Generating conformational transition paths with low potential-energy barriers for proteins

Minh Khoa Nguyen, Léonard Jaillet, Stephane Redon

Noname manuscript No.
(will be inserted by the editor)

Generating conformational transition paths with low potential-energy barriers for proteins

Minh Khoa Nguyen · Léonard Jaillet · Stéphane Redon

Received: date / Accepted: date

Abstract The knowledge of conformational transition paths in proteins can be useful for understanding protein mechanisms. Recently, we have introduced the As-Rigid-As-Possible (ARAP) interpolation method, for generating interpolation paths between two protein conformations. The method was shown to preserve well the rigidity of the initial conformation along the path. However, because the method is totally geometry-based, the generated paths may be inconsistent because the atom interactions are ignored. Therefore, in this article, we would like to introduce a new method to generate conformational transition paths with low potential-energy barriers for proteins. The method is composed of three processing stages. First, ARAP interpolation is used for generating an initial path. Then, the path conformations are enhanced by a clash remover. Finally, Nudged Elastic Band, a path-optimization method, is used to produce a low-energy path. Large energy reductions are found in the paths obtained from the method than in those obtained from the ARAP interpolation method alone. The results also show that ARAP interpolation is a good candidate for generating an initial path because it leads to lower potential-energy paths than two other common methods for path interpolation.

Keywords Protein conformational transition · as-rigid-as-possible · nudged elastic band · low-energy path

We would like to gratefully acknowledge funding from the European Research Council through the ERC Starting Grant No. 307629.

Minh Khoa Nguyen E-mail: jckhoa@yahoo.com ·
Léonard Jaillet E-mail: leonard.jaillet@inria.fr ·
Stéphane Redon E-mail: stephane.redon@inria.fr

Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP (Institute of Engineering, Univ. Grenoble Alpes), LJK, 38000 Grenoble, France

1 Introduction

Today, although a large number of protein structures are available, the transitions among them are little known due to the limitation of current experimental techniques [1, 2]. The knowledge of these transitions, however, may be key to the understanding of protein mechanisms [3].

Therefore, computational methods have become important for predicting protein conformational transition paths. Classical methods for finding molecular paths are molecular dynamics [4] and Monte Carlo [5] simulations. However, the time taken for these simulations risks to be too long before a system reaches one of the transition states which are rare, short-lived and associated with high-energy barriers [6]. Hence, recent developments focus on accelerating these simulation techniques by manipulating temperature, energy or forces [7–12].

For proteins, some methods ignore atomic interactions to predict transition pathways by only geometrical means because proteins are known to possess certain flexibility in their structures [3]. The interest of these methods is that they are typically much more time-efficient than the simulation-based methods because the computational cost related to atomic interactions is removed.

A simple geometric means to generate a path between two protein conformations is linear interpolation in the Cartesian coordinate system, where the atom positions are linearly interpolated between those in the initial and target conformations. However, the results from this method are prone to unrealistic bond lengths and bond angles. The linear interpolation in the internal coordinate system as implemented in LSQMAN [13, 14] gives more realistic results because bond lengths and bond angles are controlled. However, distortions may occur in the path conformations due to the accumulation of rigid transforms [15]. Therefore, more complicated methods have been devised to produce more realistic paths. The Linear Synchronous Transit (LST) method [16] generates a paths by using the linear interpolation of atom distances as constraint. The method is suitable for small-sized systems because it solves iteratively a quadratic-complexity problem, which is computationally expensive.

Complicated methods have been developed to find more realistic paths such as those based on hinge localization and restrained interpolation [17], or rigidity analysis combined with geometric and steric constraints (the FRODA method [18]. An extension of the latter, which is also called geometric targeting method, considers as additional constraint the Root Mean Squared Deviation (RMSD) from the target conformation [19].

Most recent methods employ a coarse grain model, and in particular the elastic network model (ENM), which represents a protein by a string of alpha carbon atoms. The advantage of such representation is typically the reduction of the number of degrees of freedom as compared to an all-atom representation, and it has shown some success in predicting conformational transition paths for proteins. Among the methods using such a model, the Climber method [20] which imposes harmonic restrains on distances among alpha carbon atoms successfully predicted intermediate structures for several proteins. The MORPHPRO method [21] performs linear interpolation of alpha-carbon atom positions

while restraining the distances between consecutive alpha carbon atoms to about 3.8 Å, to produce a path of protein-like structures. The combination of ENM with a Brownian dynamics simulation is proposed in [22] for protein path search. In [23], the idea of minimum action path [24] combined with structural relaxation has successfully predicted intermediate structures for several protein cases. These methods, however, depend on an iterative procedure to find a path, which can be computationally expensive. The ENM also allows to extract normal modes where low-frequency modes are known to associate with large-amplitude motions of proteins. Protein paths derived from normal modes have been proposed in many studies where the modes are extracted using Cartesian coordinates [1, 25], internal coordinates [26, 27] or rigid blocks representations [28, 29]. Some of these methods are integrated in web servers [28, 27] making them easily accessible to the public.Other methods using the coarse grain model for generating protein paths include [30–33]. Coarse-graining methods improve the computational cost significantly compared to many methods based on an all-atom model. The reader can refer to Orelana et al. [22] for the efficiency of some of these methods. Nevertheless, due to model approximation, these methods can give inaccurate results or miss important information [34, 35].

Recently, we proposed to apply the As-Rigid-As-Possible (ARAP) principle from computer graphics for generating a path between two protein conformations [36]. The method was shown to have linear complexity and does not require an iterative solver, and hence, is efficient even when an all-atom model is used. The obtained paths preserve well the rigidity of the original protein structures (in terms of bond lengths and bond angles) whose changes are sources for high-energy conformations.

Although the geometry-based methods can give good predictions of global motions of proteins, they may generate inconsistent paths because the atomic interactions are ignored. To generate physically feasible paths, path-optimization methods such as Nudged Elastic Band (NEB) [37], String [38], or Conjugate Peak Gradient (CPG) [39] can be applied with the atomic interactions. Their adaptations have been successfully applied for molecular systems as shown by [40–42]. In general, they use analytical approaches to minimize the structures on a given path (the NEB and String methods) or to create a new optimal structure between two given structures on the fly (the CPG method).

The purpose of this article is to introduce a three-stage method for generating a protein conformational transition path with low potential energy between two given conformations. The first stage uses ARAP interpolation for generating an initial path. The conformations in this path are then improved with a clash remover in the second stage. Finally, the third stage applies NEB to optimize this path to a low-energy path. The NEB method has been successfully applied for non-protein systems [42]. However, to the best of our knowledge, such application has never been applied for proteins.

In our study, we employ potential energy instead of free energy because the computation of potential energy is readily available in many software packages such as GROMACS [43], CHARMM [44], AMBER [45], GROMOS [46], etc. Although potential energy lacks dynamics information (kinetic energy and

entropy), which may limit the discussion of the results, it has been shown to reveal important trends and insights for many systems [47]. In any case, our goal is to produce a first physically feasible path so that this path can be used for other advanced methods such as the transition path sampling or umbrella sampling for computing free energy differences [48,49] or predicting reaction rates [50]. In the next section, the method will be presented in detail, which is followed by the section on the experiments and results.

2 Methodology

The overview of the proposed method is shown in Figure 1. The input includes an initial and a target conformation of a protein. The first stage uses ARAP interpolation (ARAPi) to generate a path representing the transition from the initial conformation to the target conformation. The second stage improves this path by removing clashes in the path conformations. The third stage applies NEB on the path to produce an optimized path with a low energy barrier. The details of each stage are presented below.

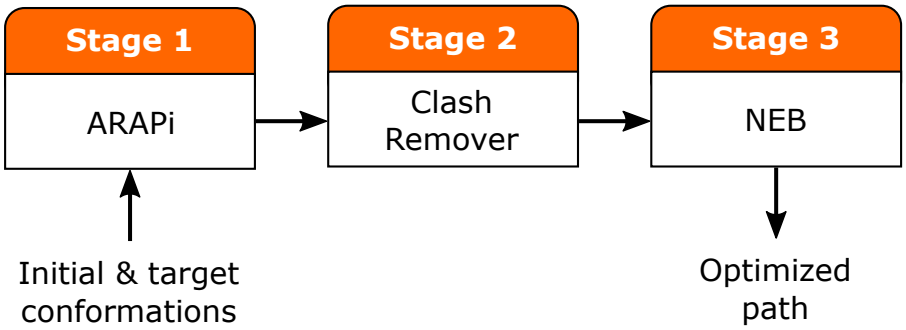

Fig. 1: The three-stage method for generating an optimized path from an initial and a target conformation.

2.1 ARAP interpolation (ARAPi)

The ARAPi method has been proposed for generating an interpolation path between two protein conformations [36]. Although the method is purely geometrical, the generated paths capture feasible motions by conserving the local rigidity of the initial conformation (i.e. the bond lengths and bond angles) as much as possible. The main principle of the method is reminded here.

Let us take a molecular structure containing n atoms $a_0, \ldots, a_{n-1}$ connected by covalent bonds. Let $\mathbf{p}_i \in \mathbf{R}^3$ be the atom positions in the initial conformation $\mathcal{S}$ and $\mathbf{p}'_i$ their positions in the target conformation $\mathcal{S}'$ for $i \in [0, n-1]$.

From the molecular topology, n sets $\mathcal{N}_i$ can be extracted, where each set consists of a central atom a_i and all the atoms covalently bonded to this central atom. This way of constructing the ARAP sets is also called one-ring neighbor topology.

From each ARAP set, a cell $\mathcal{C}_i$ is defined as a set of the positions of all the atoms in $\mathcal{N}_i$ for the conformation $\mathcal{S}$. An example of decomposing a simple molecular system into ARAP cells using the one-ring neighborhood is shown in Figure 2a and 2b.

Let us now consider $\mathcal{C}_i$ and $\mathcal{C}'_i$, the cells centered on a_i in the initial and target conformations, respectively. According to Sorkine et al. [51], the rotation to best align $\mathcal{C}_i$ onto $\mathcal{C}'_i$ can be computed by minimizing the *cell deformation energy* $E(\mathcal{C}_i, \mathcal{C}'_i)$, also called *ARAP cell energy*:

$$E(\mathcal{C}_i, \mathcal{C}'_i) = \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\mathbf{p}'_i - \mathbf{p}'_j - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)||^2 \tag{1}$$

where ω_{ij} is the weight associated to the bond b_{ij} connecting atom a_i with atom a_j in $\mathcal{N}(i)$. We use $\omega_{ij} = \omega_{ji} = 1$. Figure 2c shows an example of a rotation $\mathbf{R}_i$ which minimizes the ARAP cell energy.

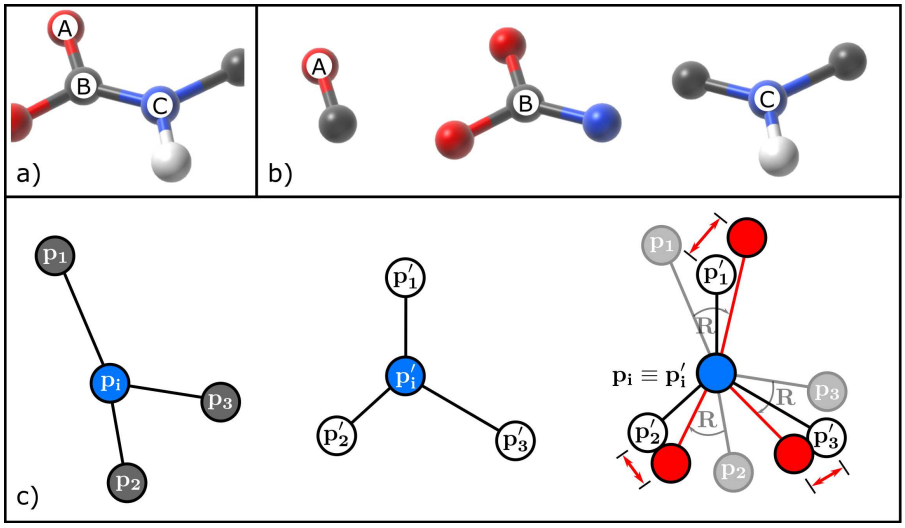

Fig. 2: A simple molecular system in (a) is decomposed into ARAP cells for 3 atoms A, B, and C using the one-ring neighborhood (in (b)). An example of ARAP-cell alignment by rotation is shown in (c): the left-most and middle pictures represent the initial and target cells with corresponding central vertices $\mathbf{p_i}$ and $\mathbf{p'_i}$, respectively; the right-most picture shows the best rotation $\mathbf{R}$ to align the initial cell with the target cell by minimizing the sum of the squared distances represented by the red arrows.

Given the aligning rotations $\mathbf{R}_i$ of all the cells, an ARAP transformation finds the vertex positions $\widehat{\mathbf{p}}_i$ which minimizes the ARAP energy, E_{ARAP}, defined as:

$$E_{ARAP} = \sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_i - \widehat{\mathbf{p}}_j - \mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j)||^2 \tag{2}$$

where ω_i is a cell weight. By default, we take $\omega_i = 1, \forall i$, i.e. uniform cell weight.

The idea of ARAP interpolation is to interpolate the cell aligning rotations found by minimizing Equation 1, and then, to compute the interpolated conformations based on these interpolated rotations. Therefore, a parameter $t \in [0, 1]$ that represents interpolation instances is introduced, where $t = 0$ and $t = 1$ correspond to the initial conformation and the target conformation, respectively. For finding the atom positions $\widehat{\mathbf{p}}_i(t)$ of an intermediate conformation, one minimizes the ARAP energy at t, $E_{ARAP}(t)$, defined as:

$$E_{ARAP}(t) = \textstyle\sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_i(t) - \widehat{\mathbf{p}}_j(t) - \mathbf{R}_i(t)(\mathbf{p}_i - \mathbf{p}_j)||^2$$

where $\mathbf{R}_i(t)$ is the interpolated rotation between the identity transform $\mathbf{I}$ and the rotation $\mathbf{R}_i$. Nguyen et al. [36] proposed to use the Spherical linear interpolation (Slerp) method [52] for rotation interpolation.

The minimization problem has been shown to lead to solving a linear algebra system $\mathbf{L}\widehat{\mathbf{p}}(t) = \mathbf{b}(t)$ [36], where $\widehat{\mathbf{p}}(t)$ is a matrix concatenating the unknown atom positions of the intermediate conformation at t and $\mathbf{b}(t)$ is a matrix of the same size. $\mathbf{L}$ is a square matrix independent of t. When the one-ring neighbor topology is used for constructing the ARAP sets, $\mathbf{L}$ is a sparse, symmetric and positive definite matrix, for which we can thus compute a Cholesky decomposition. Because its coefficients are independent on t, this decomposition can be performed only once, and used to compute all the intermediate conformations. The characteristics of $\mathbf{L}$ also makes the ARAPi method efficient because the complexity of solving the linear algebra equation is almost linear. The details on the construction of these matrices can be found in [36].

The last equation, however, does not allow the interpolation path to reach the target conformation, i.e. the conformation solved for $t = 1$ does not coincide exactly with the target conformation. Hence, [36] proposed to modify $E_{ARAP}(t)$ as follows,

$$\begin{aligned} E_{ARAP}(t) = \sum_i \omega_i \sum_{j \in \mathcal{N}_i} \omega_{ij} ||\widehat{\mathbf{p}}_i(t) - \widehat{\mathbf{p}}_j(t) \\ - s_{ij}(t)\mathbf{R}_{ij}(t)\mathbf{R}_i(t)(\mathbf{p}_i - \mathbf{p}_j)||^2 \end{aligned}$$

where $\mathbf{R}_{ij}(t)$ and $s_{ij}(t)$ are the interpolated values of the extra rotation $\mathbf{R}_{ij}$ and the scale factor s_{ij}. $\mathbf{R}_{ij}$ and s_{ij} must satisfy the following relation for $\forall i$ and $\forall j \in \mathcal{N}_i$,

$$\mathbf{p}'_i - \mathbf{p}'_j = s_{ij}\mathbf{R}_{ij}\mathbf{R}_i(\mathbf{p}_i - \mathbf{p}_j) \tag{3}$$

i.e., $\mathbf{R}_{ij}$ and s_{ij} define the extra operations that allow to transform $\mathbf{p}_i - \mathbf{p}_j$ exactly into $\mathbf{p}'_i - \mathbf{p}'_j$.

The last equation also suggests one to compute $\mathbf{R}_{ij}$ after the computation of $\mathbf{R}_i$. The computation of $\mathbf{R}_{ij}$ can be done by aligning the vector $\mathbf{R}_i(\mathbf{p}_i - \mathbf{p_j})$ onto the vector $\mathbf{p}'_i - \mathbf{p}'_j$. The scale factor can be computed as, $s_{ij} = \frac{\|\mathbf{p}'_i - \mathbf{p}'_j\|}{\|\mathbf{p}_i - \mathbf{p}_j\|}$. As stated in [36], let us notice that the method is not symmetric, i.e. an interpolation from structures A to B would give a different result than an interpolation from B to A. However, these differences are not related to hysteresis phenomena since the method only considers the geometric aspects. The Experiments and Results section shows an example that the differences between the forward and reverse paths are very limited.

2.2 Clash remover

We introduce here a method for removing clashes due to overlapping atoms (steric clashes), and clashes due to the crossing of covalent bonds through aromatic-ring surfaces (called ring clashes by analogy). As we will see, removing steric clashes strongly decreases the energy of the path conformations, which leads to lower-energy conformations after the NEB optimization. Removing ring clashes is also necessary, since the escape of the clashing bonds from these rings requires overcoming a small energy barrier (the van der Waals energy increases when the bonds cross over the ring boundaries), and hence, local minimization methods cannot resolve this problem.

A steric clash is detected whenever the distance between any pair of non-bonded atoms is smaller than a predefined threshold d_{steric}. To remove this clash type, we optimize a system of springs between these atoms to push them apart. Besides, extra springs are also established for the bonds involving these clashing atoms to preserve these bond lengths. An example of this model is shown in Figure 3a and 3b.

The spring force $\mathbf{F_C}$ applied on the atom C which is bonded to the clashing atom A is defined as

$$\mathbf{F_C} = k_{bonded}(\|\mathbf{p_A} - \mathbf{p_C}\| - d^0_{CA})^2 \frac{\mathbf{p_A} - \mathbf{p_C}}{\|\mathbf{p_A} - \mathbf{p_C}\|}$$

where k_{bonded} is the spring constant for bonded atoms and d^0_{CA} is the initial bond length between the atoms A and C. $\mathbf{p_A}$ and $\mathbf{p_C}$ are the current positions of the atoms A and C.

The force applied on the clashing atom A is $\mathbf{F_A} = -\mathbf{F_C} + \mathbf{F_{AB}}$, i.e. the sum of the opposite force to $\mathbf{F_C}$ and the spring force from the interaction with its clashing atom B, $\mathbf{F_{AB}}$, defined as,

$$\mathbf{F_{AB}} = k_{steric}(\|\mathbf{p_B} - \mathbf{p_A}\| - d_{steric})^2 \frac{\mathbf{p_B} - \mathbf{p_A}}{\|\mathbf{p_B} - \mathbf{p_A}\|} \tag{4}$$

where k_{steric} is the spring constant for steric clashing atoms and $\mathbf{p_B}$ is the current position of the atom B. The forces applied on the atoms B and D can be derived similarly as for the atoms A and C, respectively.

For ring clashes, we first detect all the aromatic rings in the structures. This is simple because the standard amino acids in proteins which have aromatic rings are known such as histidine, proline, phenylalanine, tyrosine, and tryptophan. An illustration of the ring-clash detection and removal is shown in Figure 3c, 3d and 3e. A ring clash is detected if any covalent bond crosses through a ring surface. For each ring, we first locate its center of mass made by all of the ring atoms. The ring surface is then decomposed into triangular surfaces, each of which is defined by the ring center-of-mass and a ring bond (Figure 3d). Finally, we geometrically check whether a covalent bond nearby cuts any of the triangular surfaces. To remove a ring clash, an external force is applied to push each atom of the clashing bond outside the ring. The pushing direction is a vector $\mathbf{r_{ring}}$ pointing from the ring center-of-mass to the cut-point position (the intersection of the clashing bond and any triangular surfaces of the ring). Springs are also used for bonded atoms in the ring to maintain the ring shape (Figure 3e).

Suppose A and B are the atoms of the clashing bond of a ring clash. The spring force applied on the atom A is defined as,

$$\mathbf{F_A} = k_{ring}(\|\mathbf{p_A} - \mathbf{r_c}\| - d_{ring})^2 \frac{\mathbf{r_{ring}}}{\|\mathbf{r_{ring}}\|} \tag{5}$$

where k_{ring} is the force constant for clashing-bond atoms in a ring clash. $\mathbf{p_A}$ is the current position of the atom A, $\mathbf{r_c}$ is the current ring center, and d_{ring} is the minimum distance for resolving ring clashes. The force $\mathbf{F_B}$ applied on atom B can be expressed in a similar manner.

The force applied on a ring atom is the sum of the spring forces from the interactions with its two neighbor (bonded) atoms. For example the force $\mathbf{F_{R_2}}$ applying on the ring atom R_2 is

$$\mathbf{F_{R_2}} = \mathbf{F_{R_2R_1}} + \mathbf{F_{R_2R_3}} \tag{6}$$

where $\mathbf{F_{R_2R_1}}$ and $\mathbf{F_{R_2R_3}}$ are the spring forces among the bonded atom, and hence, defined as,

$$\mathbf{F_{R_2R_1}} = k_{bonded}(\|\mathbf{p_{R_1}} - \mathbf{p_{R_2}}\| - d^0_{R_1R_2})^2 \frac{\mathbf{p_{R_1}} - \mathbf{p_{R_2}}}{\|\mathbf{p_{R_1}} - \mathbf{p_{R_2}}\|}$$
$$\mathbf{F_{R_2R_3}} = k_{bonded}(\|\mathbf{p_{R_3}} - \mathbf{p_{R_2}}\| - d^0_{R_3R_2})^2 \frac{\mathbf{p_{R_3}} - \mathbf{p_{R_2}}}{\|\mathbf{p_{R_3}} - \mathbf{p_{R_2}}\|}$$

where $d^0_{R_1R_2}$ and $d^0_{R_3R_2}$ are the initial lengths of the bonds between R_1 and R_2, and between R_3 and R_2, respectively. We use the same spring constant k_{bonded} for bonded atoms in both steric-clash and ring-clash removals.

The clash removal is an adaptive process applied for the maximum of 200 steps with an initial step size of 0.005 fs. For each step, each atom displacement is also constrained not to exceed 0.1 Å. In addition, if the new-state energy is greater than the last-state energy, the new state is rejected and the step size is reduced by half. Contrarily, if the new-state energy is smaller than the last-state energy, the new state is accepted and the step size is increased by 1.2.

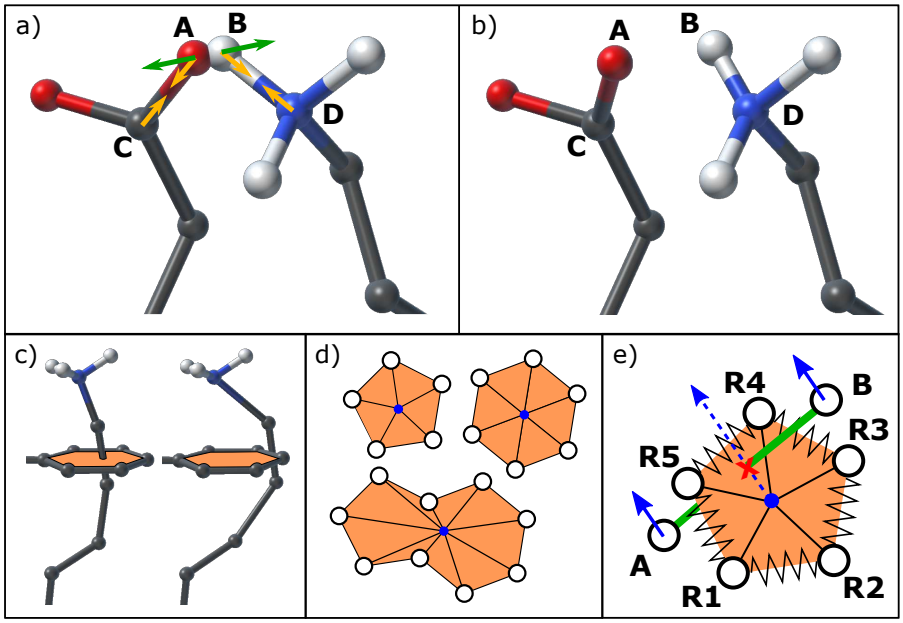

Fig. 3: Example of a steric-clash removal: a) two clashing atoms are an oxygen (red ball A) and a hydrogen (white ball B); the spring forces for the steric-clash removal are applied on the clashing atoms (green arrows) and the atoms bonded to them (orange arrows); b) the result after applying steric-clash removal. Example of a ring-clash removal: c) A ring clash occurs when a covalent bond crosses through an aromatic-ring surface (left). The result of the proposed ring-clash removal method (right). d) The three types of rings existing for the 20 standard amino acids, with their triangular-surface decomposition. White balls represent atoms and small blue circles represent the ring center-of-mass. e) The cutting point (red cross) between the clashing bond (green line) and one of the triangular surfaces is determined. Then, the force direction (blue dashed arrow) for clash removal is determined. Finally, the clash removing forces (blue solid arrows) are applied on the two atoms of the clashing bonds. The springs among the bonded atoms of the ring, which are presented by zigzag lines, are there to maintain the ring shape during the escape of the clashing bond.

The process is stopped as soon as all the steric and ring clashes are resolved. This adaptive process ensures that the clash-removing minimization always goes downhill in its energy landscape. Table 1 shows the chosen parameters for the clash removal stage. With these parameters, the method was efficient enough to remove all the clashes within 200 steps for all the tested systems.

Table 1: Parameters for the clash remover.

k_{steric}	d_{steric}	k_{ring}	d_{ring}	k_{bonded}
5 N.nm^{-2}	0.11 nm	10 N.nm^{-2}	0.3 nm	0.05 N.nm^{-2}

2.3 Nudged Elastic Band

Nudged Elastic Band is a well-known method for finding low-energy paths from a given initial path [37,53]. The method adjusts the initial path until it converges to a final minimum-energy path. To do this, the method iteratively applies forces on each conformation of the path. Each force is composed of two components. The first one, the component of the potential force perpendicular to the path, drives a conformation to its minimum-energy state on the hyperplane perpendicular to the path tangent (a property of minimum-energy paths). The second one, the spring force, maintains the distances among the consecutive conformations on the path. We use the NEB version presented in [37], which improves the kink problem in the paths due to the path-tangent approximation. For finding the best direction to move each conformation in each iteration, we apply the efficient Fast Inertial Relaxation Engine (FIRE) method [54]. We apply FIRE for moving the conformations and stop the process after a maximum number of iterations n_{NEB} is reached. We use the spring constant of 1000 eV.Å^{-2} for all the springs, and $n_{NEB} = 1000$. We found that these parameters gave stable and converging results for the tested systems. The NEB method was implemented as a multi-core module on the SAMSON platform [55] by one of our colleagues.

3 Experiments and Results

3.1 Experimental setup

We applied the method on the 12 protein cases that had been used to validate the ARAPi method [36]. We would like to show that the proposed method can give low-energy transition paths. Hence, for comparison sake, we also present the results when using two other interpolation methods: linear interpolation in Cartesian space (Linear) and Linear Synchronous Transit (LST). For generating low-energy paths with either Linear or LST, we perform the same procedure shown in Figure 1 with the replacement of the ARAPi method by one of them.

The Linear method generates intermediate conformations by linearly interpolating atom positions between the initial and target conformations. Therefore, the position of the ith atom in the tth intermediate conformation is computed as,

$$\widehat{\mathbf{p}}_i(t) = (1-t)\mathbf{p}_i + t\mathbf{p}'_i \tag{7}$$

The LST method, originally proposed by Halgren et al. [16], has been employed alone or in combination with more sophisticated methods [56–58]

for generating good initial guesses of reaction paths when studying chemical reactions. The combination of the method with NEB has been successfully applied to small systems [42] but is rarely found for protein systems due to the quadratic complexity of the method. We propose here to adapt the method for protein systems.

The idea behind LST is to generate intermediate conformations such that each conformation has the atom-pair distances as close to the linearly interpolated ones between the initial and the target conformations as possible. Because this condition cannot be satisfied for all the atom pairs, the atom positions $\widehat{\mathbf{p}}_i(t)$ in the tth intermediate conformation are solved by minimizing the following energy formula:

$$E(t) = \sum_{i>j} \frac{\left(r_{ij}(t) - \overline{r}_{ij}(t)\right)^2}{\left(\overline{r}_{ij}\right)^4} + \beta \sum_i \|\widehat{\mathbf{p}}_i(t) - \overline{\mathbf{p}}_i(t)\| \tag{8}$$

where $\overline{\mathbf{p}}_\mathbf{i}(t) = (1-t)\mathbf{p}_i + t\mathbf{p}'_i$ is the interpolated atom position. $r_{ij}(t) = \|\widehat{\mathbf{p}}_i(t) - \widehat{\mathbf{p}}_j(t)\|$ is the distance between the ith and jth atoms in the intermediate conformation and $\overline{r}_{ij}(t)$ is the prescribed distance between the ith and jth atoms and calculated by linear interpolation, i.e. $\overline{r}_{ij} = (1-t)\|\mathbf{p}_i - \mathbf{p}_j\| + t\|\mathbf{p}'_i - \mathbf{p}'_j\|$.

The role of the first term in Equation 8 is to restrain the atom-pair distances toward their linearly interpolated ones. The role of the second term is to suppress the translational and rotational variations. Hence, β is typically taken small enough so that this second term is much smaller than the first one.

In our implementation of the LST method, to locate each intermediate conformation, we start from a conformation whose atom positions are linearly interpolated between the initial and final conformations, i.e. $\widehat{\mathbf{p}}_i(t) = \overline{\mathbf{p}}_\mathbf{i}(t)$. Then, the FIRE method is applied to minimize this conformation for a number of iterations n_{LST}. Finally, since considering all the atom pairs would incur a quadratic cost, we consider only the atom pairs which are covalently bonded. In the LST method, we use $\beta = 10^{-6}$ as in [16] and $n_{LST} = 1000$.

The ARAPi, Linear, and LST methods were implemented in C++ codes, as modules on the SAMSON platform.

A summary of the experiments is shown in Table 2. In each case, the initial and target structures are obtained from the Protein Data Bank [59]. Their PDB entries and chain IDs are given in the third column of the table. Because our method requires potential-energy evaluation, the input structures must be complete and have exactly the same set of atoms. However, this is hardly the case because the protein structures obtained from online databases usually have missing residues, missing atoms, and mismatched atoms due to mutations. For reconstructing missing residues, we use MODELLER [60] integrated in Chimera [61]. For resconstructing heavy atoms, SwissPDB [62] is used. Afterwards, the command *pdb2gmx* in GROMACS [63] is used with the GROMOS96 43a1 force field parameters to add hydrogen atoms. The topology output is then used for the energy evaluation by GROMACS integrated in the SAMSON platform.

At this stage, the input structures have no missing residues or atoms. Next, we need to resolve the mutation problem by generating a new target structure

Table 2: Experiment descriptions and some results for generating low-energy paths with the proposed method.

Experiment ID	Name	Initial/Target (pdb and chain code)	no. atoms	Distance d_m (Å)	Path size $\mathcal{L}$	NEB time (s)
1	5'-Nucleotidase	1HP1(A)/1HPU(C)	5123	32.19	97	534.5
2	Adenylate Kinase	4AKE(A)/1AKE(A)	2085	21.47	64	99.9
3	Alcohol Dehydrogenase	8ADH(A)/6ADH(A)	3516	12.31	37	119.1
4	Calmodulin	1CFD(A)/1CFC(A)	1459	13.41	40	39.2
5	Collagenase	1NQD(A)/1NQJ(B)	1257	38.70	116	112.1
6	Dengue 2 Virus Envelope Glycoprotein	1OAN(A)/1OK8(A)	3866	32.66	98	336.2
7	Dihydrofolate Reductase	1RX2(A)/1RX6(A)	1602	12.10	36	42.2
8	Diphtheria Toxin	1DDT(A)/1MDT(A)	5223	49.89	150	874.4
9	DNA Polymerase	1IH7(A)/1IG9(A)	9525	30.61	92	1153.2
10	Pyrophosphokinase	1HKA(A)/1Q0N(A)	1597	24.79	74	83.7
11	Pyruvate Phosphate Dikinase	1KBL(A)/2R82(A)	8541	47.85	144	1619.7
12	Spindle Assembly Checkpoint Protein	1DUJ(A)/1KLQ(A)	1934	37.54	113	174.3

such that it has the same set of atoms as the initial structure. We found that we could use the ARAP interpolation method in [36] for this job. Firstly, the method in [36] can take the initial and target structures with mismatched atoms due to mutation, and gives a path where all the conformations are defined for the same structure. Secondly, the final conformation in the path coincides exactly with the target structure (for matched atoms). Therefore, we applied this method to obtain a path with no intermediate conformations, i.e. a path containing only two conformations. The first and last conformations in this path become the initial and target conformation, respectively. Another advantage gained from this technique for resolving the mutation issue is that the resulting initial and target conformations are also structurally aligned.

Finally, since most path-optimization methods find paths between two conformations which are at their local minima, the initial and target conformations are relaxed to their local minima. The FIRE method is also used for this job. The relaxed conformations serve as the initial and target conformations for the three-stage method shown in Figure 1.

The number of conformations to generate for a path is defined based on d_m, the maximum displacement of all the atoms between the initial and target conformations, i.e.

$$d_m = \max_{i \in [0, n-1]} (|\mathbf{p}'_i - \mathbf{p}_i|) \tag{9}$$

We chose to have 3 conformations/Å, and hence $\mathcal{L}$, the total number of conformations along a path (including the initial and target conformations) is equal to:

$$\mathcal{L} = [\frac{d_m}{d_0}] \tag{10}$$

where $[\cdot]$ is the rounding operator which rounds a value to their nearest integer value and $d_0 = \frac{1}{3}$Å. Table 2 also shows d_m and $\mathcal{L}$ for all the experiments.

3.2 Results

3.2.1 Processing Time

The total processing time of each experiment is shown in Figure 4a. It sums up the time from the path interpolation (with ARAPi, Linear, or LST), from clash removal, and from NEB optimization. Results show that, in all the experiments, the method using ARAPi is slightly less time-consuming than the one using Linear, whereas the one using LST is the most time-consuming.

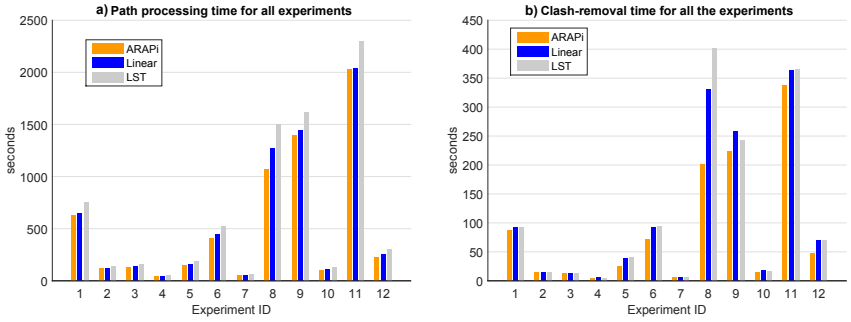

Fig. 4: a) Path-processing time and b) clash removal time of each experiment for the paths generated from the ARAPi, Linear, and LST methods.

To explain why ARAPi paths tend to consume least time, let us analyze the time in detail for each stage of the method proposed in Figure 1.

The computational complexity for the interpolation stage is almost linear with the number of conformations and the number of atoms for ARAPi and LST, whereas it is is strictly linear for Linear. The interpolation time per conformation per atom is smallest for Linear ($29.5 \pm 11.4\ \mu s$), followed closely by ARAPi ($32.3 \pm 11.5\ \mu s$), and largest for LST ($239.6 \pm 13.5\ \mu s$). In other words, LST is 7 to 8 times slower than ARAPi and Linear because it requires an iterative solver to find each intermediate conformation. Note that here, we chose the number of iterations in LST quite high, $n_{LST} = 1000$, to ensure the convergence of the method. However, we will see later that despite this setting, LST tends to lead to higher-energy paths as compared to the solutions obtained from Linear or ARAPi.

Figure 4b shows the computational time spent for removing clashes. As one can see, the clash removal takes the least time for the ARAPi paths in all the experiments. This is because steric clash is domninant in all the paths and the ARAPi paths have the least number of steric clashes (see below).

Finally, the time taken by the NEB stage is shown in the last column of Table 2. This time is independent of the interpolation methods since the number of NEB iterations and the path size are the same for each experiment. Among the three stages, the NEB stage is the most time-consuming.

3.2.2 Number of clashes

Figure 5 shows on the logarithmic scale the total initial number of clashes (augmented by 1 to avoid undefined values) detected for the interpolated paths.

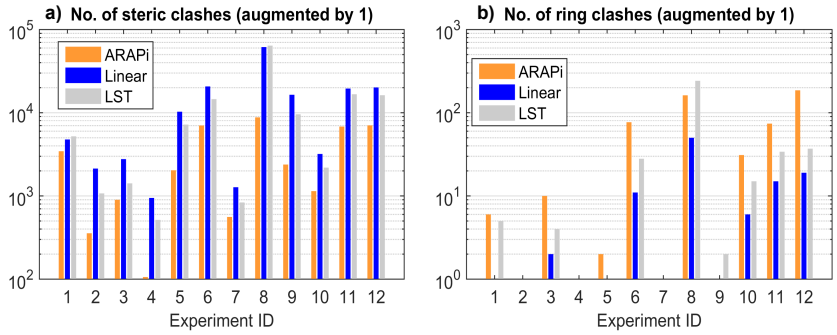

Fig. 5: a) Number of steric clashes and b) number of ring clashes, for all the experiments. The values are augmented by 1 to avoid undefined values on the logarithmic scale.

As one can see, ARAPi leads to paths with fewer steric clashes than the other interpolation methods (Figure 5a). This result was expected as ARAPi better preserves the local rigidity of the initial conformation. Surprisingly, ARAPi tends to lead to more ring clashes than the other methods (Figure 5b). The probable reason is that Linear and LST tend to give the paths where certain parts of the structure are shrunk (see Figure 7). This phenomenon reduces the ring-surface areas, and hence, ring clashes are less likely to occur. Note that the number of ring clashes remains relatively small compared to the number of steric clashes.

3.2.3 Reduction of energy barriers

Our objective is to obtain paths with energy barriers as low as possible. Therefore, to assess a path quality, we compute the potential-energy barrier $\bar{E}$ of each path. This quantity is computed as the difference between the maximum energy of the path conformations and the energy of the initial conformation, i.e.

$$\bar{E} = \max_{i \in [0, \mathcal{L}-1]} E(\mathbf{R_i}) - E(\mathbf{R_0}) \tag{11}$$

In our proposed method, the energy barrier is reduced at two stages: after the clash removal and after the path optimization. Hence, we studied the reduction factors $f_{interp/clash} = \bar{E}_{interp}/\bar{E}_{clash}$ and $f_{clash/neb} = \bar{E}_{clash}/\bar{E}_{neb}$, where $\bar{E}_{interp}$, $\bar{E}_{clash}$, $\bar{E}_{neb}$ are the path barriers after interpolation, after clash removal, and after NEB optimization, respectively. We observed large energy reductions for both stages, with $f_{interp/clash}$ ranging from $2{\times}10^2$ to $2{\times}10^{14}$ and

$f_{clash/neb}$ ranging from 5×10^2 to 6.3×10^4. The details of the energy reduction for each benchmark can be found in Section 1 of the Supplementary Material.

The potential-energy barriers of the final paths are shown in Table 3 and Figure 6. As one can see, the optimized paths arising from ARAPi have the lowest energy barriers in all the experiment (Figure 6). The optimized paths arising from LST have lower energy barriers than those arising from Linear in only two experiments (experiments 5 and 8). It means that despite the 1000 iterations required by the LST method, the final solutions obtained have typically higher energy barriers than those arising from the other two methods (ARAPi and Linear). Table 3 also shows that ARAPi can produce paths with energy barriers limited to few hundreds kcal/mol as in experiments 2, 4, and 7 where the barriers are less than 500 kcal/mol.

Table 3: Potential-energy barriers (in kcal/mol) of the optimized paths arising from ARAPi, Linear, and LST methods.

Experiment ID	ARAPi	Linear	LST
1	5135	9379	12633
2	383	634	1410
3	1917	3921	8349
4	366	1479	2119
5	2071	9063	6622
6	10486	19556	25160
7	457	1066	1341
8	8339	110543	91242
9	1289	7984	10997
10	5801	5909	6988
11	4449	16572	27042
12	22864	32149	37755

3.2.4 Preservation of bond lengths, bond angles, and dihedral angles

It has been shown previously [36] that ARAPi preserves well bond lengths and bond angles, whereas dihedral angles may largely vary to allow the conformational changes. Here, we performed the same analysis, comparing the paths obtained after optimization using either ARAPi, LST or Linear. The plots showing the deviation in bond lengths, bond angles, and dihedral angles along a path from the initial structure of one representative experiment can be found in Section 2 of the Supplementary Material. First, let us note that before optimization, the LST paths preserve well the bond lengths since the objective of LST is to directly make the bond lengths as close as possible to the linear interpolated bond-length values. However, LST poorly preserves bond angles, particularly because our implementation of LST only considers covalently bonded atom pairs.

Now when considering the paths after optimization, we observed that those generated from ARAPi largely preserve much better the bond lengths and bond angles as compared to those generated from Linear or LST.

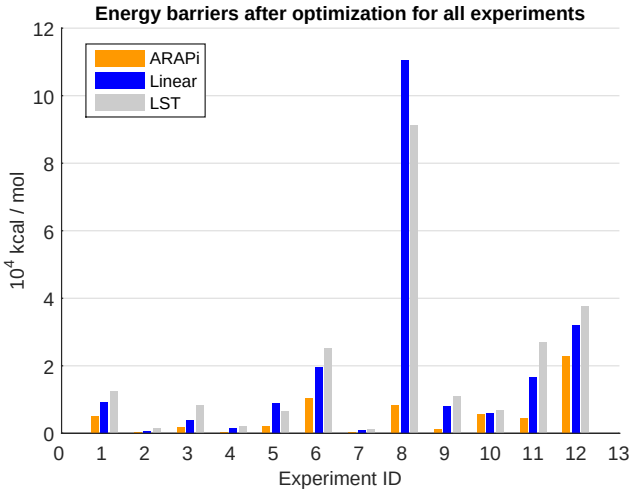

Fig. 6: Final potential-energy barriers of the optimized paths for each experiment.

Regarding the dihedral angle, the three interpolation methods allow large variations of this quantity even after optimization, which shows that the proteins mostly transition by dihedral motions. Such a result was expected as dihedral motions are the ones which best preserve the potential energy.

3.2.5 Visual inspection of the optimized paths

In general, the motions obtained with the ARAPi method are subjected to less structural degeneration than those obtained with the Linear or LST methods. Let us consider, as a representative example, the case of the Diphtheria Toxin shown in Figure 7. This figure shows the motions of Diphtheria Toxin after interpolation (with ARAPi, Linear, or LST), and after optimization (with NEB). Noticeable changes were not observed after clash removal because the clash remover only resolves steric and ring clashes locally, which does not affect significantly the secondary structures in our case. Therefore, the results after clash removal are not shown in the figure.

Figure 7 shows significant changes in the protein structures of the paths after the path optimization, especially for those arising from the Linear and LST methods. In contrast, the ARAPi path and its optimized one do not differ greatly, i.e. the ARAPi path is already very close to its optimized solution. This behavior has also been observed for the rest of the experiments (not shown). A closer examination at Figure 7 reveals that the red-and-yellow domain is shrunk at some moment along the optimized Linear and the optimized LST paths while the optimized ARAPi path does not show this behavior. We found that the most shrunk conformations were the ones with the highest energy

along the paths. This explains why the optimized Linear and optimized LST paths have higher potential-energy barriers than the optimized ARAPi path.

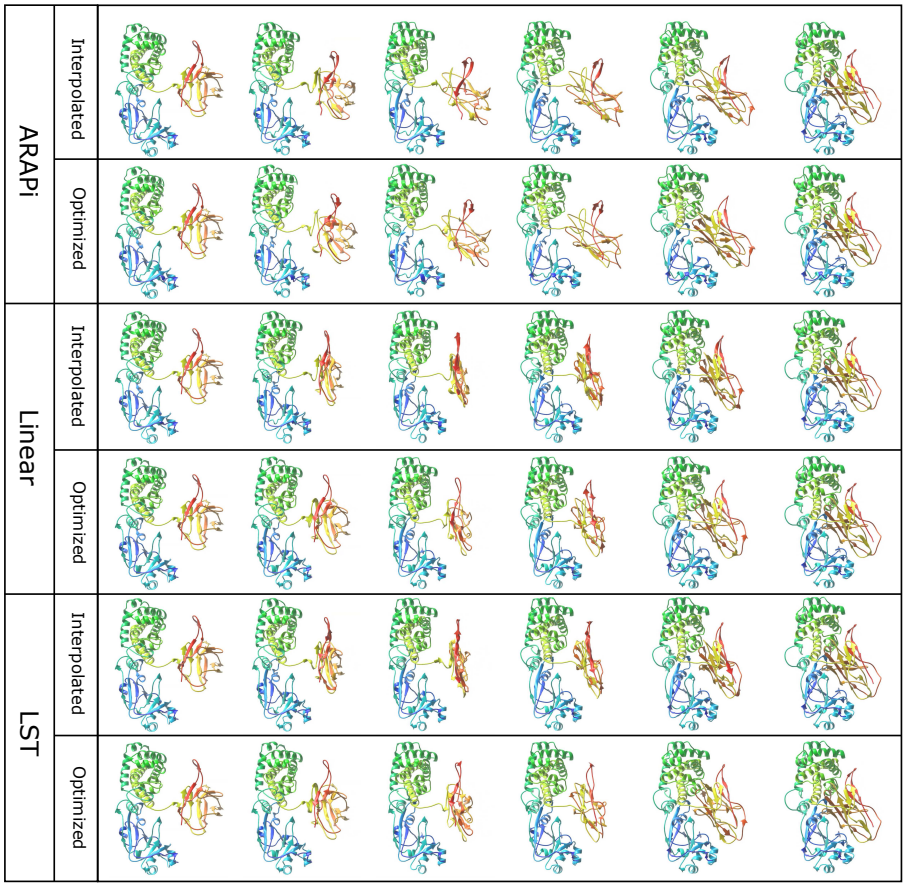

Fig. 7: The paths for Diphtheria Toxin after interpolation (with ARAPi, Linear, or LST) and after NEB optimization. The paths after clash removal are not shown because this stage does not strongly change the secondary structures. The ARAPi paths do not vary much after optimization because the ARAPi method generates paths already close to the optimized ones. Visible structural degeneration is found in the Linear and LST paths after interpolation. Thanks to the path-optimization method, this problem is reduced as seen in the optimized paths generated from the Linear and LST methods. However, the optimization cannot entirely remove the shrinkage problem, which is the source for high potential-energy barriers in the optimized Linear and LST paths.

Although the optimized ARAPi paths have lower energy barriers for all the experiments, self-intersections along the paths were detected for the three

following experiments: the 5'-Nucleotidase (Figure 8), the Dengue 2 Virus Envelope Glycoprotein (Figure 9), and the Spindle Assembly Checkpoint protein (Figure 10). This problem has two causes. First, the ARAPi method only preserves the local rigidity, and hence, does not guarantee the absence of self-intersections between two distant parts of the same structure. Second, the Slerp method used for rotation interpolation cannot generate rotation motions larger than 180 degrees, which may be necessary for certain cases such as the helix formation. In fact, the self-intersection problem is challenging to any deterministic interpolation method. The Linear and LST paths also encounter this problem for Dengue 2 Virus Envelope Glycoprotein, and Spindle Assembly Checkpoint protein as shown in Figure 9 and 10, respectively. Although the shrinkage effect in the Linear and LST paths of 5'-Nucleotidase helps to avoid this problem (see Figure 8), it raises the potential-energy barrier as discussed above for Diphtheria Toxin. Note that we also applied our method on the well-known case of Tyrosin kinase [64–67] and encountered this problem from an end-loop of the structure (see Section 3 of the Supplementary Material).

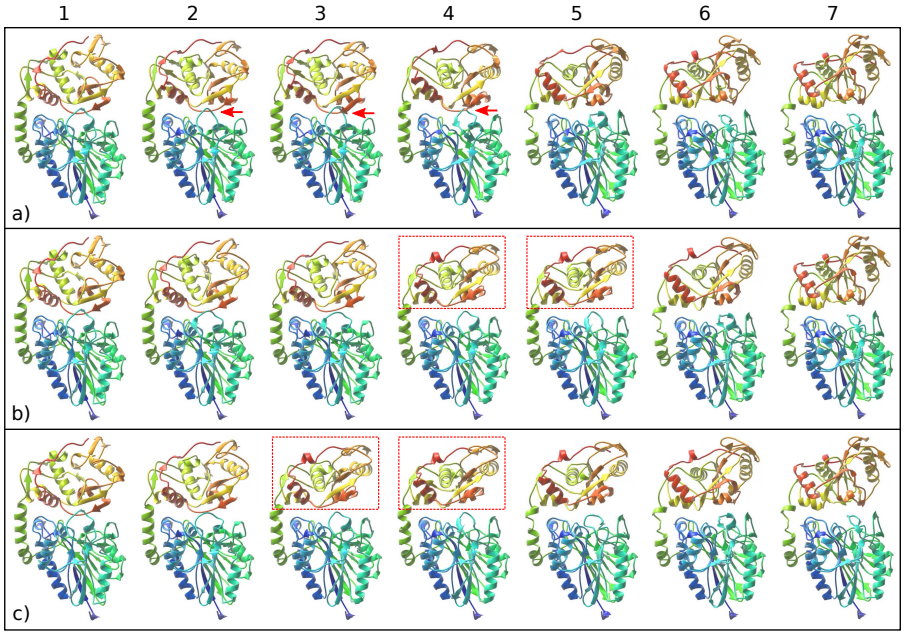

Fig. 8: Optimized paths for 5'-Nucleotidase from a) ARAPi b) Linear c) LST. Self-intersections (pointed by red arrows) are found for the optimized-ARAPi path (sequence a). The optimized Linear and LST paths do not have this problem but they have the shrinkage problem (for e.g. the protein parts in the red dashed rectangular boxes in sequences b and c are smaller in size than the same parts in the rest of the snapshots).

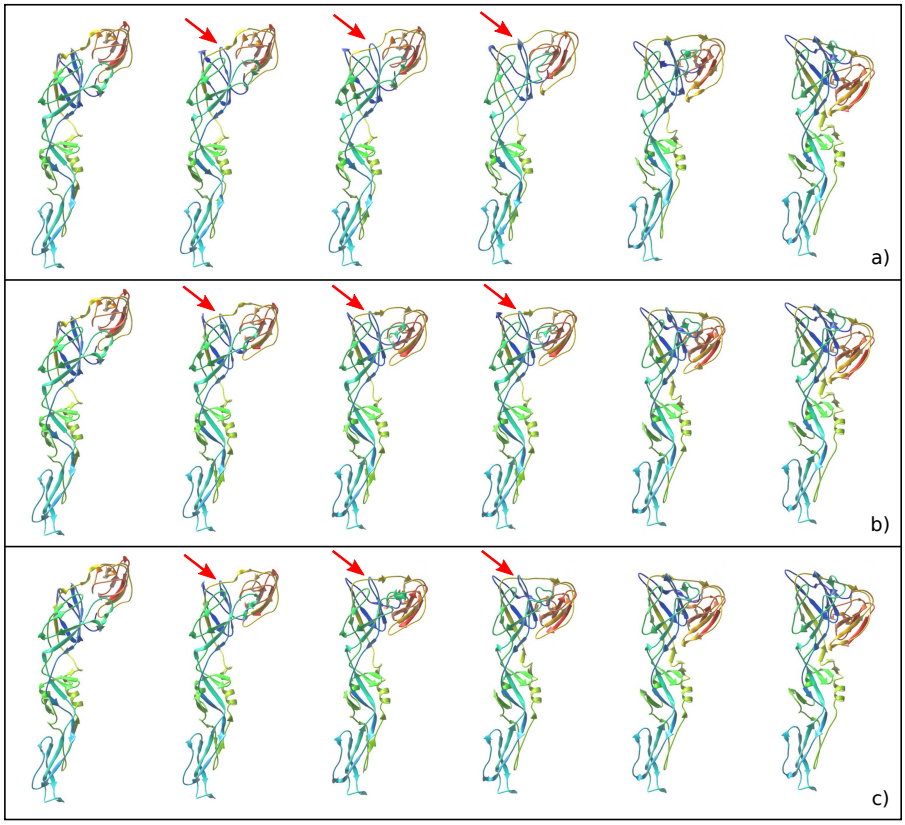

Fig. 9: Optimized paths for Dengue 2 Virus Envelope Glycoprotein from a) ARAPi b) Linear c) LST. The self-intersection problem is found in the optimized paths from all the interpolation methods, as pointed by the red arrows. For each path (each sequence), the yellow loop is in front of the blue loop in the second snapshot; it then enters the blue loop in the third snapshot; and finally escape behind the blue loop in the fourth snapshot.

3.2.6 Prediction of intermediate structures

Many methods are validated by showing their capability to predict known intermedate structures between given pairs of protein structures [20,22]. Therefore, following the same line, we applied our method for the five cases shown in Table 4 which were proposed in [22] for predicting intermediate structures. In the table, each row represents a different case of conformational transition path. For each row, the third column shows the initial and target structures, the fourth column the number of conformations generated by our method, the fifth column the intermediate structure mentioned in [22]. The final two columns show the shortest RMSD distances (considering only alpha carbons) from the

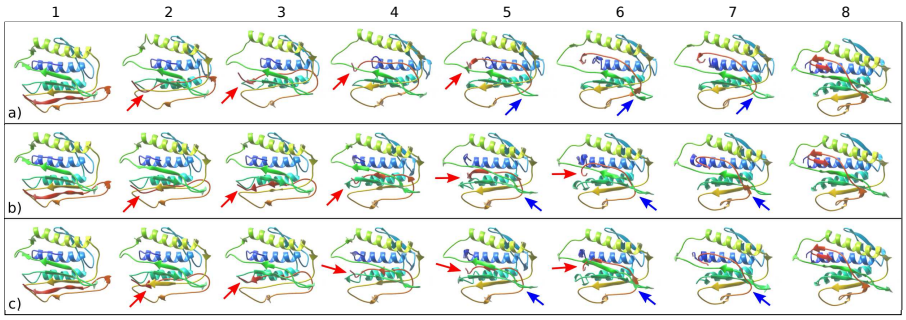

Fig. 10: Optimized paths for Spindle Assembly Checkpoint protein from a) ARAPi b) Linear c) LST. The self-intersection problem is found in the optimized paths from all the interpolation methods, as pointed by the red and blue arrows. In sequence (a), the red-end loop crosses the yellow loop (transition from a2 to a3), the cyan loop (a3 to a4), and the green loop (a4 to a5). The blue arrows in sequence (a) point to another location of self-intersection: the orange loop enters (a5 to a6) and then escapes (a6 to a7) the green loop. Similar self-intersection locations and behaviors are found for sequences b and c.

intermediate structure of our path and of the path in [22], respectively, for each case.

To generate the paths, we applied our method following the pipeline shown in Figure 1 and using the same tools and parameters presented above. However, for multiple-chains cases, applying straightforwardly the ARAP interpolation would not give valid results due to the disconnections between the chains. Therefore, for such cases, instead of taking covalent bonds in the initial structure to construct the ARAP topology, we built ARAP edges between any pair of atoms whose distance is less than 3 Å in the initial structure.

When observing the shortest RMSD distances based on alpha carbons from the intermediate structures, we found that our results and those reported in [22] are close (see the last two columns in Table 4). In particular, the lowest RMSD is only increased by 0.31 Å in the worst case. Moreover, the variation of RMSD distances along the paths we obtained has similar patterns to those obtained in [22] except for the reverse path in experiment 15 (see Section 4 of the Supplementary Material). This is because the forward and reverse paths generated by our method do not differ significantly and ARAP interpolation does not capture hysteresis phenomena. Apart from that, these results show that our method can generate paths of quality relatively similar to the paths of [22] without relying on coarse grain representation.

Table 4: Experiments for the prediction of intermediate structures.

Experiment ID	Name	Initial/Target (PDB and chain code)	Path size $\mathcal{L}$	Intermediate structures (from [22])	Shortest RMSD distance (in Å) from intermediate structures	
					in our paths	in the paths reported in [22]
13	Ribose-binding protein	1BA2(A)/2DRI(A)	48	1URP(A)	1.05	1.47
				1BA2(B)	1.04	-
				2GX6(A)	0.98	-
14	5'-Nucleotidase	1OID(A)/1HPU(A)	93	1OI8(A)	2.10	1.84
				1OID(B)	1.23	-
15	Ribonuclease III	1YYO(AB)/1YYW(AB)	170	1YZ9(AB)	4.19	4.11
				2NUG(AB)*	12.00	-
				4M30(AB)*	12.42	-
16	CA^{2+}-ATPase	2C9M(A)/1T5S(A)	128	3W5A(A)	4.85	4.58
				4H1W(A)	4.76	-
17	Ligand-gated ion channel	4NPQ(ABCDE)/4HFI(ABCDE)	42	3TLS(ABCDE)	1.36	1.05
				3TLU(ABCDE)	1.60	-
				3TLW(ABCDE)	1.48	-
				4NPP(ABCDE)	1.14	-

* This intermediate structure is considered for the reverse path.

4 Conclusion

We have presented a method to generate conformational transition paths with low potential-energy barriers for proteins. The method is composed of three stages. The first stage uses ARAPi for generating an initial path. The second stage improves the path conformations by removing steric and ring clashes. The third stage applies NEB to give a low-energy path. The method was applied on 12 protein cases. With this method, we assessed the quality of the solutions obtained with two variants where the Linear or the LST methods replace the ARAPi method in the first stage for interpolation.

The results showed that the initial paths generated with ARAPi are least prone to steric clashes although this tendency is the opposite for ring clashes. Our proposed clash remover completely removed both types of clashes. Moreover, the computational time for removing the clashes of ARAPi paths is lower than that for Linear and LST paths. The clash remover and NEB method strongly reduced the energy barriers of the paths. However, they only changed the paths locally in the case of ARAPi paths because these paths already had high quality, which rendered them close to their optimized ones. In contrast, the Linear and LST paths were very different from their optimized ones. Hence, despite a huge reduction in energy barriers, the optimized paths arising from Linear or LST had higher energy barriers than those arising from ARAPi. This means that the interpolation method had a great impact on the optimized solutions and ARAPi appeared to be the best choice for interpolation as compared with Linear and LST. We have also shown that our method can predict intermediate structures without relying on coarse grain representation. The paths obtained from our proposed method can then be used for other advanced methods such as for starting a transition path sampling or umbrella sampling to estimate free energy differences [48,49] or to predict reaction rates [50].

There are several improvements that we would like to consider for future work. First, the NEB stage is the most computationally demanding in our method, because it is currently applied for 1000 iterations. In the future, we

would like to rely on a smarter stopping criterion to terminate this minimization process such as when the energy barrier does not decrease after a certain number of iterations. Secondly, the NEB method used can miss "true" energy barriers because the paths are discrete. Hence, the methods for estimating the exact locations of the transition states such as the Dimer method [68] or the climbing-image NEB [69] could be incorporated in our approach. Finally, the proposed method only gives one path among numerous possible paths. As we have seen, even though the obtained paths have lower energy barriers, self-intersections can still be present along a path as in the case of 5'-Nucleotidase, Dengue 2 Virus Envelope Glycoprotein, Spindle Assembly Checkpoint protein, and Tyrosine kinase. This problem which makes the paths invalid would require a more global exploration of the energy landscape to find physically feasible solutions. In the future, we would like to investigate the replacement of ARAPi by an exploration-based method such as the ART-RRT method [70] to address such challenging scenarios.

5 Supplementary Material

The path energy reduction thanks to the clash remover and NEB method, as well as the deviation in bond lengths, bond angles, and dihedral angles of conformations along a path from the initial structure in the experiment for Diphtheria Toxin were commented in Section 3.2.3 and 3.2.4. The plots for these results can be found in Section 1 and 2 of the Supplementary Material. The self-intersection problem encountered by our path for the case of Tyrosine kinase is shown in Section 3 of the Supplementary Material. For the prediction of intermediate structures in Section 3.2.6, Section 4 of the Supplementary Material also provides the plots describing for our paths the variation of RSMD distances from intermediate structures.

References

1. A. Korkut, W.A. Hendrickson, Proceedings of the national academy of sciences **106**(37), 15673 (2009)
2. M. Karplus, J.A. McCammon, Nature Structural and Molecular Biology **9**(9), 646 (2002)
3. M.K. Kim, R.L. Jernigan, G.S. Chirikjian, Biophys J **83** (2002). DOI 10.1016/S0006-3495(02)73931-3
4. B.J. Alder, T. Wainwright, The Journal of Chemical Physics **31**(2), 459 (1959)
5. N. Metropolis, S. Ulam, Journal of the American statistical association **44**(247), 335 (1949)
6. C. Dellago, P.G. Bolhuis, *Transition Path Sampling and Other Advanced Simulation Techniques for Rare Events* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2009), pp. 167–233
7. M.R. So/rensen, A.F. Voter, The Journal of Chemical Physics **112**(21), 9599 (2000)
8. S.P. Brooks, B.J. Morgan, The Statistician pp. 241–257 (1995)
9. D.J. Wales, H.A. Scheraga, Science **285**(5432), 1368 (1999)
10. K.A. Fichthorn, S. Mubin, Computational Materials Science **100**, 104 (2015)
11. J. Schlitter, M. Engels, P. Krüger, E. Jacoby, A. Wollmer, Molecular Simulation **10**(2-6), 291 (1993)

12. S. Izrailev, S. Stepaniants, B. Isralewitz, D. Kosztin, H. Lu, F. Molnar, W. Wriggers, K. Schulten, in *Computational molecular dynamics: challenges, methods, ideas* (Springer, 1999), pp. 39–65
13. G.J. Kleywegt, T.A. Jones, Structure **3**(6), 535 (1995)
14. G.J. Kleywegt, T.A. Jones, Structure **4**(12), 1395 (1996)
15. A.G. Booth, Journal of Molecular Graphics and Modelling **19**(6), 481 (2001)
16. T.A. Halgren, W.N. Lipscomb, Chemical Physics Letters **49**(2), 225 (1977)
17. N. Echols, D. Milburn, M. Gerstein, Nucleic Acids Research **31**(1), 478 (2003)
18. S. Wells, S. Menor, B. Hespenheide, M. Thorpe, Physical Biology **2**(4), S127 (2005)
19. D.W. Farrell, K. Speranskiy, M. Thorpe, Proteins: Structure, Function, and Bioinformatics **78**(14), 2908 (2010)
20. D.R. Weiss, M. Levitt, Journal of molecular biology **385**(2), 665 (2009)
21. N.E. Castellana, A. Lushnikov, P. Rotkiewicz, N. Sefcovic, P.A. Pevzner, A. Godzik, K. Vyatkina, Algorithms for Molecular Biology **8**(1), 1 (2013). DOI 10.1186/1748-7188-8-19
22. L. Orellana, O. Yoluk, O. Carrillo, M. Orozco, E. Lindahl, Nature Communications **7**, 12575 (2016). URL http://dx.doi.org/10.1038/ncomms12575
23. P. Koehl, The Journal of Chemical Physics **145**(18), 184111 (2016). DOI 10.1063/1.4966974. URL https://doi.org/10.1063/1.4966974
24. J. Franklin, P. Koehl, S. Doniach, M. Delarue, Nucleic Acids Research **35**(suppl_2), W477 (2007). DOI 10.1093/nar/gkm342. URL http://dx.doi.org/10.1093/nar/gkm342
25. O. Miyashita, J.N. Onuchic, P.G. Wolynes, Proceedings of the National Academy of Sciences of the United States of America **100**(PMC240658), 12570 (2003). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC240658/
26. J.K. Bray, D.R. Weiss, M. Levitt, Biophysical journal **101**(12), 2966 (2011)
27. J.R. López-Blanco, J.I. Aliaga, E.S. Quintana-Ortí, P. Chacón, Nucleic acids research **42**(W1), W271 (2014)
28. D.M. Krüger, A. Ahmed, H. Gohlke, Nucleic acids research **40**(W1), W310 (2012)
29. A. Hoffmann, S. Grudinin, J. Chem. Theory Comput. **13**(5), 2123 (2017). DOI 10.1021/acs.jctc.7b00197. URL https://doi.org/10.1021/acs.jctc.7b00197
30. W. Zheng, M. Tekpinar, in *Protein Dynamics: Methods and Protocols*, ed. by D.R. Livesay (Humana Press, Totowa, NJ, 2014), pp. 159–172. URL https://doi.org/10.1007/978-1-62703-658-0_9
31. P. Maragakis, M. Karplus, Journal of Molecular Biology **352**(4), 807 (2005). URL http://www.sciencedirect.com/science/article/pii/S0022283605008193
32. M. Delarue, P. Koehl, H. Orland, The Journal of Chemical Physics **147**(15), 152703 (2018). DOI 10.1063/1.4985651. URL https://doi.org/10.1063/1.4985651
33. A. Das, M. Gur, M.H. Cheng, S. Jo, I. Bahar, B. Roux, PLoS Computational Biology **10**(PMC3974643), e1003521 (2014). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3974643/
34. S. Takada, Current Opinion in Structural Biology **22**(2), 130 (2012). DOI https://doi.org/10.1016/j.sbi.2012.01.010. URL http://www.sciencedirect.com/science/article/pii/S0959440X12000280. Theory and simulation/Macromolecular assemblages
35. S. Kmiecik, J. Wabik, M. Kolinski, M. Kouza, A. Koliński, *Coarse-Grained Modeling of Protein Dynamics* (Springer Berlin Heidelberg, 2014), vol. 1, pp. 55–79. DOI 10.1007/978-3-642-28554-7_3
36. M.K. Nguyen, L. Jaillet, S. Redon, Journal of Computer-Aided Molecular Design pp. 1–15 (2017). DOI 10.1007/s10822-017-0012-y. URL http:
37. H. Jónsson, G. Mills, K.W. Jacobsen, *Nudged elastic band method for finding minimum energy paths of transition* (World Scientific, 1998), chap. 16, pp. 385–404
38. E. Weinan, W. Ren, E. Vanden-Eijnden, Physical Review B **66**(5), 052301 (2002)
39. S. Fischer, M. Karplus, Chemical Physics Letters **194**(3), 252 (1992). URL http://www.sciencedirect.com/science/article/pii/000926149285543J
40. A.C. Pan, D. Sezer, B. Roux, The journal of physical chemistry. B **112**(PMC2757167), 3432 (2008). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2757167/
41. S. Fischer, K.W. Olsen, K. Nam, M. Karplus, Proceedings of the National Academy of Sciences of the United States of America **108**(PMC3078355), 5608 (2011). URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3078355/

42. S. Smidstrup, A. Pedersen, K. Stokbro, H. Jónsson, The Journal of chemical physics **140**(21), 214106 (2014)
43. M.J. Abraham, T. Murtola, R. Schulz, S. Páll, J.C. Smith, B. Hess, E. Lindahl, SoftwareX **1**, 19 (2015)
44. A.D. MacKerell Jr, D. Bashford, M. Bellott, R.L. Dunbrack Jr, J.D. Evanseck, M.J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, et al., The journal of physical chemistry B **102**(18), 3586 (1998)
45. D.A. Case, T.E. Cheatham, T. Darden, H. Gohlke, R. Luo, K.M. Merz, A. Onufriev, C. Simmerling, B. Wang, R.J. Woods, Journal of computational chemistry **26**(16), 1668 (2005)
46. W.R. Scott, P.H. Hünenberger, I.G. Tironi, A.E. Mark, S.R. Billeter, J. Fennen, A.E. Torda, T. Huber, P. Krüger, W.F. van Gunsteren, The Journal of Physical Chemistry A **103**(19), 3596 (1999)
47. D.J. Wales, T.V. Bogdan. Potential energy and free energy landscapes (2006)
48. G.M. Torrie, J.P. Valleau, Journal of Computational Physics **23**(2), 187 (1977)
49. C. Dellago, in *Free Energy Calculations: Theory and Applications in Chemistry and Biology*, ed. by C. Chipot, A. Pohorille, 1st edn. (Springer-Verlag Berlin Heidelberg, 2007), pp. 249–276
50. C. Dellago, P.G. Bolhuis, F.S. Csajka, D. Chandler, The Journal of Chemical Physics **108**(5), 1964 (1998)
51. O. Sorkine, M. Alexa, in *Symposium on Geometry processing*, vol. 4 (2007), vol. 4
52. K. Shoemake, in *ACM SIGGRAPH computer graphics*, vol. 19 (ACM, 1985), vol. 19, pp. 245–254
53. G. Henkelman, H. Jónsson, The Journal of chemical physics **113**(22), 9978 (2000)
54. E. Bitzek, P. Koskinen, F. Gähler, M. Moseler, P. Gumbsch, Physical review letters **97**(17), 170201 (2006)
55. INRIA. SAMSON: Software for Adaptive Modeling and Simulation Of Nanosystems. Version 0.6.0. (2017). URL https://www.samson-connect.net
56. C. Peng, H. Bernhard Schlegel, Israel Journal of Chemistry **33**(4), 449 (1993)
57. A. Behn, P.M. Zimmerman, A.T. Bell, M. Head-Gordon, The Journal of chemical physics **135**(22), 224108 (2011)
58. A. Behn, P.M. Zimmerman, A.T. Bell, M. Head-Gordon, Journal of chemical theory and computation **7**(12), 4019 (2011)
59. H.M. Berman, T. Battistuz, T.N. Bhat, W.F. Bluhm, P.E. Bourne, K. Burkhardt, Z. Feng, G.L. Gilliland, L. Iype, S. Jain, P. Fagan, J. Marvin, D. Padilla, V. Ravichandran, B. Schneider, N. Thanki, H. Weissig, J.D. Westbrook, C. Zardecki, Acta Crystallographica Section D **58**, 899 (2002)
60. A. Šali, T.L. Blundell, Journal of molecular biology **234**(3), 779 (1993)
61. C.C. Huang, G.S. Couch, E.F. Pettersen, T.E. Ferrin, in *Pac. Symp. Biocomput*, vol. 1 (World Scientific, 1996), vol. 1, p. 724
62. N. Guex, M.C. Peitsch, electrophoresis **18**(15), 2714 (1997)
63. J.A. Lemkul, GROMACS Tutorials (2017)
64. D. Shukla, Y. Meng, B. Roux, V.S. Pande, Nature communications **5**, 3397 (2014)
65. C.F. Wong, Protein Science **25**(1), 192 (2016)
66. M. Fajer, Y. Meng, B. Roux, The Journal of Physical Chemistry B **121**(15), 3352 (2016)
67. H.J. Yoon, S. Lee, S.J. Park, S. Wu, Scientific reports **8**(1), 5673 (2018)
68. G. Henkelman, H. Jónsson, The Journal of chemical physics **111**(15), 7010 (1999)
69. G. Henkelman, B.P. Uberuaga, H. Jónsson, The Journal of chemical physics **113**(22), 9901 (2000)
70. M.K. Nguyen, L. Jaillet, S. Redon, Journal of Computational Chemistry (2018). DOI 10.1002/jcc.25132