HAL
archives-ouvertes.fr

RapidRMSD: Rapid determination of RMSDs corresponding to motions of flexible molecules

Emilie Neveu, Petr Popov, Alexandre Hoffmann, Angelo Migliosi, Xavier Besseron, Gregoire Danoy, Pascal Bouvry, Sergei Grudinin

▶ **To cite this version:**

Emilie Neveu, Petr Popov, Alexandre Hoffmann, Angelo Migliosi, Xavier Besseron, et al.. RapidRMSD: Rapid determination of RMSDs corresponding to motions of flexible molecules. Bioinformatics, Oxford University Press (OUP), 2018, 34 (16), pp.2757-2765. 10.1093/bioinformatics/bty160 . hal-01735214

HAL Id: hal-01735214
https://hal.inria.fr/hal-01735214

Submitted on 19 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

RapidRMSD : Rapid determination of RMSDs corresponding to motions of flexible molecules.

Emilie Neveu [1,2], Petr Popov [1, 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, 6, rue Richard Coudenhove-Kalergi, L-1359, Luxembourg.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: The root mean square deviation (RMSD) is one of the most used similarity criteria in structural biology and bioinformatics. Standard computation of the RMSD has a linear complexity with respect to the number of atoms in a molecule, making RMSD calculations time-consuming for the large-scale modeling applications, such as assessment of molecular docking predictions or clustering of spatially proximate molecular conformations. Previously we introduced the *RigidRMSD* algorithm to compute the RMSD corresponding to the rigid-body motion of a molecule. In this study we go beyond the limits of the rigid-body approximation by taking into account conformational flexibility of the molecule. We model the flexibility with a reduced set of collective motions computed with e.g. normal modes or principal component analysis.
Results: The initialization of our algorithm is linear in the number of atoms and all the subsequent evaluations of RMSD values between flexible molecular conformations depend only on the number of collective motions that are selected to model the flexibility. Therefore, our algorithm is much faster compared to the standard RMSD computation for large-scale modeling applications. We demonstrate the efficiency of our method on several clustering examples, including clustering of flexible docking results and molecular dynamics (MD) trajectories. We also demonstrate how to use the presented formalism to generate pseudo-random constant-RMSD structural molecular ensembles and how to use these in cross-docking.
Availability: We provide the algorithm written in C++ as the open-source *RapidRMSD* library governed by the BSD-compatible license, which is available at `http://team.inria.fr/nano-d/software/RapidRMSD/`. The constant-RMSD structural ensemble application and clustering of MD trajectories is available at `http://team.inria.fr/nano-d/software/nolb-normal-modes/`.
Contact: sergei.grudinin@inria.fr
Supplementary information: Supplementary data are available at *Bioinformatics*

1 Introduction

With the constant growth of computing and experimental resources, computational biology and bioinformatics face big collections of data, such as large datasets of conformations of molecules that have been determined either experimentally or using computer-aided simulations. Dealing with large lists of molecular conformations requires the development of numerically efficient algorithms.

One of the most widely accepted characteristics when analyzing biological structures is the Root Mean Square Deviation (RMSD), a similarity metric between molecular conformations, or more generally, two ordered sets of points. RMSD is commonly used for classification and comparison of multiple structures when searching for similar proteins in a database (Salem *et al.*, 2010; Ye and Godzik, 2003; Emekli *et al.*, 2008; Shatsky M, 2002; Shatsky *et al.*, 2004; Magis *et al.*, 2013; Holm and Sander, 1993), but also when studying computer-simulated structures. In docking benchmarks and assessment experiments, such as the Critical Assessment of PRediction of Interactions (Janin, 2005; Méndez *et al.*, 2003), indicators such as the fraction of native and non-native contacts estimate the biological quality of the predictions, while RMSD from a reference solution measures the geometrical quality of a putative binding pose. Particularly, RMSD computations help to compare molecular structures and to reduce a large list of predicted molecular conformations to

© The Author 2015. Published by Oxford University Press. All rights reserved. For permissions, please e-mail: journals.permissions@oup.com

a smaller set of clustered solutions. For example, clustering is particularly useful to identify and characterize binding sites (Kozakov *et al.*, 2005; Comeau and Camacho, 2005; Popov *et al.*, 2014; Huang, 2014) or to discover near-native docking poses (Zhang and Skolnick, 2004; Kozakov *et al.*, 2005; Lorenzen and Zhang, 2007) Another application of the RMSD computations is a generation of constant-RMSD structural ensembles. These are increasingly used to build training sets for data-driven biochemical models (Popov and Grudinin, 2015; Rupp *et al.*, 2015; Smith *et al.*, 2017), and can also be applied to construct near-native structures for cross-docking (Cavasotto *et al.*, 2005; Mustard and Ritchie, 2005), for example.

While there has been some progress in effective RMSD computation by optimal superposition of conformations (Horn, 1987; Diamond, 1988; Kearsley, 1989; Kneller, 1991; Coutsias *et al.*, 2004; Theobald, 2005) there has not been so much done for computing RMSDs between molecules coupled to a fixed reference frame (Popov and Grudinin, 2014; Hildebrandt *et al.*, 2014), which is very useful, for example, when clustering molecular docking solutions (Kozakov *et al.*, 2005; Comeau and Camacho, 2005; Popov *et al.*, 2014). Also, RMSD computation is an essential step in the analysis of ensembles of molecular conformations (Gil and Guallar, 2014; Hung and Samudrala, 2014), which might require billions of RMSD computations when doing a pairwise comparison of the predicted structures. The growing number of required computations led us to develop an efficient way to compute RMSDs in the specific case of rigid-body motions (Popov and Grudinin, 2014). Here, we present an extension of our previous work to flexible molecules, with flexibility being represented as an *affine deformation* of the original structure. More technically, we describe the flexibility with a finite number of linear collective motions that can be computed, for example, with Normal Mode Analysis (NMA) (Wilson, 1955) or Principal Component Analysis (PCA). While being computationally affordable, this type of collective motions also often captures the unbound to bound protein transitions (Hinsen, 1998; Kovacs *et al.*, 2004; Dobbins *et al.*, 2008). Thus, this is typically the approach of choice when dealing with the challenges of flexible protein-protein docking (Zacharias, 2010). The capability of NMA- and PCA-based principal motions to model large conformational changes of the protein backbone at a low computational cost enables their use at all the stages of protein-protein docking protocols, e.g., in the generation of ensembles for cross-docking (Cavasotto *et al.*, 2005; Mustard and Ritchie, 2005), in the prediction of hinge regions (Emekli *et al.*, 2008; Schneidman-Duhovny *et al.*, 2007), during the conformational sampling (Moal and Bates, 2010; Fiorucci and Zacharias, 2010; May and Zacharias, 2008), or when refining the solutions (Maschiach *et al.*, 2010; Venkatraman and Ritchie, 2012; Lindahl and Delarue, 2005).

As we recently demonstrated for the rigid-body motion case, RMSD computation can be split into two parts, initialization, which is linear in the number of atoms in the molecule, and the RMSD calculation itself, which takes only a constant number of arithmetic operations (Popov and Grudinin, 2014, 2018). This allows a dramatic reduction of the computational cost. For example, a rigid-body clustering requiring $O(N \times D)$ operations will be solved in $O(N + D)$ operations, with N being the number of atoms, and D being the number of pairwise structure comparisons required by the clustering. Having N usually greater than 1000, the algorithm typically results in at least one order of magnitude computational speed-up compared to a standard clustering method. In this study we relax the rigid-body approximation by taking into account molecular flexibility described with a linear transformation of the original structure (Brooks and Karplus, 1983). A linear, or, more generally, affine transformation can be represented as a weighted sum of orthogonal vectors, computed with, e.g., PCA or NMA methods. Below we demonstrate that we can cluster putative binding poses produced by flexible docking algorithms significantly faster than a standard approach, given that most of the flexible docking methods use only a small number of collective motions (Emekli *et al.*, 2008; Moal and Bates, 2010; Fiorucci and Zacharias, 2010; Venkatraman and Ritchie, 2012; May and Zacharias, 2008; Schneidman-Duhovny *et al.*, 2007).

The rest of the article is organized as follows. First, we summarize our approach and our previous results for fast RMSD computations in the rigid-body motion case (Popov and Grudinin, 2014). This is followed by the derivation of equations for an efficient computation of RMSD between molecular conformations using the corresponding rigid and flexible transforms. Finally, we demonstrate the superiority of our algorithm over the standard one with several practical examples.

The algorithm is implemented as an open-source *RapidRMSD* library, available at `https://team.inria.fr/nano-d/software/RapidRMSD/`. To guide the users, the library also provides a number of source-code examples that demonstrate its usage in different scenarios.

2 Approach

Using the rigid-body motion formalism extended with collective motions, here we present an efficient approach to compute RMSD between flexible molecular conformations. This approach expresses the RMSD according to the spatial transformation operators. These are the rotation, translation and a set of vectors that describe the collective motions, which can be the main vibrational modes of a molecule, for example. Our method comprises the initialization step followed by a set of RMSD calculation steps.The former computes the internal geometric properties of the reference molecule such as the inertia tensor. Performed only once, it has a linear complexity with the number of atoms in the molecule and at worst a quadratic complexity with the number of vectors describing the collective motions. Each of the latter steps – one per a pair of conformations – takes only constant time with respect to the number of atoms. Our approach is particularly useful when applied to a large set of conformations of a flexible molecule, for instance as it happens in clustering applications or when generating near-native structural ensembles. In the former case, our algorithm has a linear cost with the number of collective motion vectors, which makes it extremely fast, since usually only an order of ten collective motions is sufficient to accurately describe the global flexibility of a macromolecule - no matter how big it is.

3 Methods

3.1 Theoretical foundation

3.1.1 Weighted RMSD

Let a molecule be defined by N atoms at positions $A = \{\mathbf{a}_i\}_N$ with coordinates $\mathbf{a}_i = \{x_i, y_i, z_i\}^T$ and associated weights $w = \{w_i\}_N$. Given two sets of N points, A and A', of respective coordinates $\{\mathbf{a}_i\}_N$ and $\{\mathbf{a}'_i\}_N$, describing two conformations of a molecule, we can define the weighted RMSD between them as

$$\mathrm{RMSD}(A, A')^2 = \frac{1}{W} \sum_i w_i \left|\mathbf{a}_i - \mathbf{a}'_i\right|^2, \qquad (1)$$

where $W = \sum_i w_i$. Statistical weights $\{w_i\}_N$ may emphasize the importance of a certain part of the structure, for example in case of a protein, the backbone or the side chains. More commonly, these weights will be equal to the atomic masses (in this case W equals to the total mass of the molecule) or may be set to 1 (in this case $W = N$).

3.1.2 Quaternion arithmetic

A quaternion Q can be represented as a combination of a scalar s with a 3-component vector $\mathbf{q} = \{q_x, q_y, q_z\}^T$, $Q = [s, \mathbf{q}]$. This is a compact

representation of a spatial transformation operator, particularly suited for rotations. For example, a rotation operator can be described with a rotation quaternion $\hat{Q}$, which has a unit norm. Generally, quaternion algebra defines multiplication, division, inversion and norm, among other operations. A short summary of quaternion arithmetics can be found in our previous paper (Popov and Grudinin, 2014).

3.1.3 The rigid-body motion case

Here we summarize the principal result of our previous work (Popov and Grudinin, 2014). Let us assume that a molecule A with coordinates $\mathbf{a}_i = \{x_i, y_i, z_i\}^T$ is translated and rotated to new positions $A' = \{\mathbf{a}'_i\}_N$, which are given as $\mathbf{a}'_i = \mathbf{R}\mathbf{a}_i + \mathbf{T}$. Here $\mathbf{R}$ is the 3×3 rotation matrix and $\mathbf{T}$ is the translation 3-vector. It is convenient to use the quaternion representation $Q = [s, \mathbf{q}]$ of the rotation matrix $\mathbf{R}$. Then, the weighted RMSD between A, the original positions, and A', the transformed positions, can be written according to Eq. 4 from (Popov and Grudinin, 2014) as

$$\text{RMSD}^2 = \mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} + 2\mathbf{T}^T\left(\mathbf{R} - \mathbf{E}_3\right)\mathbf{C}, \tag{2}$$

with $\mathbf{E}_3$ being the 3×3 identity matrix, $\mathbf{C}$ the center of mass $\frac{1}{W}\{\sum w_i x_i, \sum w_i y_i, \sum w_i z_i\}^T$ and $\mathbf{I}$ the inertia tensor:

$$\mathbf{I} = \begin{pmatrix} \sum w_i(y_i^2 + z_i^2) & -\sum w_i x_i y_i & -\sum w_i x_i z_i \\ -\sum w_i x_i y_i & \sum w_i(x_i^2 + z_i^2) & -\sum w_i y_i z_i \\ -\sum w_i x_i z_i & -\sum w_i y_i z_i & \sum w_i(x_i^2 + y_i^2) \end{pmatrix}. \tag{3}$$

We should mention that it is practical to work in the center-of-mass reference frame where $\mathbf{C} = \mathbf{0}$.

Thus, in this frame the RMSD can be expressed with fewer arithmetic operations as

$$\text{RMSD}^2 = \mathbf{T}^2_{\text{COM}} + \frac{4}{W}\mathbf{q}^T\mathbf{I}_{\text{COM}}\mathbf{q}. \tag{4}$$

3.1.4 RMSD for flexible molecules modeled with collective motions

We extend our previous work (Popov and Grudinin, 2014) by adding molecular flexibility via *linear collective motions*. These can be computed using, for example, the normal mode analysis or the principal component analysis techniques. More precisely, let $\{\mathbf{f}_i^j\}_N^M$ be a set of M vectors that describe the linear collective motions applied to a molecule, where $\mathbf{f}_i^j = \{f_{ix}^j, f_{iy}^j, f_{iz}^j\}^T$. Let i be the atom index ranging from 1 to N, and j be the index of the collective motions ranging from 1 to M. Let $\{\mu^j\}^M$ be the amplitudes of the collective motions for the *reference conformation* of the molecule. Then the reference *flexible coordinates* $A^F = \{\mathbf{a}_i^F\}_N$ are given as $\mathbf{a}_i^F = \mathbf{a}_i + \sum_{j=1}^M \mu^j \mathbf{f}_i^j$. Now, to compute the flexible coordinates of the *target conformation* A'^F, we first add flexible displacements to the *rigid coordinates* of the reference conformation and then apply to the result the rigid-body transformation. Let $\{\lambda^j\}^M$ be the amplitudes of the collective motions for the *target conformation* of the molecule, where the collective motion vectors are the same as in the reference conformation. Similarly to the rigid-body case, let $\mathbf{R}$ and $\mathbf{T}$ be the rotation matrix and the translation vector of the target rigid-body transformation, respectively, so that the flexible target coordinates $\{\mathbf{a}_i'^F\}_N$ are given as $\mathbf{a}_i'^F = \mathbf{R}\left(\mathbf{a}_i + \sum_{j=1}^M \lambda^j \mathbf{f}_i^j\right) + \mathbf{T}$. Then, the weighted RMSD between positions A^F and A'^F is given as

$$\text{RMSD}^2(A^F, A'^F) = \frac{1}{W}\sum_i w_i \left| \mathbf{a}_i + \sum_j \mu^j \mathbf{f}_i^j - \mathbf{R}\left(\mathbf{a}_i + \sum_j \lambda^j \mathbf{f}_i^j\right) - \mathbf{T} \right|^2. \tag{5}$$

We can rewrite the previous expression using the quaternion representation of vectors $\mathbf{a}_i$, $\mathbf{T}$, and the rotation matrix $\mathbf{R}$ as

$$\text{RMSD}^2 = \frac{1}{W}\sum_i w_i \left| [0, \mathbf{a}_i + \sum_j \mu^j \mathbf{f}_i^j] - \hat{Q}[0, \mathbf{a}_i + \sum_j \lambda^j \mathbf{f}_i^j]\hat{Q}^{-1} - [0, \mathbf{T}] \right|^2. \tag{6}$$

Here, the unit quaternion $\hat{Q}$ corresponds to the rotation matrix $\mathbf{R}$. Since the norm of a quaternion does not change if we multiply it by a unit quaternion, we may right-multiply the kernel of the previous expression by $\hat{Q}$ to obtain

$$\text{RMSD}^2 = \frac{1}{W}\sum_i w_i \left| [0, \mathbf{a}_i + \sum_j \mu^j \mathbf{f}_i^j]\hat{Q} - \hat{Q}[0, a_i + \sum_j \lambda^j \mathbf{f}_i^j] - [0, \mathbf{T}]\hat{Q} \right|^2. \tag{7}$$

Using the scalar–vector representation of a quaternion, $\hat{Q} = [s, \mathbf{q}]$, we rewrite the previous RMSD expression as

$$\begin{aligned} \text{RMSD}^2 = & \frac{1}{W}\sum_i w_i \left[\mathbf{q} \cdot \left(\mathbf{T} + \sum_j \lambda^j \mathbf{f}_i^j - \sum_j \mu^j \mathbf{f}_i^j \right) \right. \\ & - s\left(\mathbf{T} + \sum_j \lambda^j \mathbf{f}_i^j - \sum_j \mu^j \mathbf{f}_i^j \right) \\ & \left. + (2\mathbf{a}_i - T + \sum_j \mu^j \mathbf{f}_i^j + \sum_j \lambda^j \mathbf{f}_i^j) \times \mathbf{q} \right]^2 \end{aligned} \tag{8}$$

Performing scalar and vector products in Eq. (8), then grouping the terms that depend on atomic positions together, and after introducing the inertia tensor $\mathbf{I}$, the center of mass vector $\mathbf{C}$ and reintroducing the rotation matrix $\mathbf{R}$, we obtain

$$\begin{aligned} \text{RMSD}^2 = & \mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} + 2\mathbf{T}^T\left(\mathbf{R} - \mathbf{E}_3\right)\mathbf{C} \\ & -2\sum_j \mu^j \mathbf{T}^T \mathbf{B}^j + \sum_{jk} \mu^j \mu^k Tr\left(\mathbf{F}^{jk}\right) \\ & -2\sum_j \mu^j Tr\left(\left(\mathbf{R} - \mathbf{E}_3\right)^T \mathbf{D}^j\right) \\ & +2\sum_j \lambda^j \mathbf{T}^T \mathbf{R}\mathbf{B}^j + \sum_{jk} \lambda^j \lambda^k Tr\left(\mathbf{F}^{jk}\right) \\ & -2\sum_j \lambda^j Tr\left(\left(\mathbf{R} - \mathbf{E}_3\right)\mathbf{D}^j\right) - 2\sum_j \sum_k \lambda^j \mu^k Tr\left(\mathbf{R}\mathbf{F}^{jk}\right). \end{aligned} \tag{9}$$

$$\begin{aligned} \text{RMSD}^2 = & \frac{1}{W}\sum_i w_i \left[\mathbf{q} \cdot \left(\mathbf{T} + \sum_j \lambda^j \mathbf{f}_i^j - \sum_j \mu^j \mathbf{f}_i^j \right) \right. \\ & - s\left(\mathbf{T} + \sum_j \lambda^j \mathbf{f}_i^j - \sum_j \mu^j \mathbf{f}_i^j \right) \\ & \left. + (2\mathbf{a}_i - T + \sum_j \mu^j \mathbf{f}_i^j + \sum_j \lambda^j \mathbf{f}_i^j) \times \mathbf{q} \right]^2 \end{aligned} \tag{10}$$

Here, $Tr()$ is the matrix trace operator, $\mathbf{D}^j$ is the set of M 3×3 matrices of cross-products

$$\mathbf{D}^j = \frac{1}{W}\begin{pmatrix} \sum w_i x_i f_{ix}^j & \sum w_i y_i f_{ix}^j & \sum w_i z_i f_{ix}^j \\ \sum w_i x_i f_{iy}^j & \sum w_i y_i f_{iy}^j & \sum w_i z_i f_{iy}^j \\ \sum w_i x_i f_{iz}^j & \sum w_i y_i f_{iz}^j & \sum w_i z_i f_{iz}^j \end{pmatrix}, \tag{11}$$

$\mathbf{F}^{jk}$ is the set of M^2 3×3 matrices of cross-products

$$\mathbf{F}^{jk} = \frac{1}{W}\begin{pmatrix} \sum w_i f_{ix}^k f_{ix}^j & \sum w_i f_{iy}^k f_{ix}^j & \sum w_i f_{iz}^k f_{ix}^j \\ \sum w_i f_{ix}^k f_{iy}^j & \sum w_i f_{iy}^k f_{iy}^j & \sum w_i f_{iz}^k f_{iy}^j \\ \sum w_i f_{ix}^k f_{iz}^j & \sum w_i f_{iy}^k f_{iz}^j & \sum w_i f_{iz}^k f_{iz}^j \end{pmatrix}, \tag{12}$$

and $\mathbf{B}^j = \frac{1}{W}\left\{\sum w_i f_{ix}^j, \sum w_i f_{iy}^j, \sum w_i f_{iz}^j\right\}^T$ are the centers of the collective motions. Again, it is practical to choose the reference frame of the target molecule such that $\mathbf{C} = \mathbf{0}$. Also, commonly used collective motions, e.g. those computed using the normal mode analysis or the principal component analysis, possess the weight-orthonormality property, i.e., $Tr\left(\mathbf{F}^{jk}\right) = \delta_{jk}/W$. From now on, we will consider only this type of motions. Thus, in this case, the RMSD equation simplifies to

$$\begin{aligned}\text{RMSD}^2 =&\mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} + \frac{1}{W}\sum_j\left(\mu^{j\,2} + \lambda^{j\,2}\right)\\ &-2\sum_j \mu^j \mathbf{T}^T\mathbf{B}^j - 2\sum_j \mu^j Tr\left(\left(\mathbf{R}-\mathbf{E}_3\right)^T\mathbf{D}^j\right)\\ &+2\sum_j \lambda^j \mathbf{T}^T\mathbf{R}\mathbf{B}^j - 2\sum_j \lambda^j Tr\left(\left(\mathbf{R}-\mathbf{E}_3\right)\mathbf{D}^j\right)\\ &-2\sum_{jk}\lambda^j\mu^k Tr\left(\mathbf{R}\mathbf{F}^{jk}\right).\end{aligned} \tag{13}$$

Equation (13) is our *master RMSD equation* and the principal result of this work. It consists of the rigid contribution $\mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q}$, the flexible contribution $\frac{1}{W}\sum_j\left(\mu^{j\,2} + \lambda^{j\,2}\right)$, and the cross–terms. Once the matrices and vectors $\mathbf{I}$, $\mathbf{B}^j$, $\mathbf{D}^j$, and $\mathbf{F}^{jk}$ that depend on the number of atoms and on the number of collective motions are computed, the calculation of RMSD takes time independent of the number of atoms and is at most quadratic with the number of collective motions. Below we will explicitly consider several special cases of simplified motions that also simplify the *master equation* (13) and reduce its computational cost.

3.1.5 RMSD corresponding to a rigid reference conformation

A practical consequence of the *master equation* (13) is the expression of RMSD for a flexible target conformation with respect to a rigid reference conformation. In this case, all the amplitudes of the collective motions for the reference conformation $\{\mu^j\}^M$ are zero, and the RMSD expression reduces to

$$\begin{aligned}\text{RMSD}^2 =&\mathbf{T}^2 + \frac{4}{W}\mathbf{q}^T\mathbf{I}\mathbf{q} + \frac{1}{W}\sum_j\lambda^{j\,2}\\ &+2\sum_j \lambda^j \mathbf{T}^T\mathbf{R}\mathbf{B}^j - 2\sum_j \lambda^j Tr\left(\left(\mathbf{R}-\mathbf{E}_3\right)\mathbf{D}^j\right).\end{aligned} \tag{14}$$

In this case, the calculation of one RMSD takes linear time with the number of collective motions M.

3.1.6 RMSD corresponding to a pure flexible motion

When studying only flexible movements of a molecule, which can be the case when refining the docking poses or when generating pseudo-random structural ensembles, the *master equation* (13) reduces to

$$\text{RMSD}^2 = \frac{1}{W}\sum_j\left(\mu^j - \lambda^j\right)^2. \tag{15}$$

Indeed, in this case, the rotation matrix $\mathbf{R}$ is identity and the translation vector $\mathbf{T}$ is zero. Thus, the RMSD expression becomes linear in the number of collective motions M.

3.1.7 RMSD corresponding to the relative rigid-body motion

The RMSD *master equation* (13) can be also adapted for the particularly useful clustering application case. Here, one compares two possible *target* conformations, for which the transformation operators are defined with respect to the original *reference* conformation. Let $A = \{\mathbf{a}_i\}_N$ define the reference coordinates and $A_1 = \{\mathbf{a}_i^{(1)}\}_N$ and $A_2 = \{\mathbf{a}_i^{(2)}\}_N$ be the two target conformations we want to compare. Let set $\{\lambda^j\}^M$ define the collective motion amplitudes of A_1, and set $\{\mu^j\}^M$ define the ones of A_2. Let also $\mathbf{R}_1$ and $\mathbf{T}_1$ describe the rigid-body transformation applied to A to obtain A_1, and $\mathbf{R}_2$ and $\mathbf{T}_2$ be the rigid-body transformation applied to A to obtain A_2. Finally, let a unit quaternion $[s_{12}, \mathbf{q}_{12}]$ correspond to the relative rotation $\mathbf{R}_{12} \equiv \mathbf{R}_2^T\mathbf{R}_1$ and let the relative translation be $\mathbf{T}_{12} \equiv \mathbf{R}_1^T(\mathbf{T}_2 - \mathbf{T}_1)$. Then, the weighted RMSD between A_1 and A_2 is given by a generalized version of the *master equation* (13) as

$$\begin{aligned}\text{RMSD}^2 =&(\mathbf{T}_2 - \mathbf{T}_1)^2 + \frac{4}{W}\mathbf{q}_{12}^T\mathbf{I}\mathbf{q}_{12} + \frac{1}{W}\sum_j\left(\mu^{j\,2} + \lambda^{j\,2}\right)\\ &-2\sum_j \mu^j \mathbf{T}_{12}^T\mathbf{B}^j - 2\sum_j \mu^j Tr\left(\left(\mathbf{R}_r-\mathbf{E}_3\right)^T\mathbf{D}^j\right)\\ &+2\sum_j \lambda^j \mathbf{T}_{12}^T\mathbf{R}_{12}\mathbf{B}^j - 2\sum_j \lambda^j Tr\left(\left(\mathbf{R}_{12}-\mathbf{E}_3\right)\mathbf{D}^j\right)\\ &-2\sum_{jk}\lambda^j\mu^k Tr\left(\mathbf{R}_{12}\mathbf{F}^{jk}\right).\end{aligned} \tag{16}$$

3.2 Algorithm Implementation

3.2.1 Computational considerations

In the above equations (9–16) two variables depend solely on the atomic positions of the reference molecular structure: the inertia tensor $\mathbf{I}$ and the center of mass vector $\mathbf{C}$, whereas matrices $\mathbf{F}^{ij}$ and vectors $\mathbf{B}^j$ depend solely on the collective motion vectors, and matrices $\mathbf{D}^j$ depend both on the atomic positions of the reference structure and the motion vectors. Therefore, given a set of D spatial transformations, we can compute these variables only once at the beginning and define this as the initialization step. The computational complexity of this step is linear in N, the number of atoms in the molecule, and quadratic in M, the number of collective motion vectors.

Following this initialization, each RMSD computation takes linear or quadratic time in the number of collective motions M, depending on whether the reference molecule is rigid or flexible. The total cost to compute D RMSD values for a molecule with N atoms and M collective motion vectors thus will be at most $O(N+DM^2)$, which is typically much smaller compared to the cost of standard algorithms, $O(DN)$, particularly at large values of D and N with M^2 usually smaller than N.

3.2.2 Numerical tests

The complexity of the different algorithms presented in the paper are based on the arithmetic operations appearing in the equations. The effective computational cost will depend on a particular hardware architecture. However, we can focus on the speed-up ratio, which must be reproducible on different computers and/or with different compilers. In the following section, we implemented the tests using the C++ programming language and compiled them using the clang++ compiler version 6.0. We ran the tests on a 64-bit MacOs operating system with a 2,6 GHz Intel Core i7 processor.

3.3 Results and Discussion

This section presents numerical tests with several practical applications of the equations derived above. One of the practical ways to compute collective motions for molecular systems is the NMA. In all the demonstrations below we will use the NOLB NMA approach to compute some number of the lowest-frequency linear normal modes (Hoffmann and Grudinin, 2017). This is a particularly efficient implementation of NMA, if only a few normal modes are required. Please see SI for more details. For the docking test cases we have chosen some examples from the Protein Docking Benchmark v5 (Vreven *et al.*, 2015). This benchmark lists protein

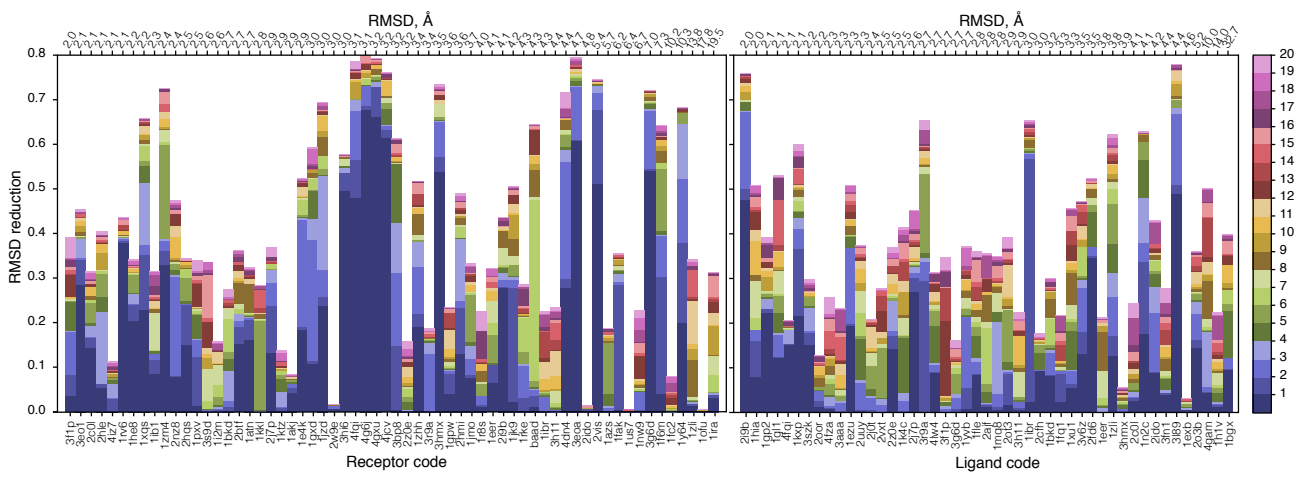

Fig. 1. Transitions between the unbound (u) and bound (b) states of proteins from the Protein Docking Benchmark v5 (Vreven et al., 2015). The top x-axis shows C_α RMSD between the two states. The bottom x-axis lists the corresponding PDB codes of the complexes. The left plot shows the receptors, and the right plot shows the ligands, as labelled by the authors of the benchmark. Only structures with u-b RMSD ≥ 2 Å are shown. The y-axis shows the relative u-b transition that can be predicted using the optimal linear combination of some number of normal modes. Results for the range between 1 and 20 are shown in different colors (see the colorbar at the right). These computations were done using the NOLB NMA method (Hoffmann and Grudinin, 2017), please see SI for more details.

complexes in two states, bound (b) and unbound (u). Figure 1 shows how well lowest-frequency normal modes can predict the u-b transitions. As we can see, in some cases just a few modes are sufficient to describe more than 50% of structural transitions. However, on average, the 10 lowest modes contribute to about 32% of the u-b transitions.

We selected three complexes for our docking experiments, 1ibr (4811 atoms), 1zli (2984 atoms), and 2i9b (2998 atoms). All these are classified as highly flexible and as difficult targets. To represent the flexibility of the docking partners, we used 2 normal modes for the ligand in 1ibr, 5 for the ligand and the receptor in 2i9b, and 6 for the ligand in 1zli. We should also mention that we constrained the amplitudes of the modes so that the RMSD induced by the conformational changes of each partner does not exceed 3 Å for the 1ibr ligand, 2 Å for the 2i9b ligand, 4 Å for the 2i9b receptor, and 3.9 Å for the 1zli ligand.

3.3.1 Flexible docking using collective motions

Our first application will be a clustering algorithm that illustrates the utilization of the derived equations (16). Clustering algorithms are particularly useful in molecular docking, where similar putative binding poses are often grouped together. The goal of clustering is to reduce a large list of possibly redundant docking solutions to a smaller list of clusters, which can also be used as indicators of binding sites at the first stage of the docking pipeline. Most of the docking algorithms use pair-wise RMSD between the docking poses as the similarity metric for clustering (Kozakov *et al.*, 2005; Ritchie and Kemp, 2000; Chen *et al.*, 2003). For these types of applications our algorithm proves to be especially time-efficient since many computations of RMSD are to be performed on the same set of atoms. In our previous work we demonstrated the efficiency of the *RigidRMSD* library for the clustering docking poses of molecular complexes corresponding to rigid-body transformations. However, many docking tools, e.g. SwarmDock (Moal and Bates, 2010), ATTRACT (Fiorucci and Zacharias, 2010), EigenHex (Venkatraman and Ritchie, 2012) , HingeProt (Emekli *et al.*, 2008), FlexDock (Schneidman-Duhovny *et al.*, 2007), FiberDock (Maschiach *et al.*, 2010), etc., use collective motions in order to take into account global flexibility of a molecule. Generally, computational docking of flexible molecules can be performed in many different ways. More precisely, flexibility can be introduced implicitly in soft docking approaches (Palma *et al.*, 2000; Heifetz and Eisenstein, 2003) or ensemble docking (Grünberg *et al.*, 2004; Zavodszky *et al.*, 2004), as well as explicitly, by rotating the side chains, or using the molecular dynamics (Dominguez *et al.*, 2003). More often, however, large conformational changes are approximated with low-frequency collective motions (Fiorucci and Zacharias, 2010; Moal and Bates, 2010; Venkatraman and Ritchie, 2012; May and Zacharias, 2008). The latter can be computed, e.g., using all-atom force fields (Hinsen, 2000) or in a simpler way using, e.g., the elastic-network model (Tirion, 1996) or the Gaussian network model (Bahar *et al.*, 1997), both often combined with the rotation-translation of blocks method to reduce the dimensionality of the problem (Tama *et al.*, 2000; Hoffmann and Grudinin, 2017), etc. These motions describe equilibrium vibrations of a molecule and are selected using PCA or NMA by the diagonalization of the corresponding covariance or Hessian interaction matrices, respectively. Below, we demonstrate the use of our extended RMSD library for clustering of flexible docking poses using collective motions.

3.3.2 Clustering of randomly generated proteins

To test the efficiency of our algorithm, we first applied it to the clustering of artificially generated flexible molecular docking poses. The reference molecule is described with a $3N$-vector for the positions of the atoms and M orthogonal $3N$-vectors for the directions of the collective motions. To construct the rigid-body deformations, we then randomly generated D rigid-body transforms with the translation vector ranging from 0 to 100 Å, and randomly chose the rotations with a unit quaternion. To model molecular flexibility, we also randomly generated $D \times M$ amplitudes of flexible deformations in the range of $\{0; 100\}$.

We implemented the standard RMSD-based clustering algorithm which consists in the following steps. First, a docking prediction with the best score (yet unassigned to any cluster) is taken as the seed for the new cluster. Second, the pair-wise RMSDs between the seed and all other predictions unassigned to any cluster are measured. Third, predictions with the RMSD from the seed lower than a certain threshold are assigned to the current cluster. These steps are repeated until all docking predictions are assigned to the corresponding clusters.

The worst-case complexity of this algorithm is achieved when all clusters have unit size, i.e. there are no similar predictions and thus we result in D clusters. In this case, D^2 RMSDs have to be computed. Conversely, the

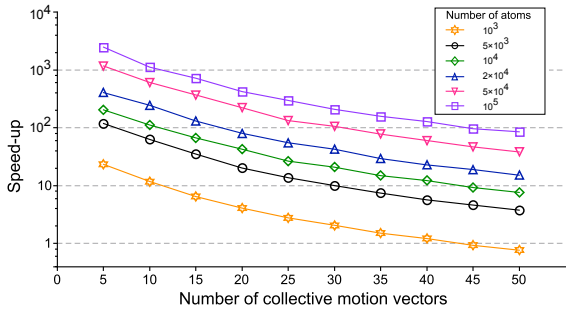

Fig. 2. Speed-up of the (i) fast RMSD method compared to the (ii) standard one (given as time(ii)/time(i)) as a function of the number of collective motion vectors in log-scale. Different curves correspond to a different number of atoms in the reference molecule.

best-case complexity is achieved when all docking predictions are similar and form a single cluster, which leads to D computations of RMSD. Thus, the number of RMSDs that are to be computed ranges from D to D^2. In order to evaluate the efficiency of our fast RMSD calculation method, we ran clustering tests for $D = 10^3$ docking candidates with various values of N and M parameters, precisely, $N \in \{10^3, 5 \times 10^3, 10^4, 2 \times 10^4, 5 \times 10^4, 10^5\}$, $M \in \{5, 10, 20, 30, 40, 50\}$. It is worth noting that, typically, flexible docking methods use very few (e.g. 20) collective motions to describe molecular flexibility (Moal and Bates, 2010; Fiorucci and Zacharias, 2010; Schneidman-Duhovny *et al.*, 2007). We adjusted the threshold RMSD to 120 Å to get a nearly uniform clustering case with approximately 28 clusters. For each of the randomly generated molecules we also randomly generated rigid and flexible deformations and grouped the obtained docking candidates with the described clustering algorithm using (*i*) the fast RMSD calculation based on the generalised *master equation* (Eq. 16), and (*ii*) the standard method (Eq. 5).

To analyze the results, we computed the speed-up of the fast method as a function to the number of collective motion vectors and the number of atoms in the reference molecule. The speed-up was computed by dividing the computational time of the standard algorithm by the computational time of the fast one. Figure 2 shows the speed-up, i.e. time(*ii*)/time(*i*) as a function of the number of collective motion vectors in log-scale. Different curves correspond to different number of atoms in the reference molecule. We observe the same general behavior for all the curves, namely, the speed-up is linear with the number of atoms in the reference molecule.

We should make an important remark regarding the fast algorithm. It becomes practically useful when $N \gg M^2$. In our experiments, this assumption does not hold in a few cases, for example when we have as few as 1,000 atoms and more than 30 collective motion vectors. However, the fast algorithm shows a significant speed-up for molecules of ten thousand of atoms and more. As we have mentioned previously, 20 lowest-frequency collective motions often provide a rather accurate description of protein flexibility at equilibrium (Moal and Bates, 2010). In this particular case, we achieve a speed-up of about 500 for a protein with a hundred thousands of atoms. For most common protein examples, the expected speed-up will be in the order of 10 to 100.

3.3.3 Rapid clustering in a flexible docking method

The second clustering experiment is an illustration of the application of RapidRMSD to a real flexible docking algorithm. We predicted docking poses using both standard and RapidRMSD clustering methods and compared the computational cost on a realistic run. We will briefly mention the quality of the best found conformations and the total timings of the docking experiments. Details about the flexible docking algorithm can be found in SI, since it is not the main focus of the current study.

Example	Standard time	RapidRMSD time	L-RMSD	I-RMSD
1ibr	393 s	335 s	7.5 Å	3.3 Å
1zli	461 s	117 s	12.1 Å	5.3 Å
2i9b	314 s	184 s	7.9 Å	3.6 Å

Table 1. Flexible docking of three difficult targets. Columns 2 and 3 present a comparison of two clustering methods, the standard one, and the rapid one. Timings are given in seconds for a MacBook Pro laptop. Columns 4 and 5 present the best clusters found by flexible docking method.

Briefly, we used an implementation of a genetic algorithm (NSGA-II) (Deb *et al.*, 2002), designed for a multi-objective optimization. The objective function was a sum of a knowledge-based interaction energy between the docking partners (Popov and Grudinin, 2015) and the energy of the flexible deformation. Starting from a randomly initialized population, we end up with a final population state, which was a subject to clustering.

Concerning the RapidRMSD library, we first initialize it when reading the input structures, and we apply it for the clustering at the end of the genetic search. After 100 iterations of 1000 genetic individuals, for the 1ibr example, 4213 solutions remained and these were grouped into 707 clusters. For the 1zli example, 4161 solutions remained and were grouped into 1167 clusters. Finally, for the 2i9b example, 4687 solutions were grouped in 1424 clusters.

Let us first compare the computational cost of the RapidRMSD and the standard clustering algorithms. Table 1 lists the computational time required for the fast and the standard clustering (in seconds) for the 3 given examples. The RapidRMSD cost includes the initialization step plus the clustering time, whereas the standard cost is given only for the clustering. RapidRMSD is faster than the standard algorithm while giving exactly the same results. On the 1zli example, we have gained about 6 minutes. If we increase the number of iterations or the number of the populations, or if we make several runs of predictions, the benefit will be even more significant. As the tested systems are relatively small, the observed clustering speed-up is lower than in the previous example.

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) values to the known solution. For example, for the 1ibr example, the best cluster has L-RMSD of 7.5 Å and I-RMSD of 3.3 Å. Table 1 lists results for all the examples. These can be classified as acceptable-quality predictions, which is a good result for highly flexible proteins.

3.3.4 Clustering of MD trajectories

Finally, using Equation 15 we constructed a fast clustering method for the analysis of MD trajectories. More precisely, we aligned the trajectory frames and projected them into the first M principal components extracted from the covariance matrix of the trajectory. Then, we implemented the clustering method described above. We tested its performance on an MD trajectory of lysozyme (1960 atoms) consisting of 10,000 frames and $M = 10$ PCA components. When the number of clusters was small compared to the length of the trajectory, the speed-up of the fast clustering method compared to the standard one was about 40. It would have been even more significant for larger proteins. However, when the number of clusters was about the length of the trajectory, the fast clustering method performed slower compared to the naive clustering method because of additional computations of the covariance matrix and the principal components. We added the fast clustering method into the NOLB NMA package (Hoffmann and Grudinin, 2017). Please see SI for more information.

3.3.5 Generation of pseudo-random structural ensembles

Another application of our method is the generation of pseudo-random structural ensembles along a few lowest vibration modes. These structural ensembles can be useful to describe molecular fluctuations at a constant temperature (Dobbins *et al.*, 2008; Kovacs *et al.*, 2004), to create inputs for cross-docking algorithms (Cavasotto *et al.*, 2005; Mustard and Ritchie, 2005) or be used in machine-learning applications to mimic non-native docking poses (Popov and Grudinin, 2015; Rupp *et al.*, 2015; Smith *et al.*, 2017). In this case, Equation 15 provides a straightforward way to adjust the amplitudes of a flexible deformation that would result in a given RMSD value. More precisely, in this application the amplitudes for the reference conformation $\{\mu^j\}^M$ are set to zero, and one only randomly selects a set of target deformation amplitudes $\{\lambda^j\}^M$. Then, these are scaled with a factor s to produce the desired value of RMSD d, where the scaling factor is adjusted according to Eq. 15 as $s = d\sqrt{W}/\sqrt{\sum_j (\lambda^j)^2}$. An implementation of this approach can be found inside the NOLB NMA method (Hoffmann and Grudinin, 2017).

3.3.6 Cross-docking with pseudo-random structural ensembles

To further demonstrate the practical applications of the pseudo-random structural ensembles, we have performed cross-docking studies of the selected protein complexes. First, we have generated random constant-RMSD docking partners, as it is described in SI. Then, we used Hex 8 rigid-body docking package (Ritchie and Kemp, 2000) to exhaustively dock all the generated docking partners. For 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. For the 1zli example, Hex found 17 acceptable docking solutions 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 best solution rank of 40, and the total number of 17 hits. Finally, for the 2i9b example, Hex did not find acceptable docking solutions for the initial docking setup of the unbound receptor and ligand. However, among all the cross-docking runs (2400 in total), 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 cases. However, a more sophisticated scoring function should be used when selecting hits during the rescoring stage.

4 Conclusion

We presented a fast and efficient algorithm that computes the RMSD between flexible molecules with the flexibility modeled by means of collective motions. These motions can be computed with the normal mode or principal component analyses and only a few lowest-frequency components are very often sufficient to describe the global flexibility of a molecule. Given this, our algorithm is much faster compared to the standard RMSD computation for the large-scale modeling applications. We implemented the algorithm as an open-source C++ library, called *RapidRMSD*, which now includes both rigid-body and flexible cases. We demonstrated the superiority of *RapidRMSD* compared to the standard RMSD computation on several clustering examples. We also proved that the analytical equations derived in this work can be used for the generation of a set of pseudo-random molecular structures with a constant RMSD from a reference molecule and that these sets can be used in cross-docking calculations. *RapidRMSD* is available at `https://team.inria.fr/nano-d/software/RapidRMSD/` or by request from the authors and the constant-RMSD structural ensemble application is available as a part of the NOLB NMA approach at `http://team.inria.fr/nano-d/software/nolb-normal-modes/`.

Acknowledgements

This work was supported by the Agence Nationale de la Recherche (ANR-11-MONU-006-01), by the Russian Science Foundation (project no. 16-14-10273), and with grant of the President of the Russian Federation (project no. MK-5279.2018.4).

References

Bahar, I., Atilgan, A. R., and Erman, B. (1997). Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Folding Des.*, **2**(3), 173–81.

Brooks, B. and Karplus, M. (1983). Harmonic dynamics of proteins: normal modes and fluctuations in bovine pancreatic trypsin inhibitor. *Proc. Natl. Acad. Sci. U.S.A.*, **80**(21), 6571–6575.

Cavasotto, C. N., Kovacs, J. A., and Abagyan, R. A. (2005). Representing receptor flexibility in ligand docking through relevant normal modes. *J. Am. Chem. Soc.*, **127**(26), 9632–9640.

Chen, R., Li, L., and Weng, Z. (2003). ZDOCK: an initial-stage protein-docking algorithm. *Proteins Struct. Funct. Bioinf.*, **52**(1), 80–87.

Comeau, S. R. and Camacho, C. J. (2005). Predicting oligomeric assemblies: N-mers a primer. *J. Struct. Biol.*, **150**(3), 233–244.

Coutsias, E. A., Seok, C., and Dill, K. A. (2004). Using quaternions to calculate RMSD. *J. Comput. Chem.*, **25**(15), 1849–1857.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput*, **6**(2), 181–197.

Diamond, R. (1988). A note on the rotational superposition problem. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, **44**(2), 211–216.

Dobbins, S. E., Lesk, V. I., and Sternberg, M. J. E. (2008). Insights into protein flexibility: the relationship between normal modes and conformational change upon protein–protein docking. *Proc. Natl. Acad. Sci. U.S.A.*, **105**(30), 10390–10395.

Dominguez, C., Boelens, R., and Bonvin, A. M. J. J. (2003). Haddock: a protein-protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.*, **125**(7), 1731–1737.

Emekli, U., Schneidman-Duhovny, D., Wolfson, H., Nussinov, R., and Haliloglu, T. (2008). Hingeprot: Automated prediction of hinges in protein structures. *Proteins Struct. Funct. Bioinf.*, **70**(4), 1219–1227.

Fiorucci, S. and Zacharias, M. (2010). Binding site prediction and improved scoring during flexible protein–protein docking with attract. *Proteins Struct. Funct. Bioinf.*, **78**(15), 3131–3139.

Gil, V. A. and Guallar, V. (2014). pyproct: Automated cluster analysis for structural bioinformatics. *J. Chem. Theory Comput.*, **10**(8), 3236–3243.

Grünberg, R., Leckner, J., and Nilges, M. (2004). Complementarity of structure ensembles in protein-protein binding. *Structure*, **12**(12), 2125–2136.

Heifetz, A. and Eisenstein, M. (2003). Effect of local shape modifications of molecular surfaces on rigid-body protein–protein docking. *Protein Eng.*, **16**(3), 179–185.

Hildebrandt, A. K., Dietzen, M., Lengauer, T., Lenhof, H.-P., Althaus, E., and Hildebrandt, A. (2014). Efficient computation of root mean square deviations under rigid transformations. *J. Comput. Chem.*, **35**(10), 765–771.

Hinsen, K. (1998). Analysis of domain motions by approximate normalmode calculations. *Proteins Struct. Funct. Bioinf.*, **33**(3), 417–429.

Hinsen, K. (2000). The molecular modeling toolkit: a new approach to molecular simulations. *J. Comput. Chem.*, **21**(2), 79–85.

Hoffmann, A. and Grudinin, S. (2017). NOLB: Nonlinear rigid block normal-mode analysis method. *J. Chem. Theory Comput.*, **13**(5), 2123–2134.

Holm, L. and Sander, C. (1993). Protein structure comparison by alignment of distance matrices. *J. Mol. Biol.*, **233**(1), 123–138.

Horn, B. K. (1987). Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A.*, **4**(4), 629–642.

Huang, S.-Y. (2014). Search strategies and evaluation in protein–protein docking: principles, advances and challenges. *Drug Discovery Today*, **19**(8), 1081–1096.

Hung, L.-H. and Samudrala, R. (2014). fast_protein_cluster: parallel and optimized clustering of large-scale protein modeling data. *Bioinformatics*, **30**(12), 1774–1776.

Janin, J. (2005). Assessing predictions of protein–protein interaction: The CAPRI experiment. *Protein Sci.*, **14**, 278–283.

Kearsley, S. K. (1989). On the orthogonal transformation used for structural comparisons. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, **45**(2), 208–210.

Kneller, G. R. (1991). Superposition of molecular structures using quaternions. *Mol. Simul.*, **7**(1-2), 113–119.

Kovacs, J. A., Chacón, P., and Abagyan, R. (2004). Predictions of protein flexibility: First-order measures. *Proteins Struct. Funct. Bioinf.*, **56**(4), 661–668.

Kozakov, D., Clodfelter, K. H., Vajda, S., and Camacho, C. J. (2005). Optimal clustering for detecting near-native conformations in protein docking. *Biophys. J.*, **89**(2), 867–875.

Lindahl, E. and Delarue, M. (2005). Refinement of docked protein–ligand and protein–DNA structures using low frequency normal mode amplitude optimization. *Nucleic Acids Res.*, **33**(14), 4496–4506.

Lorenzen, S. and Zhang, Y. (2007). Identification of near-native structures by clustering protein docking conformations. *Proteins Struct. Funct. Bioinf.*, **68**(1), 187–194.

Magis, C., Di Tommaso, P., and Notredame, C. (2013). T-rmsd: a web server for automated fine-grained protein structural classification. *Nucleic Acids Res.*, **41**(W1), W358–W362.

Maschiach, E., Nussinov, R., and Haim, W. (2010). Fiberdock: Flexible induced-fit backbone refinement in molecular docking. *Proteins Struct. Funct. Bioinf.*, **78**(6), 1503–1519.

May, A. and Zacharias, M. (2008). Energy minimization in low-frequency normal modes to efficiently allow for global flexibility during systematic protein–protein docking. *Proteins Struct. Funct. Bioinf.*, **70**(3), 794–809.

Méndez, R., Leplae, R., De Maria, L., and J., W. S. (2003). Assessment of blind predictions of protein–protein interactions: Current status of docking methods. *Proteins: Struct., Funct., Genet.*, **52**, 51–67.

Moal, I. H. and Bates, P. A. (2010). Swarmdock and the use of normal modes in protein-protein docking. *Int. J. Mol. Sci.*, **11**, 3623–3648.

Mustard, D. and Ritchie, D. W. (2005). Docking essential dynamics eigenstructures. *Proteins Struct. Funct. Bioinf.*, **60**(2), 269–274.

Palma, P. N., Krippahl, L., Wampler, J. E., and Moura, J. J. (2000). Bigger: a new (soft) docking algorithm for predicting protein interactions. *Proteins Struct. Funct. Bioinf.*, **39**(4), 372–384.

Popov, P. and Grudinin, S. (2014). Rapid determination of rmsds corresponding to macromolecular rigid body motions. *J. Comput. Chem.*, **35**(12), 950–956.

Popov, P. and Grudinin, S. (2015). Knowledge of Native Protein-Protein Interfaces Is Sufficient to Construct Predictive Models for the Selection of Binding Candidates. *J. Chem. Inf. Model.*, **55**(10), 2242–2255.

Popov, P., Ritchie, D. W., and Grudinin, S. (2014). Docktrina: Docking triangular protein trimers. *Proteins Struct. Funct. Bioinf.*, **82**, 34–44.

Popov, P. and Grudinin, S. (2018). Eurecon: Equidistant uniform rigid-body ensemble constructor. *J. Mol. Graphics Modell.*, **80**, 313–319.

Ritchie, D. W. and Kemp, G. J. (2000). Protein docking using spherical polar fourier correlations. *Proteins Struct. Funct. Bioinf.*, **39**(2), 178–194.

Rupp, M., Ramakrishnan, R., and von Lilienfeld, O. A. (2015). Machine learning for quantum mechanical properties of atoms in molecules. *J. Phys. Chem. Lett.*, **6**(16), 3309–3313.

Salem, S., Zaki, M. J., and Bystroff, C. (2010). Flexsnap: Flexible non-sequential protein structure alignment. *Algorithms Mol. Biol.*, **5**(1), 12.

Schneidman-Duhovny, D., Nussinov, R., and Wolfson, H. J. (2007). Automatic prediction of protein interactions with large scale motion. *Proteins Struct. Funct. Bioinf.*, **69**(4), 764–773.

Shatsky, M., Nussinov, R., and H., W. (2004). A method for simultaneous alignment of multiple protein structures. *Proteins Struct. Funct. Bioinf.*, **56**, 143–156.

Shatsky M, Nussinov R, W. H. (2002). Flexible protein alignment and hinge detection. *Proteins: Struct., Funct., Genet.*, **48**(2), 242–256.

Smith, J. S., Isayev, O., and Roitberg, A. E. (2017). ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chem. Sci.*, **8**(4), 3192–3203.

Tama, F., Gadea, F., Marques, O., and Sanejouand, Y. H. (2000). Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins Struct. Funct. Bioinf.*, **41**(1), 1–7.

Theobald, D. L. (2005). Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Acta Crystallogr., Sect. A: Found. Crystallogr.*, **61**(4), 478–480.

Tirion, M. M. (1996). Large amplitude elastic motions in proteins from a single-parameter, atomic analysis. *Phys. Rev. Lett.*, **77**(9), 1905.

Venkatraman, V. and Ritchie, D. W. (2012). Flexible protein docking refinement using pose-dependent normal mode analysis. *Proteins Struct. Funct. Bioinf.*, **80**(9), 2262–2274.

Vreven, T., Moal, I. H., Vangone, A., Pierce, B. G., Kastritis, P. L., Torchala, M., Chaleil, R., Jiménez-García, B., Bates, P. A., Fernandez-Recio, J., *et al.* (2015). Updates to the integrated protein–protein interaction benchmarks: docking benchmark version 5 and affinity benchmark version 2. *J. Mol. Biol.*, **427**(19), 3031–3041.

Wilson, E. B. (1955). *Molecular vibrations: the theory of infrared and Raman vibrational spectra*. Courier Dover Publications.

Ye, Y. and Godzik, A. (2003). Flexible structure alignment by chaining aligned fragment pairs allowing twists. *Bioinformatics*, **19**, ii246–ii255.

Zacharias, M. (2010). Accounting for conformational changes during protein–protein docking. *Curr. Opin. Struct. Biol.*, **20**(2), 180–186.

Zavodszky, M. I., Lei, M., Thorpe, M., Day, A. R., and Kuhn, L. A. (2004). Modeling correlated main-chain motions in proteins for flexible molecular recognition. *Proteins Struct. Funct. Bioinf.*, **57**(2), 243–261.

Zhang, Y. and Skolnick, J. (2004). Spicker: a clustering approach to identify near-native protein folds. *J. Comput. Chem.*, **25**(6), 865–871.