

Grant, B. J., Rodrigues, A. P., ElSawy, K. M., McCammon, J. A., & Caves, L. S. (2006). Bio3d: An R package for the comparative analysis of protein structures. *Bioinformatics*, *22*(21), 2695–2696. https://doi.org/10.1093/bioinformatics/btl461.

Grossfield, A., & Zuckerman, D. M. (2009). Quantifying uncertainty and sampling quality in biomolecular simulations. *Annual Reports in Computational Chemistry*, *5*, 23–48. https://doi.org/10.1016/S1574-1400(09)00502-7.

Guex, N., & Peitsch, M. C. (1997). SWISS-MODEL and the Swiss-PdbViewer: An environment for comparative protein modeling. *Electrophoresis*, *18*(15), 2714–2723. https://doi.org/10.1002/elps.1150181505.

Guimaraes, C. R. W., Barreiro, G., de Oliveira, C. A. F., & de Alencastro, R. B. (2004). On the application of simple explicit water models to the simulations of biomolecules. *Brazilian Journal of Physics*, *34*(1), 126–136. https://doi.org/10.1590/S0103-97332004000100016.

Guvench, O., & MacKerell, A. D., Jr. (2008). Comparison of protein force fields for molecular dynamics simulations. *Methods in Molecular Biology*, *443*, 63–88. https://doi.org/10.1007/978-1-59745-177-2_4.

Henzler-Wildman, K., & Kern, D. (2007). Dynamic personalities of proteins. *Nature*, *450*(7172), 964–972. https://doi.org/10.1038/nature06522.

Herzberg, O., & Moult, J. (1991). Analysis of the steric strain in the polypeptide backbone of protein molecules. *Proteins*, *11*(3), 223–229. https://doi.org/10.1002/prot.340110307.

Ichiye, T., & Karplus, M. (1991). Collective motions in proteins: A covariance analysis of atomic fluctuations in molecular dynamics and normal mode simulations. *Proteins*, *11*(3), 205–217. https://doi.org/10.1002/prot.340110305.

Jorgensen, W. L. (1981). Quantum and statistical mechanical studies of liquids .10. Transferable intermolecular potential functions for water, alcohols, and ethers—Application to liquid water. *Journal of the American Chemical Society*, *103*(2), 335–340. https://doi.org/10.1021/ja00392a016.

Jorgensen, W. L., Maxwell, D. S., & TiradoRives, J. (1996). Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *Journal of the American Chemical Society*, *118*(45), 11225–11236. https://doi.org/10.1021/ja9621760.

Lindahl, E., Hess, B., & van der Spoel, D. (2001). GROMACS 3.0: A package for molecular simulation and trajectory analysis. *Journal of Molecular Modeling*, 7(8), 306–317. https://doi.org/10.1007/s008940100045.

Lindorff-Larsen, K., Maragakis, P., Piana, S., Eastwood, M. P., Dror, R. O., & Shaw, D. E. (2012). Systematic validation of protein force fields against experimental data. *PLoS One*, 7(2), e32131. https://doi.org/10.1371/journal.pone.0032131.

Littlechild, J., Turner, N., Hobbs, G., Lilly, M., Rawas, A., & Watson, H. (1995). Crystallization and preliminary X-ray crystallographic data with Escherichia coli transketolase. *Acta Crystallographica. Section D, Biological Crystallography*, *51*(Pt. 6), 1074–1076. https://doi.org/10.1107/S0907444995005415.

MacKerell, A. D., Bashford, D., Bellott, M., Dunbrack, R. L., Evanseck, J. D., Field, M. J., et al. (1998). All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry. B*, *102*(18), 3586–3616. https://doi.org/10.1021/jp973084f.

Mazal, H., & Haran, G. (2019). Single-molecule FRET methods to study the dynamics of proteins at work. *Current Opinion in Biomedical Engineering*, *12*, 8–17. https://doi.org/10.1016/j.cobme.2019.08.007.

Murdock, S. E., Tai, K., Ng, M. H., Johnston, S., Wu, B., Fangohr, H., et al. (2006). Quality assurance for biomolecular simulations. *Journal of Chemical Theory and Computation*, *2*(6), 1477–1481. https://doi.org/10.1021/ct6001708.

Naganathan, A. N. (2019). Modulation of allosteric coupling by mutations: From protein dynamics and packing to altered native ensembles and function. *Current Opinion in Structural Biology*, *54*, 1–9. https://doi.org/10.1016/j.sbi.2018.09.004.

Oostenbrink, C., Villa, A., Mark, A. E., & van Gunsteren, W. F. (2004). A biomolecular force field based on the free enthalpy of hydration and solvation: The GROMOS force-field parameter sets 53A5 and 53A6. *Journal of Computational Chemistry*, *25*(13), 1656–1676. https://doi.org/10.1002/jcc.20090.

O'Rourke, K. F., Gorman, S. D., & Boehr, D. D. (2016). Biophysical and computational methods to analyze amino acid interaction networks in proteins. *Computational and Structural Biotechnology Journal*, *14*, 245–251. https://doi.org/10.1016/j.csbj.2016.06.002.

Palmer, A. G., 3rd. (2014). Chemical exchange in biomacromolecules: Past, present, and future. *Journal of Magnetic Resonance*, *241*, 3–17. https://doi.org/10.1016/j.jmr.2014.01.008.

Phillips, J. C., Braun, R., Wang, W., Gumbart, J., Tajkhorshid, E., Villa, E., et al. (2005). Scalable molecular dynamics with NAMD. *Journal of Computational Chemistry*, *26*(16), 1781–1802. https://doi.org/10.1002/jcc.20289.

Plimpton, S. (1995). Fast parallel algorithms for short-range molecular-dynamics. *Journal of Computational Physics*, *117*(1), 1–19. https://doi.org/10.1006/jcph.1995.1039.

Reetz, M. T. (2013). The importance of additive and non-additive mutational effects in protein engineering. *Angewandte Chemie (International Ed. in English)*, *52*(10), 2658–2666. https://doi.org/10.1002/anie.201207842.

Robertson, M. J., Tirado-Rives, J., & Jorgensen, W. L. (2016). Performance of protein-ligand force fields for the flavodoxin-flavin mononucleotide system. *Journal of Physical Chemistry Letters*, *7*(15), 3032–3036. https://doi.org/10.1021/acs.jpclett.6b01229.

Roy, A., Kucukural, A., & Zhang, Y. (2010). I-TASSER: A unified platform for automated protein structure and function prediction. *Nature Protocols*, *5*(4), 725–738. https://doi.org/10.1038/nprot.2010.5.

Sali, A., & Blundell, T. L. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, *234*(3), 779–815. https://doi.org/10.1006/jmbi.1993.1626.

Selvaratnam, R., Chowdhury, S., VanSchouwen, B., & Melacini, G. (2011). Mapping allostery through the covariance analysis of NMR chemical shifts. *Proceedings of the National Academy of Sciences of the United States of America*, *108*(15), 6133–6138. https://doi.org/10.1073/pnas.1017311108.

Sharir-Ivry, A., & Xia, Y. (2018). Nature of long-range evolutionary constraint in enzymes: Insights from comparison to pseudoenzymes with similar structures. *Molecular Biology and Evolution*, *35*(11), 2597–2606. https://doi.org/10.1093/molbev/msy177.

Singh, B. (2004). PepBuild: A web server for building structure data of peptides/proteins. *Nucleic Acids Research*, *32*(Web Server issue), W559–W561. https://doi.org/10.1093/nar/gkh472.

Skjaerven, L., Yao, X. Q., Scarabelli, G., & Grant, B. J. (2014). Integrating protein structural dynamics and evolutionary analysis with Bio3D. *BMC Bioinformatics*, *15*, 399. https://doi.org/10.1186/s12859-014-0399-6.

Souza, V. P., Ikegami, C. M., Arantes, G. M., & Marana, S. R. (2018). Mutations close to a hub residue affect the distant active site of a GH1 beta-glucosidase. *PLoS One*, *13*(6), e0198696. https://doi.org/10.1371/journal.pone.0198696.

Sprenger, G. A., Schorken, U., Sprenger, G., & Sahm, H. (1995). Transketolase A of Escherichia coli K12. Purification and properties of the enzyme from recombinant strains. *European Journal of Biochemistry*, *230*(2), 525–532. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/7607225.

Subramanian, K., Mitusinska, K., Raedts, J., Almourfi, F., Joosten, H. J., Hendriks, S., et al. (2019). Distant non-obvious mutations influence the activity of a hyperthermophilic pyrococcus furiosus phosphoglucose isomerase. *Biomolecules*, *9*(6), 212. https://doi.org/10.3390/biom9060212.

Sun, H. (1998). COMPASS: An ab initio force-field optimized for condensed-phase applications—Overview with details on alkane and benzene compounds. *Journal of Physical Chemistry B*, *102*(38), 7338–7364. https://doi.org/10.1021/jp980939v.

Swaminathan, S., Harte, W. E., & Beveridge, D. L. (1991). Investigation of domain-structure in proteins via molecular-dynamics simulation—Application to Hiv-1 protease dimer. *Journal of the American Chemical Society*, *113*(7), 2717–2721. https://doi.org/10.1021/ja00007a054.

Tyukhtenko, S., Rajarshi, G., Karageorgos, I., Zvonok, N., Gallagher, E. S., Huang, H., et al. (2018). Effects of distal mutations on the structure, dynamics and catalysis of human monoacylglycerol lipase. *Scientific Reports*, *8*(1), 1719. https://doi.org/10.1038/s41598-017-19135-7.

Vega, C., & Abascal, J. L. (2011). Simulating water with rigid non-polarizable models: A general perspective. *Physical Chemistry Chemical Physics*, *13*(44), 19663–19688. https://doi.org/10.1039/c1cp22168j.

Wilding, M., Hong, N., Spence, M., Buckle, A. M., & Jackson, C. J. (2019). Protein engineering: The potential of remote mutations. *Biochemical Society Transactions*, *47*(2), 701–711. https://doi.org/10.1042/BST20180614.

Wrenbeck, E. E., Azouz, L. R., & Whitehead, T. A. (2017). Single-mutation fitness landscapes for an enzyme on multiple substrates reveal specificity is globally encoded. *Nature Communications*, *8*, 15695. https://doi.org/10.1038/ncomms15695.

Yang, L. Q., Sang, P., Tao, Y., Fu, Y. X., Zhang, K. Q., Xie, Y. H., et al. (2014). Protein dynamics and motions in relation to their functions: Several case studies and the underlying mechanisms. *Journal of Biomolecular Structure & Dynamics*, *32*(3), 372–393. https://doi.org/10.1080/07391102.2013.770372.

Yu, H., & Dalby, P. A. (2018a). Coupled molecular dynamics mediate long- and short-range epistasis between mutations that affect stability and aggregation kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(47), E11043–E11052. https://doi.org/10.1073/pnas.1810324115.

Yu, H., & Dalby, P. A. (2018b). Exploiting correlated molecular-dynamics networks to counteract enzyme activity-stability trade-off. *Proceedings of the National Academy of Sciences of the United States of America*, *115*(52), E12192–E12200. https://doi.org/10.1073/pnas.1812204115.

Zhang, T., Liu, L. A., Lewis, D. F. V., & Wei, D. Q. (2011). Long-range effects of a peripheral mutation on the enzymatic activity of cytochrome P450 1A2. *Journal of Chemical Information and Modeling*, *51*(6), 1336–1346. https://doi.org/10.1021/ci200112b.

Zhuravleva, A., Korzhnev, D. M., Nolde, S. B., Kay, L. E., Arseniev, A. S., Billeter, M., et al. (2007). Propagation of dynamic changes in barnase upon binding of barstar: An NMR and computational study. *Journal of Molecular Biology*, *367*(4), 1079–1092. https://doi.org/10.1016/j.jmb.2007.01.051.