Supplementary Information for RapidRMSD : Rapid determination of RMSDs corresponding to motions of flexible molecules

Emilie Neveu[1,2], Petr Popov[3], Alexandre Hoffmann[1], Angelo Migliosi[4], Xavier Besseron[4], Grégoire Danoy[4], Pascal Bouvry[4] and Sergei Grudinin*[1]

[1]Inria / Univ. Grenoble Alpes / LJK-CNRS, F-38000 Grenoble, France
[2]Faculty of Biology and Medicine, University of Lausanne, 1015 Lausanne, Switzerland
[3]Moscow Institute of Physics and Technology, Dolgoprudniy, Russia
[4]University of Luxembourg, L-1359 Luxembourg

This document describes technical details and protocols presented in the main text of the article.

Methods

NMA computations

As we have mentioned in the main text, one of the practical ways to compute collective motions for molecular systems is the normal mode analysis (NMA). For all the computational experiments that involve normal modes, we have used our NMA approach called NOLB, which stand for NOnLinear rigid Block method [1]. This is an extension of the rotation-translation of blocks approach developed by Y.-H. Sanejouand and colleagues [2, 3]. NOLB is a particularly efficient implementation of NMA in terms of CPU and memory consumption, if only a few normal modes are required. This is typically the scenario in flexible docking studies, when flexibility is modeled with collective coordinates. We should mention that NOLB allows computation of the nonlinear normal modes. However, only the linear normal modes were used in the presented study. For example, transitions between the unbound and bound states of proteins from Figure 3 in the main text were computed as follows, "NOLB unbound.pdb bound.pdb". Calculations for all the receptors and ligands of 216 complexes of the benchmark took about 2 minutes on a MacBook laptop. The cutoff interaction distance of 5 Å was used, and 20 lowest normal modes were computed. The unbound and bound proteins were first aligned and then superposed. The best combination of normal modes was calculated solving the least square problem.

Rigid-body cross-docking

For the cross-docking experiment we used three examples described in the main text. For the first example we used the bound form of the 1ibr receptor and 100 pseudo-random structural ensembles of the unbound form of the 1ibr ligand at 3 Å deformation amplitude generated along the first two modes. These were generated using the following command, "NOLB 1IBR_l_u.pdb -r 3.0 -n 2 -s 100". We performed 100 rigid-body docking computations for this example. For the second example we used the bound form of the 1zli receptor and 100

*sergei.grudinin@inria.fr

pseudo-random structural ensembles of the unbound form of the 1zli ligand at 3.9 Å deformation amplitude generated along the first six modes. These were generated using the following command, "NOLB 1ZLI_l_u.pdb -n 6 -r 3.9 -s 100". We performed 100 rigid-body docking computations for this example. Finally, as the last example we chose a challenging case of the 2i9b complex with both receptor and ligand being flexible. We generated 40 pseudo-random structural ensembles of the unbound form of the 2i9b receptor at 4 Å deformation amplitude generated along the first four modes. We also generated 60 pseudo-random structural ensembles of the unbound form of the 2i9b ligand at 2 Å deformation amplitude generated along the first four modes. We used a larger number of ligand models because of long flexible loop at the binding site of the ligand. These models were generated using the following two commands, "NOLB 2I9B_r_u.pdb -r 4.0 -n 4 -s 40", and "NOLB 2I9B_l_u.pdb -r 2.0 -n 4 -s 60". Overall, we performed 2,400 rigid-body docking computations for this example.

All the rigid-body cross-docking computations were performed using Hex 8 package [4, 5]. We used the maximum polynomial expansion order of 31, the complete angular sampling, the range of radial search of 40 Å, only the shape-complementarity scoring function, and the final number of solutions was set to 2,000. These were clustered with the threshold of 4 Å, with the maximum allowed number of clustered to be 1,000. Each computation took about 1 minute on a Linux desktop equipped with NVIDIA GeForce GTX 680 graphics card.

In the 1ibr example, Hex did not find acceptable docking solutions (with the ligand-RMSD $<$ 10 Å) for the initial docking setup of the bound receptor and the unbound ligand. However, among all the cross-docking experiments (100 in total), Hex could find acceptable solutions in 43 runs with the best solution rank of 10, and the total number of 603 hits. Moreover, in 3 runs it could find average-quality solutions (with the ligand-RMSD $<$ 5 Å) with the best solution rank of 56, and the total number of 48 hits.

In the 1zli example, Hex found 17 acceptable docking solutions (with the ligand-RMSD $<$ 10 Å) for the initial docking setup of the bound receptor and the unbound ligand. The best rank of the hit was 23. Among all the cross-docking experiments (100 in total), Hex could find acceptable solutions in 66 runs. Multiple runs had the best solution rank of 1, and the total number of hits was 1635. Also, in 4 runs it could find average-quality solutions (with the ligand-RMSD $<$ 5 Å) with the best solution rank of 40, and the total number of 17 hits.

In the 2i9b example, Hex did not find acceptable docking solutions (with the ligand-RMSD $<$ 10 Å) for the initial docking setup of the unbound receptor and ligand. However, among all the cross-docking results, Hex could find acceptable solutions in 36 runs with the best solution rank of 7, and the total number of 211 hits. Moreover, in 2 runs it could find average-quality solutions (with the ligand-RMSD $<$ 5 Å) with the best solution rank of 243, and the total number of 4 hits. This example clearly demonstrates that it is possible to enrich the number of hits for the subsequent rescoring even for very challenging flexible docking examples. However, a more sophisticated scoring function should be used in order to select the hits during the rescoring stage.

Flexible docking

As mentioned in the main text, to lead the flexible docking experiments, we took advantage of an unpublished work [6]. In collaboration with a team of the University of Luxembourg, we developed a flexible docking algorithm using collective motions and an evolutionary-based search algorithm. To be more precise, the evolutionary algorithm is an extension of the Genetic Algorithm, a Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [7].

This algorithm uses mechanisms such as mutation, recombination (crossover) and selection that are inspired from biological evolution. It first randomly generates a "population" of solutions. Then it evaluates each "individual" of the population. More precisely, it computes its fitness, which is composed of two terms, the rigid-body energy and the flexible energy. The rigid-body energy is based on the knowledge-based potential

developed in our previous works [8, 9], where it is fully described. The flexible energy term is the deformation energy of the molecular structure along the normal modes. This energy term is weighted by the corresponding frequencies. Through the iterations, the population increases as in sexual reproduction through cross-over, and is the subject of possible mutations. Before the end of the iterations, solutions are kept so that to describe the Pareto frontier. The later was helpful in assigning weights to different contributions in the total energy.

Finally, at the end of the search, a clustering is performed to extract putative near-native conformations that could be further refined. The clustering is based on the ligand-RMSD measure between all computed conformations. A new cluster is set if a conformation is distant of more than 5 Å of the already-existing cluster seeds. We should say that the clustering step can be also performed during the generation of the populations.

To implement the NSCGA-II method, we used the *jMetal* library, already mentioned in previous docking study with AutoDock [10]. The crossover is performed with a probability of 0.9 and a distribution index of 5, whereas the mutation is performed with a probability 1/ (6 + number of modes) and a distribution index of 100. A maximum of 100,000 individuals is kept.

The number of iterations of the genetic search was set to 100 and the number of initial solutions to 1000. The population evolves and increases trough cross-over but not all individuals are kept. These are chosen so that to describe the Pareto frontier and such that their number does not exceed 10,000.

Regarding the accuracy of the predictions, from the list of found clusters we could select a few ones with a low ligand-RMSD (L-RMSD) and interface RMSD (I-RMSD) to the known solution. For example, for the 1ibr example, the best cluster had L-RMSD of 7.5 Å and I-RMSD of 3.3 Å. Table 1 from the main text lists results for all the examples. These can be classified as acceptable quality predictions, which is a good result for highly flexible proteins.

MD clustering

Clustering of molecular dynamics trajectories was performed using the NOLB NMA method available at `https://team.inria.fr/nano-d/software/nolb-normal-modes/` [1]. First, we superposed the trajectory frames with the first frame. Then, we computed the position covariance matrix. Finally, we partially diagonalized it to determine a certain number of the principal components, which correspond to the largest eigenvalues of the covariance matrix. Once the principal components are computed, we can project the trajectory frames into these and then use the projection coefficients to rapidly determine the RMSD of flexible deformation using equation 15 from the main text. We should specifically mention that a few principal components very often cannot fully describe the dynamics of the molecular system. However, they represent the *essential* dynamics that can be used for practical applications, such as clustering of MD trajectory frames. Table 1 lists the results of the clustering computational experiment with a lysozyme MD trajectory. One can see that even for a such small protein, with only 1960 atoms, the RapidRMSD-based clustering method very often significantly outperforms the standard algorithm. These experiments were done using the following command, "NOLB input.pdb --dcd input.dcd --clust --rmsd RMSD", where the trajectories are currently supported in two formats, dcd, and multi-model pdb, and where RMSD is the clustering threshold value in Angstroms.

Generation of pseudo-random structural ensembles

A practical demonstration of the constant-RMSD pseudo-random structural ensembles approach can be found at `https://team.inria.fr/nano-d/software/nolb-normal-modes/`, where the collective motions are computed with the NOLB NMA method [1]. For example, to create a constant-RMSD ensemble of the 1ibr ligand at 3 Å amplitude deformation form the starting structure using the 10 lowest-frequency modes, one can type the following, "NOLB 1IBR_l_u.pdb -r 3 -n 10 -s 100". This will generate a multi-model pdb

RMSD threshold	Standard time	RapidRMSD time	Number of Clusters	Number of RapidRMSD clusters
0.1 Å	380 s	1.00 s	10,000	9,803
0.25 Å	380 s	0.61 s	10,000	2,900
0.5 Å	380 s	0.53 s	10,000	492
0.75 Å	127 s	0.52 s	3,210	140
1.0 Å	31.8 s	0.51 s	748	57
1.25 Å	9.2 s	0.51 s	233	21
1.5 Å	3.4 s	0.50 s	80	16
1.75 Å	0.85 s	0.50 s	25	9
2.0 Å	0.24 s	0.50 s	10	7

Table 1: Comparison of RapidRMSD clustering of MD trajectories with a standard method. MD trajectory of lysozyme (1960 atoms) consisting of 10,000 frames was used. $M = 10$ PCA components were computed. Additional time for the construction of the covariance matrix and its partial diagonalization was constant and equal to 8.02 seconds. The RapidRMSD timing includes the time required to compute projection coefficients of each frame on the chosen principal components. Therefore, this does not reduce for large clustering threshold values.

file with pseudo-random 100 models.

Corresponding Author : Sergei Grudinin, NANO-D, INRIA Rhone-Alpes Research Center Minatec Campus 17 rue des Martyrs 38054 Grenoble France. Phone: +33 4 38 78 16 91. E-mail: sergei.grudinin@inria.fr.

References

[1] Alexandre Hoffmann and Sergei Grudinin. Nolb: Nonlinear rigid block normal-mode analysis method. *Journal of chemical theory and computation*, 13(5):2123–2134, 2017.

[2] Philippe Durand, Georges Trinquier, and Yves Henri Sanejouand. A new approach for determining low-frequency normal modes in macromolecules. *Biopolymers*, 34(6):759–771, jun 1994.

[3] Florence Tama, Florent Xavier Gadea, Osni Marques, and Yves-Henri Sanejouand. Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins: Struct., Funct., Bioinf.*, 41(1):1–7, 2000.

[4] David W Ritchie and Graham J L Kemp. Protein docking using spherical polar fourier correlations. *Proteins: Struct., Funct., Bioinf.*, 39(2):178–194, 2000.

[5] D. W. Ritchie, D. Kozakov, and S. Vajda. Accelerating and focusing protein-protein docking correlations using multi-dimensional rotational FFT generating functions. *Bioinformatics*, 24(17):1865–1873, jun 2008.

[6] Angelo Migliosi, Xavier Besseron, Emilie Neveu, Sergei Grudinin, Gregoire Danoy, and Pascal. Bouvry. Flexible Protein Docking. Technical report, 2015.

[7] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] Petr Popov and Sergei Grudinin. Knowledge of native protein–protein interfaces is sufficient to construct predictive models for the selection of binding candidates. *Journal of chemical information and modeling*, 55(10):2242–2255, 2015.

[9] Emilie Neveu, David W. Ritchie, Petr Popov, and Sergei Grudinin. PEPSI-Dock: A detailed data-driven protein-protein interaction potential accelerated by polar Fourier correlation. *Bioinformatics*, 32(17):i693–i701, 2016.

[10] Esteban López-Camacho, María Jesús García Godoy, Antonio J. Nebro, and José F. Aldana-Montes. JMetalCpp: Optimizing molecular docking problems with a C++ metaheuristic framework. *Bioinformatics*, 30(3):437–438, 2014.