

RESEARCH

Open Access



Quantifying steric hindrance and topological obstruction to protein structure superposition

Peter Røgen*

Abstract

Background: In computational structural biology, structure comparison is fundamental for our understanding of proteins. Structure comparison is, e.g., algorithmically the starting point for computational studies of structural evolution and it guides our efforts to predict protein structures from their amino acid sequences. Most methods for structural alignment of protein structures optimize the distances between aligned and superimposed residue pairs, i.e., the distances traveled by the aligned and superimposed residues during linear interpolation. Considering such a linear interpolation, these methods do not differentiate if there is room for the interpolation, if it causes steric clashes, or more severely, if it changes the topology of the compared protein backbone curves.

Results: To distinguish such cases, we analyze the linear interpolation between two aligned and superimposed backbones. We quantify the amount of steric clashes and find all self-intersections in a linear backbone interpolation. To determine if the self-intersections alter the protein's backbone curve significantly or not, we present a path-finding algorithm that checks if there exists a self-avoiding path in a neighborhood of the linear interpolation. A new path is constructed by altering the linear interpolation using a novel interpretation of Reidemeister moves from knot theory working on three-dimensional curves rather than on knot diagrams. Either the algorithm finds a self-avoiding path or it returns a smallest set of essential self-intersections. Each of these indicates a significant difference between the folds of the aligned protein structures. As expected, we find at least one essential self-intersection separating most unknotted structures from a knotted structure, and we find even larger motions in proteins connected by obstruction free linear interpolations. We also find examples of homologous proteins that are differently threaded, and we find many distinct folds connected by longer but simple deformations. TM-align is one of the most restrictive alignment programs. With standard parameters, it only aligns residues superimposed within 5 Ångström distance. We find 42165 topological obstructions between aligned parts in 142068 TM-alignments. Thus, this restrictive alignment procedure still allows topological dissimilarity of the aligned parts.

Conclusions: Based on the data we conclude that our program `ProteinAlignmentObstruction` provides significant additional information to alignment scores based solely on distances between aligned and superimposed residue pairs.

Keywords: Protein topology, Structural classification of proteins, Self-avoiding morphs

Mathematic subject classification: Primary 92C40, Secondary 92E10, 57K10

Background

Proteins are essential cellular tools and as macroscopic tools, some shapes are preferential for fulfilling specific functions. Protein shapes are known to be more preserved than their sequences of amino acids: structural comparison of proteins is therefore an important and

*Correspondence: prog@dtu.dk
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Artillerivej 322, Kongens Lyngby, Denmark



© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>. The Creative Commons Public Domain Dedication waiver (<http://creativecommons.org/publicdomain/zero/1.0/>) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

active area of research with new structural alignment methods reported to double every five years for three decades [1]. Aside from the rapid growth of the number of known protein structures, protein structure comparison is challenging for several reasons. One is that even for a fixed sequence of amino acids, some proteins are highly flexible [2]. Another is that mutations cause plastic deformation that comparison methods should also take into account [1]. On the other hand, proteins reuse the same types of folds, but with the growing number of known protein structures, the global view is changing from discrete folds into considering larger parts of fold space as a continuum [3].

In protein structure comparison either the shapes of proteins' solvent accessible surfaces [4, 5] or their folded backbones are compared. Surface comparison mainly address how a protein is seen from and possibly interacts with its exterior. Chain or curve comparison mainly address how a protein is folded upon itself and is typically used when searching structural evolutionary evidence [6] and when judging efforts to predict protein structures [7]. One exception to this general trend is that surfaces of protein models generally can be discriminated from surfaces of native proteins [8]. It is not hard to imagine two long protein chains sharing a surface but being folded differently inside the surface. Here, we therefore only consider structural comparison based on curve comparison.

Most methods for pairwise curve comparison of protein structures combine methods for searching alignment and optimal superposition of subsets of two proteins with a score function assessing the quality of this alignment. See [1] for an overview of the most used methods. All these score functions are based on distances, and it is our hypothesis that in practice they are blind to topological changes even in the aligned parts as we now explain. Each score function is based on the distances that aligned residues are moved when the structures are in optimal superposition. But its not taken into account if the motion may be performed by a continuous deformation or if it will lead to self-intersections of the protein that may change its fold or even tie a knot in it. Mathematically that is, the score functions of most alignment and superposition methods report similarity or distances in the space of immersions of proteins. Theoretically, we will argue, a pair of topologically distinct embeddings of a protein may have a similarity score typically found between highly similar protein structures. This is apparently also found in practice as [9] states "*These classic structure comparison metrics*" (Root Mean Square Deviation (RMSD) [10], Global Distance Test-Total Score (GDT-TS) [7], and TM-score [11]) "*need to be supplemented by more sophisticated measures, which quantify*

topographical differences in chain progression in 3D space".

Superposition free alignment methods based on changes in internal distances such as DALI [12], FlexE [13] and IDDT [14] also do not check for topological changes. Mathematically they cannot tell a structure from its mirror image. More interestingly, local chirality changes can occur under minor changes to internal distances. And the same is true for the type of apparent topological incorrectness reported in [9] among their top 400 distance constrained models, "*with the polypeptide chain passing through loops in a way that is, according to visual intuition, atypical of fully correct structures*".

There are methods for protein structure comparison that do take topological features of proteins into account. Except for [15], none of these are alignment methods. Each of these methods first produces a global description of each protein chain in the form of a set of descriptor values. Protein chains are subsequently compared by comparing their descriptor values. The number of proteins and not the number of protein pairs dominate the calculation time. Hence, descriptor-based methods are very fast on large data sets as they neither need nor provide a structural alignment of sub structures. The first method to distinguish different threading of protein models was by calculating the writhe of each model [16]. The writhe is a signed measure of how coiled a space curve is and jumps discontinuously by ± 2 when a curve passes through itself once. In [16] Michael Levitt reported a writhe separation between differently threaded models. However, two self-intersections may have canceling writhe jumps, and also continuous deformations change the writhe; thus, the writhe cannot separate folds by itself. The four distinct points involved in two self-intersections on an open directed curve form one of four orderings. Similarly, three self-intersections define 15 orderings. Generalizations of the writhe called Generalized Gauss integrals used in [17] are constructed such that at least one of these descriptors jumps discontinuously [18] in each of these $1 + 4 + 15$ cases. The set of descriptors introduced in [19] also has this property. The generalized Gauss integrals are powerful enough to tell all fold classes in the protein classification CATH version 2.4 [17] apart, and they can be normalized to have desired metric properties [18]. However, as in the case of writhe, these types of descriptors are mathematically not powerful enough to tell any two configurations apart due to the dimension reduction taking place. As they apparently manage to separate protein folds, it is here more interesting that such global descriptors do not point out where along the backbone two structures are topologically different—the descriptor values are just significantly different. A similar remark holds for the topological

descriptors based on fat graphs [20, 21], alpha shape [22] and on the average occurrences of all patterns of 3 crossings in planar projections of a protein backbone [19]. Not all descriptor-based methods are directly sensitive to topology. Examples are the method with the most accurate re-classification of protein structures reported, Frag-Bag [23], and the distance excess-based descriptors [24]. The later method however presents an elegant solution to the problem of detecting (plastic) deformation as caused by mutations. The exact differential form underlying the descriptors makes them path independent such that they e.g. are invariant under any deformation of a loop in a protein that preserves the loops terminal points.

The effort put into detecting special topological features in a single protein structure illustrates the difficulty in detecting topological changes between pairs of protein structures. Knots in proteins [25, 26] have been studied for two decades [27, 28]. And lately also lassos [29] and links [30] in proteins have been investigated. The discrete nature of Monte Carlo sampling allows the protein backbone to pass through itself between samples, but in protein configuration space such a jump corresponds to a long and unlikely motion. Thus, in protein structure prediction, Monte Carlo sampling produces more knotted models than found in real proteins. The reader is encouraged to read the well told story [31] about the efforts taken both by modelers and assessors in the series of Critical Assessment of techniques for protein Structures Prediction (CASP) to avoid knotted and slip-knotted structures that in some cases have been the first model. The algorithm Pokefind [31] detects if a disk spanned by a shorter closed loop in a protein structure is penetrated by another part of the chain. In a protein model, this local non-protein like topological feature is likely to be a modeling error caused by the backbone passing through itself. Such topological model errors are apparently not detected by current structural alignment methods as [31] reports that neither the presence of knots nor of pokes in protein models show any significant correlation to the native-model distance measured by Global Distance Test (GDT). [31] concludes by suggesting adding Pokefind and Knotfind to the current metric. Their suggested method will tell if the number of knots and pokes differs between two protein structures. Hence, one would know that there is at least one topological obstruction between a knotted and an unknotted protein and at least one obstruction between a poked and a poke free structure. But topological obstructions separating differently folded structures will not be detected as long as knots and pokes are unchanged; most likely, by being absent in both structures. We conclude that no current algorithm can guarantee to find topological obstructions to a structural alignment and superposition of one protein structure

onto another and present novel algorithms `ProteinAlignmentObstruction` for finding such obstructions and for evaluating how severe they are.

All protein backbones form open curves and are produced by extrusion. We here ignore the covalent links found when including disulfide bridges [30]. Hence, combining a complete unfolding of the first structure with a folding of the second from its complete unfolded stage gives a continuous self-avoiding path between two arbitrary protein structures. Thus, mathematically all protein chains share topology class and any sub-division of protein configuration space into folds depend on human judgement and is likely to depend on the specific structural biological context. Consider a pair of shoes with almost identical shoelaces except that the one contains a left-handed trefoil knot and the other a right-handed trefoil knot. In the space of immersions, they are very close, and any structural alignment program will find that, but in the space of embeddings, they are not close meaning no short self-avoiding morph between them exists. In structural biology terms, these two shoelaces share architecture, as the overall building blocks are very similar. They may or may not belong to different folds, also referred to as topology classes, depending if their different threading is recognized and considered important. But the folding paths have to be very different which may not be recognized. To address the difference between immersed and self-avoiding path-length `ProteinAlignmentObstruction` starts from the alignment and superposition provided by an alignment program and considers the linear interpolation between the two protein structures implied by the alignment score. First, it provides a novel measure of the amount of steric overlap caused by the morph. Next, it detects all self-intersections that occur during the interpolation and determines which are avoidable by moves similar to knot theory's Reidemeister moves within a user defined tubular neighborhood of the original linear morph. Next, it solves the problem of avoiding the maximal number of self-intersections at the shortest additional morph-length. Either it finds a self-avoiding morph close to the original linear interpolation or it labels the remaining self-intersections essential, as they cannot be resolved sufficiently close to the original interpolation. Finally if wanted, the length of a self-avoiding morph that resolves these essential self-intersections but gets far from the original alignment is found. In this work, the focus is to make a fast algorithm that can detect steric and topological obstruction to a structural alignment and superposition of two protein structures at a calculation time compatible to that of a structural alignment program. The length of the self-avoiding morph provided here is

at best a crude approximate of a shortest self-avoiding morph-length; a calculation that lies outside the scope of this paper as this author expect it to be too slow for practical application in pairwise protein structure comparison. Also it is not the aim to find geometrically or energetically plausible paths as e.g. considered in [32]. Conceptually, the main aim is to introduce homotopy of curve embeddings and Reidemeister moves to the field of computational structural biology as spoken for in [33].

The description of algorithms presented in the following section assumes no prior knowledge of knot theory and homotopy theory. It is somewhat lengthy, occasionally involved and may be skipped by readers not interested in the method development. Such readers should read the motivating example below and from there jump to the results section.

Methods

We start by estimating how similar two differently threaded tube structures fulfilling typical distance inequalities of alpha carbon atoms in proteins can be when measured by Root Mean Square Deviation (RMSD) [10], Global Distance Test - Total Score (GDT-TS) [7], and TM-score [11].

A motivating example

Alpha carbon atom neighbors along the backbone are usually 3.8Å (Ångström) apart. Otherwise, alpha carbon pairs are rarely less than 4.5Å apart. Imagine a

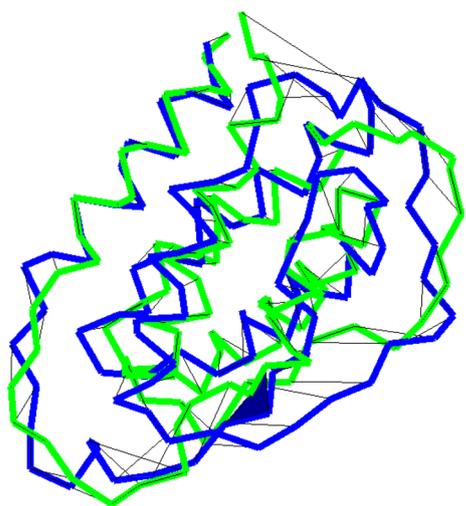


Fig. 1 A self-intersection indicated by a black face in the morph between the alpha carbon curves of CATH2.4 domains 1csgA0 and 1-j1i00. The two domains share the homology-class 1.20.120.200

symmetrical crossing in the xy -plane where the over-crossing part of a chain has heights (measured on the z -axis) 0, 0, 1.25, 2.5, 2.5, 1.25, 0, 0Å. Similarly, the under-crossing part follows the y -axis at heights 0, 0, -1.25, -2.5, -2.5, -1.25, 0, 0Å. Consider two larger structures that are identical except that the above over- and under-crossings are interchanged. The Euclidean motion of the optimal superposition of two structures is close to the identity, which we assume for simplicity. The total motion is thus given by 4 residues traveling 5Å and 4 residues traveling 2.5Å. For an n -residue structure we find a topology change (tc) for

$$\begin{aligned} RMSD_n^{tc} &\leq \sqrt{\frac{4(5\text{\AA})^2 + 4(2.5\text{\AA})^2}{n}} < \frac{12\text{\AA}}{\sqrt{n}} \\ GDT - TS_n^{tc} &\geq \frac{1}{4} \sum_{m=1,2,4,8} \text{aligned residues within } m\text{\AA} \\ &= \frac{1}{4} \left(\frac{n-8}{n} + \frac{n-8}{n} + \frac{n-4}{n} + \frac{n}{n} \right) = 1 - \frac{5}{n} \\ TM_n^{tc} &\geq \frac{1}{n} \sum_{i=1}^n \frac{1}{1 + \left(\frac{d_i}{d_0}\right)^2} \\ &= \frac{n-8}{n} + \frac{4}{n} \frac{1}{1 + \left(\frac{5}{d_0}\right)^2} + \frac{4}{n} \frac{1}{1 + \left(\frac{2.5}{d_0}\right)^2} \\ &> 1 - \frac{4}{n}, \text{ for } n \geq 100, \end{aligned}$$

where $d_0 = 1.24(n-15)^{1/3} - 1.8$. For $n = 100$ we find $RMSD_{100}^{tc} = 1.2\text{\AA}$, $GDT - TS_{100}^{tc} = 0.95$ and $TM_{100}^{tc} = 0.96$. These scores are typical for very similar structures despite the constructed topology change. The persistence length of native proteins may require longer deformations than used in this example; but in computational modeling when, e.g., trying to minimize the distance between a model and a native structure one may get close to this example. Figure 1 shows a similar crossing change between two native 112 residue structures aligned with global RMSD 5.6Å and TM= 0.60. Before describing a method to detect such topological changes during a linear interpolation between structures we start with an algorithm for detecting steric clashes under a linear interpolation.

Measuring protein-protein overlap under a linear interpolation

Let two protein chains be represented by the coordinates of their alpha carbon atoms $P_0 = [p_0^1 \dots p_0^n]$ and $P_1 = [p_1^1 \dots p_1^n]$ each with index starting at the N-terminus. Assume they are placed in optimal superposition by RMSD or any other alignment method. We study the linear interpolation where each point $p^i(t) = (1-t)p_0^i + tp_1^i$ traverses the straight line

segment connecting the i 'th alpha carbon in each of the two structures and the parameter $t \in [0, 1]$ may be thought of as a time parameter. We want to check if typical protein distance constraints are violated during the interpolation and quantify how much the protein overlaps itself. For this, we first find the minimal distance between p^i and p^j during the interpolation. Let $p_0^{ij} = p_0^j - p_0^i$, $p_1^{ij} = p_1^j - p_1^i$ and denote $\|p_0^{ij}\| = a$, $\|p_1^{ij}\| = b$, and $p_0^{ij} \cdot p_1^{ij} = ab \cos \theta$. For all t we have

$$d_{ij}^2(t) = \|p^j(t) - p^i(t)\|_2^2 = \|(1-t)p_0^{ij} + tp_1^{ij}\|_2^2 = (1-t)^2 a^2 + t^2 b^2 + 2t(1-t)ab \cos \theta.$$

The global minimum of d_{ij}^2 is found for $t^* = \frac{a^2 - ab \cos \theta}{a^2 + b^2 - 2ab \cos \theta}$ and equals

$$m = \frac{\sin \theta a^2 b^2}{a^2 + b^2 - 2ab \cos \theta} = \frac{\|p_1^{ij} \times p_0^{ij}\|_2^2}{\|p_1^{ij} - p_0^{ij}\|_2^2} = \frac{\|(p_1^{ij} - p_0^{ij}) \times p_0^{ij}\|_2^2}{\|p_1^{ij} - p_0^{ij}\|_2^2}.$$

The last equation shows that the fraction is bounded by $\|p_0^{ij}\|_2^2$ and by symmetry also by $\|p_1^{ij}\|_2^2$. Setting $\tilde{t} = \{0, \text{if } t^* < 0; t^*, \text{if } 0 \leq t^* \leq 1; \text{and } 1, \text{if } 1 < t^*\}$, the minimum of d_{ij}^2 for $t \in [0, 1]$ denoted $d_{ij}^{2, \text{interpolation}} = d_{ij}^2(\tilde{t})$. If $\|p_1^{ij} - p_0^{ij}\|_2^2$ is vanishing t^* is ill-defined; but then d_{ij}^2 is constant and this value is returned. From a set of native protein structures, we estimate the shortest possible distances between alpha carbons i and j to be $d_{\min}(|i-j|) = 2.8, 4.5, 3.86, 3.47, 3.52, 3.48, 3.6\text{\AA}$ for $|i-j| = 1, \dots, 7$ and $d_{\min}(|i-j|) = 3.7\text{\AA}$ for $|i-j| > 7$. The overlap, overlap_{ij} , is thus the max of $d_{\min}(|i-j|) - d_{ij}^{\text{interpolation}}$ and 0(zero). The mean overlap is given by $\text{MeanOverlap}(P_1, P_2) = \frac{1}{n} \sum_{i < j} \text{overlap}_{ij}$. Especially in α -helices the distance between neighbouring alpha carbon atoms may be shortened substantially during the interpolation. Therefore $d_{\min}(1) = 2.8$ is chosen small to avoid large contributions from local deformations. If both protein structure P_0 and P_1 are overlap free, overlap_{ij} is given by the max of $d_{\min}(|i-j|) - \sqrt{m}$ and 0(zero) if $0 < t^* < 1$ and 0(zero) otherwise. To be able to handle models not fulfilling basic distance constraints, we do not make this assumption.

It is a priori not clear how to penalize protein-protein overlap. Here, we have chosen to penalize pairwise overlap linearly.

Topology check of a protein under a linear interpolation

We now leave the volumetric point of view and turn to a purely topological point of view. Given two superimposed protein structures, we consider the linear interpolation

of the piecewise linear curve connecting the alpha carbons. Let P denote the piecewise linear curve through all points $p_i, i = 1, \dots, n$. That is $P_a = p_i$ for $a = 1, \dots, n$, but P_a is defined for all $a \in [1, n]$. The linear interpolation between two piecewise linear curves P^0 and P^1 is given by $P_a(t) = (1-t)P_a^0 + tP_a^1$ for $t \in [0, 1]$ and $a \in [1, n]$.

We want to detect all self-intersections of the backbone curve during the linear interpolation. Furthermore, we want to identify all self-intersections we can avoid by changing the initial morph locally by rearranging at most a user defined number of residues. The remaining self-intersections are expected to do essential changes to the fold, e.g., as avoiding them require larger rearrangements than allowed by the user. To do this, we borrow some notation from knot theory concerning knots on closed curves in 3-space. For a knot in 3-space a planar projection with over and under crossings indicated is called a knot diagram if all crossings are isolated. Hence, at most two points on the knot are allowed to be projected to the same point. When changing the plane of projection the crossings will move, crossings may disappear, new crossings may appear, and in the planar projection existing crossings may pass through each other. The same happens to the planar projection when deforming the knot through a self-avoiding morph. A fundamental result found in any text book on knot theory states that two closed curves share knot type, i.e., the one may be morphed into the other through imbeddings by a ambient isotopy if and only if their knot diagrams are connected by a finite series of Reidemeister moves, shown in Fig. 2, plus deformations not changing crossings. Crossings of a knot may be associated with a sign using the usual right-hand rule explained in the same figure. Note, that changing the direction of traversing a knot does not change the sign of the crossing. The sign of a crossing comes from the orientation of 3-space and is for line segments $P_i P_{i+1}$ and $P_j P_{j+1}$ given by the sign of the determinant $\det_{ij} = \det(P_{i+1} - P_i, P_{j+1} - P_j, P_i - P_j)$.

Detecting transversal self-intersections

We consider only so-called transversal self-intersections where two line segments actually go through each other and consequently the sign of their crossing changes at the time of the self-intersection. Algorithmically, we thus first find isolated real roots $t^* \in [0, 1]$ for the 3'rd order polynomial

$$\det_{ij}(t) = \det(P_{i+1}(t) - P_i(t), P_{j+1}(t) - P_j(t), P_i(t) - P_j(t)).$$

For each of these roots, $t^* \in [0, 1]$, the line segments $P_i(t^*)P_{i+1}(t^*)$ and $P_j(t^*)P_{j+1}(t^*)$ lie in a plane and intersect only if the planar problem

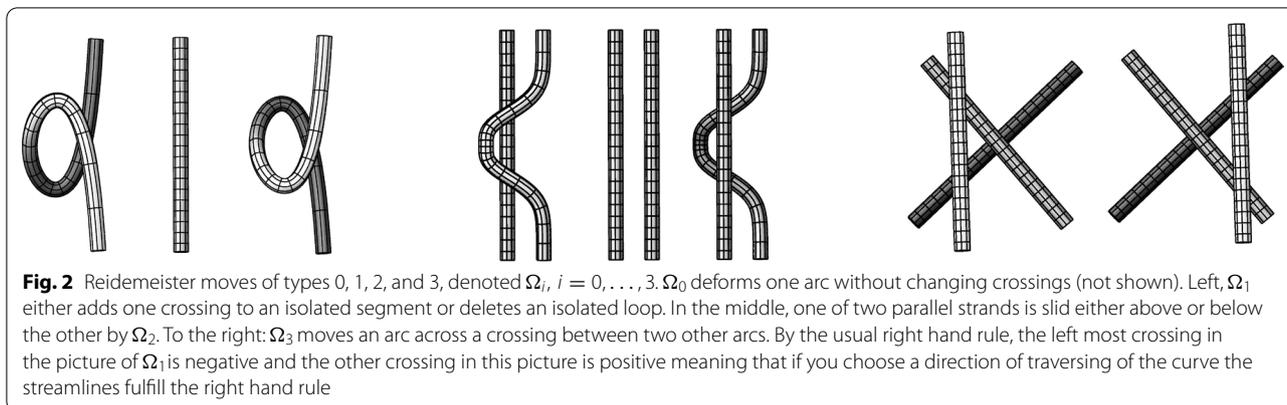


Fig. 2 Reidemeister moves of types 0, 1, 2, and 3, denoted $\Omega_i, i = 0, \dots, 3$. Ω_0 deforms one arc without changing crossings (not shown). Left, Ω_1 either adds one crossing to an isolated segment or deletes an isolated loop. In the middle, one of two parallel strands is slid either above or below the other by Ω_2 . To the right: Ω_3 moves an arc across a crossing between two other arcs. By the usual right hand rule, the left most crossing in the picture of Ω_1 is negative and the other crossing in this picture is positive meaning that if you choose a direction of traversing of the curve the streamlines fulfill the right hand rule

$$(1 - s_i)P_i(t^*) + s_iP_{i+1}(t^*) = (1 - s_j)P_j(t^*) + s_jP_{j+1}(t^*)$$

has a solution with $s_i, s_j \in [0, 1]$. If a self-intersection is found, we store the interpolation parameter t^* , the curve parameters of the self-intersection $a = i + s_i$ and $b = j + s_j$, together with the crossings sign change given by the sign of the derivative of $\det_{ij}(t)$ at t^* . By checking all pairs of line segments we get a list of self-intersections with data $(a_k, b_k, sign_k, t_k^*)$, where k is an arbitrary index.

A self-intersection between line segments P_iP_{i+1} and P_jP_{j+1} requires that at least some of the spheres centered at P_i, P_{i+1}, P_j and P_{j+1} are overlapping during the interpolation as calculated in the previous section. The self-intersection check may be skipped if the alpha carbon neighbors are less than 4Å apart and $\text{overlap}_{i,j} + \text{overlap}_{i,j+1} + \text{overlap}_{i+1,j} + \text{overlap}_{i+1,j+1} < 2.6\text{\AA}$. The pre-computed overlap gives an efficient filter before the self-intersection check, but one cannot tell a nearby passing from a self-intersection by the overlap values.

Locally avoidable self-intersections

Some morph self-intersections make a significant change to a proteins fold, while others do not. In the mathematical notion of topology however all protein chains are topologically equivalent as they all may be refolded e.g. into one long beta strand. Algorithmically such a morph may be constructed by a blow-up technique. A distinction between significant and insignificant fold changes can thus only be decided by human judgement and may differ depending on the application at hand. Here, we search for insignificant self-intersections that we can resolve by local changes of the morph and quantify how much longer the resulting morph is.

Imagine the left- and right-handed loops in Fig. 2 aligned on top of each other. The linear interpolation between the two structures will have a self-intersection. Loops are the most flexible parts of proteins and such a

self-intersection is a minor structural change as we now explain. Mathematically this self-intersection can be removed from the morph as this structural change may be performed by two Reidemeister moves of type one, denoted Ω_1 . We have used that the projection of the loop in Fig. 2 is isolated from the rest of the projected curve. In a projection of a globular protein structure, it is likely that the projection of other parts of the backbone pass through the projected loop. Hereby a longer series of Reidemeister moves may be needed to connect the projections of the two loops. To avoid this algorithmic complexity we choose to make a 3-dimensional analogue of Reidemeister moves. We want to know if we can remove a self-intersection by just changing the morph of the arc forming the loop while leaving the rest of the curve and its morph unchanged. We therefore say that a self-intersection $(a_k, b_k, sign_k, t_k^*)$ is locally Ω_1 -removable if the topological disk given by all the triangles connecting two consecutive points of the loop

$$P_{a_k}(t_k^*), P_{ciel(a_k)}(t_k^*), \dots, P_{floor(b_k)}(t_k^*), P_{b_k}(t_k^*) = P_{a_k}(t_k^*)$$

to the center of mass of all these points is disjoint from all the other line segments of the t_k^* -curve $P_1(t_k^*), \dots, P_{floor(a_k)}(t_k^*)$ and $P_{ciel(b_k)}(t_k^*), \dots, P_n(t_k^*)$. In this case, one can freely deform the loop in a neighborhood of the topological disk without intersecting the rest of the curve. One possible morph that avoids the self-intersection contracts the loop almost to its center of mass where there is room to throw one arc around the loop. However, for less complicated loops one would simply deform the loop by rotating it around the line through the point of self-intersection and the center of mass of the loop. We therefore penalize a locally Ω_1 -removable self-intersection by a price $\mathcal{P}1$ equal to twice the sum of the distances from the points of the loop to this line. The self-intersection shown in Fig. 1 may be removed by an Ω_1 move involving a large part of the protein. We therefore let the user decide the maximal

accepted backbone length, denoted MaxLength involved in the alternate morph. Only self-intersections with $b_k - a_k \leq \text{MaxLength}$ are tested for being Ω_1 removable. In a planer projection into a knot diagram the curve segment from a_k to b_k need neither be simple nor isolated from the remainder of the projected curve. The constructed Ω_1 -move is operating on a 3-dimensional curve. It is therefore not a real Reidemeister move that acts on 2-dimensional knot diagrams. The Ω_1 -move need not project to a Reidemeister 1 move for two main reasons. Firstly, the 3-dimensional loop need not project to a simple curve. For larger MaxLength the loop may even be knotted. Secondly, in the knot diagram the projection of the loop is not likely to be isolated from the remaining projection. Thus the 3-dimensional Ω_1 -move may involve many Reidemeister moves in a knot diagram and can be seen as a meta move in the context of knot diagrams.

If a closed double stranded DNA undergoes the self-intersection considered here, it will change its writhe and hereby also the linking number between the two strands by plus or minus two [34]. To feel this coupling between writhe and twisting, hold a length of torsional stiff wire in the shape of a left-handed loop such that it is in self-contact almost forming a self-intersection. By rotating both ends of the wire, you can force the wire to morph into a right-handed loop. That is, a morph avoiding the self-intersection requires two full rotations around the backbone. Proteins are in principle free to do full rotations around the backbone. One may choose not to penalize the discontinuous writhing caused by Ω_1 loops but we offer the possibility.

Most self-intersections are not Ω_1 -removable. A significant speed up of the calculation time is obtained by: First, move the origin of the coordinate system to the center of mass of the loop, which then is a vertex in all the triangles of the topological disc. Then to sort all line segments by increasing distance from the origin to the center of the line-segments. Finally, to check all line-segment triangle pairs for intersection and stop either when an intersection is found or when the triangle inequality between the disk and line segments implies that an intersection is impossible.

The local zigzag shape of general alpha carbon curves often make self-intersections occur in pairs during a morph. Such a pair corresponds to interchanging over and under sliding in Reidemeister moves of type two, Ω_2 , shown in the middle of Fig. 2. The two crossings, $(a_j, b_j, \text{sign}_j, t_j^*)$ and $(a_k, b_k, \text{sign}_k, t_k^*)$, in an Ω_2 move have opposite signs and both change sign during transversal self-intersections. Without loss of generality, we may assume that $t_j^* \leq t_k^*$. Consider first the case where $t_j^* = t_k^*$. At time t_k^* the two arcs connecting the two points of

self-intersection form a closed curve. If a topological disc spanned by this closed curve is disjoint from the remaining parts of the curve then the two self-intersections can be removed essentially by performing two Reidemeister moves of type two close to this disc. You may construct this such that one of the moves occurs before the time t_k^* and one after. In the general case where $t_j^* \leq t_k^*$ it is sufficient that a topological disc whose boundary includes the two arcs is isolated for some $t \in [t_j^*, t_k^*]$, see Fig. 3. To close the curve and to have room for a morph avoiding the two self-intersections we have to add the line segments traversed by the self-intersection points in the time interval $t_j^* \leq t \leq t_k^*$. In practice we check if the topological disc connecting the loop for $t_{average}^* = (t_j^* + t_k^*)/2$ to its center of mass is disjoint from the remainder of the curves for $t = t_{average}^*$ and if the additional line segments traversed by the self-intersection points are disjoint

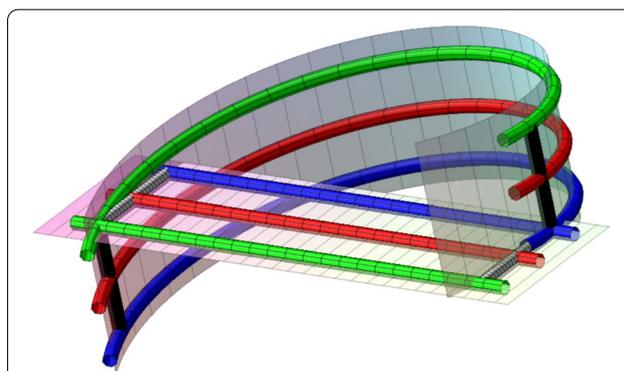


Fig. 3 A space curve homotopy where the helical segment moves upwards and the straight segment moves inclined forward. The first intersection of the two segments is shown in blue and the second in green. Following the motion of the point of intersection on the blue helical segment between the two times of intersection, you get the black cylinder to the right-hand side. Inside this black cylinder the green helical segment can go down and around the intersection point of the two blue curve segments and back up to the green curve. Similarly, the gray cylinder to the right-hand side shows the motion of the first intersection point on the straight segment. Inside the gray cylinder, the green line segment may be altered to go back and around the intersection point. Hence, if the right hand side black and gray cylinders do not intersect other parts of the curve at a time between the two times of intersection, then we can alter all curves inside the two cylinders and postpone the intersection. Similarly, if the black and gray cylinders to the left-hand side do not intersect other parts of the curve between the times of the two intersections, then that intersection can be moved forward in time. Now move both intersections to happen at the mean of the two original intersection times (shown in red). The two red arcs combined with parts of the black and gray cylinders now form a closed loop. If a topological disc spanned by this loop avoids the remaining parts of the red curve, then two Reidemeister moves of type 2 can change the initial under-sliding of the helical segment to an over-sliding and avoid the two intersection points

from the remainder of the curve in the time interval $t_j^* \leq t \leq t_k^*$. See Fig. 3. The first check is speeded up as in the case of the Ω_1 move. If the backbone length of the loop is shorter than the user-defined MaxLength, i.e., if $b_j - a_j + b_k - a_k \leq \text{MaxLength}$, and no obstructions are found we say that $(a_j, b_j, \text{sign}_j, t_j^*)$ and $(a_k, b_k, \text{sign}_k, t_k^*)$ form a locally Ω_2 -removeable loop and penalize it by price $\mathcal{P}2$ twice the sum of the distances from the points of the loop to the line connecting the two points $P_{a_j}(t_{average}^*)$ and $P_{a_k}(t_{average}^*)$.

As in the case of Ω_1 moves, an Ω_2 move acts on space curves. When projected to a knot diagram, an Ω_2 move may require several ordinary Reidemeister moves and may thus be thought of as a meta move of knot diagrams. The sum of the crossings signs is unchanged by an Ω_2 move and there is thus no net discontinuous contribution to the writhe of the curve. E.g., the linking number of double stranded DNA undergoing an Ω_2 move is unchanged and there is therefore no torsional penalty involved.

Essential self-intersections

Let $\{v_i\}$ denote all the self-intersections. Some are Ω_1 -removable at price $\mathcal{P}1(v_i) > 0$. Some pairs $[v_i, v_j]$ are Ω_2 -removable at price $\mathcal{P}2(v_i, v_j) > 0$. We want to remove as many self-intersections as possible at the lowest possible cost when being restricted to removing each v_i at most once. These two problems may be solved simultaneously by one maximal weight matching problem as we now explain. We will denote the self-intersections left after this process a set of essential self-intersections of the morph. Consider the v_i 's as vertices in a graph and let Ω_1 denote the Ω_1 -removable subset. Let the constant ε equal one over twice the sum of the prices of all Ω_1 and Ω_2 moves. Each vertex is associated a weight

$$w_v(v_i) = \begin{cases} \varepsilon \mathcal{P}1(v_i), & \text{if } v_i \in \Omega_1 \\ 1, & \text{else.} \end{cases}$$

We want to remove as many vertices as possible. Hence, each vertex in Ω_1 will be removed by an Ω_1 move unless it's more favorable to let it take part in an Ω_2 move. Hence, we only need to consider how to use the Ω_2 moves. For this, let each Ω_2 move be represented by an edge, e , connecting vertices v_i and v_j and associate this edge with the weight

$$w(e) = w_v(v_i) + w_v(v_j) - \varepsilon \mathcal{P}2(v_i, v_j).$$

Finally, let K be the number of vertices not in Ω_1 , denote the total cost of the Ω_1 moves by $\mathcal{P}1_{all} = \sum_{v_i \in \Omega_1} \mathcal{P}1(v_i)$, and recall that a matching is a subset E' of all the edges that uses each vertex at most once. Let $N(E')$ denote the number of vertices left after performing the Ω_2 moves

specified by E' followed by all the possible Ω_1 moves and let $\mathcal{P}(E')$ be the sum of the prices of all the Ω_1 and Ω_2 moves used. We claim that for each matching E'

$$N(E') + \varepsilon \mathcal{P}(E') = K + \varepsilon \mathcal{P}1_{all} - \sum_{e \in E'} w(e)$$

and denote this quantity by $A(E')$. By this claim a maximal weighted matching E^* maximizes $\sum_{e \in E'} w(e)$ and thus minimizes the left-hand side. By construction $0 \leq \varepsilon \mathcal{P}(E') \leq \frac{1}{2}$ and hereby $N(E^*)$ is minimized and $\mathcal{P}(E^*)$ is minimized subject to the constraint that $N(E^*)$ is minimal, solving both problems at once.

We find it most instructive to prove this claim by induction on the cardinality of E' and start with the case $E' = \emptyset$ where all Ω_1 moves are performed and the claim is trivial. Consider a matching E'' where $e_k \in E''$ and the induction claim is true for $E' = E'' - \{e_k\}$. Then

$$\begin{aligned} A(E'') &= K + \varepsilon \mathcal{P}1_{all} - \sum_{e \in E''} w(e) \\ &= K + \varepsilon \mathcal{P}1_{all} - \sum_{e \in E'} w(e) - w(e_k) \\ &= N(E') + \varepsilon \mathcal{P}(E') - w(e_k), \end{aligned}$$

where the last equality follows by induction. We use the shorthand $\mathcal{P}2(e_k)$ for the price $\mathcal{P}2(v_i, v_j)$ of the edge e_k between the two vertices v_i and v_j .

If $v_i, v_j \notin \Omega_1$ then $w(e_k) = 2 - \varepsilon \mathcal{P}2(e_k)$ and $A(E'') = N(E') - 2 + \varepsilon(\mathcal{P}(E') + \mathcal{P}2(e_k))$. As two additional vertices are removed at the cost $\mathcal{P}2(e_k)$ we find $A(E'') = N(E'') + \varepsilon \mathcal{P}(E'')$.

If $v_i \in \Omega_1$ and $v_j \notin \Omega_1$ then $w(e_k) = \varepsilon \mathcal{P}1(v_i) + 1 - \varepsilon \mathcal{P}2(e_k)$ and $A(E'') = N(E') - 1 + \varepsilon(\mathcal{P}(E') - \mathcal{P}1(v_i) + \mathcal{P}2(e_k))$. Here one additional vertex, v_j , is removed and the change of cost is $\mathcal{P}2(e_k) - \mathcal{P}1(v_i)$ as v_i not is removed by a Ω_1 move but by the Ω_2 move e_k . Thus $A(E'') = N(E'') + \varepsilon \mathcal{P}(E'')$.

Finally, if $v_i, v_j \in \Omega_1$ then $w(e_k) = \varepsilon \mathcal{P}1(v_i) + \varepsilon \mathcal{P}1(v_j) - \varepsilon \mathcal{P}2(e_k)$ and $A(E'') = N(E') + \varepsilon(\mathcal{P}(E') - \mathcal{P}1(v_i) - \mathcal{P}1(v_j) + \mathcal{P}2(e_k)) = N(E'') + \varepsilon \mathcal{P}(E'')$ as no additional vertices are removed and the cost is changed by using the Ω_2 move e_k instead of the two Ω_1 moves. This completes the proof of the claim.

We use the Matlab implementation [35] based on [36]. In practice, this calculation uses only a small fraction of the computation time and may be considered efficient. We also implemented a greedy algorithm that searches for locally contractible Ω_2 moves sorted by increasing backbone length between the two points of self-intersection and removes the ones found. Usually considerable fewer candidate Ω_2 moves have to be checked. Hence, the greedy algorithm is faster but in some cases it cannot remove as many self-intersections as the optimal method based on maximum weighted matching in graphs. There may be several

sets with a minimal number of essential self-intersections, but in most cases the real valued prices of Ω_1 and Ω_2 moves will result in a unique cheapest minimal set. However, for some morphs a slight change in these prices will change the selected minimal set. For more complicated morphs we therefore stress not to emphasise which self-intersections are denoted essential but solely to report their cardinality.

End-contractions

When a self-intersection is not locally removable, we construct a morph using a standard strategy for unknotting ropes or shoelaces namely: while keeping most of the rope fixed, pull the rope until a terminus gets free of the loop restraining it. That is, a self-intersection $(a_j, b_j, sign_j, t_j^*)$ may be avoided by N- or C-terminus contraction. An N-terminus contraction first moves all points upstream of $P_{a_j}(0)$ to this point for interpolation parameter $t = 0$, then uses the corresponding restriction of the original morph from $t = 0$ to $t = 1$, and finally does a reversed end-contraction on the protein structure for $t = 1$. If the end-contraction is restricted to be performed along the backbone path, $P_1(0)$ is moved a total distance of approximately $a_j 3.8 \text{ \AA}$, P_2 is moved approximately $(a_j - 1) 3.8 \text{ \AA}$ etc. The total morph length is therefore quadratic in the distance to the terminus. In general, the following algorithm illustrated on an N-terminus contraction finds a much shorter end-contraction.

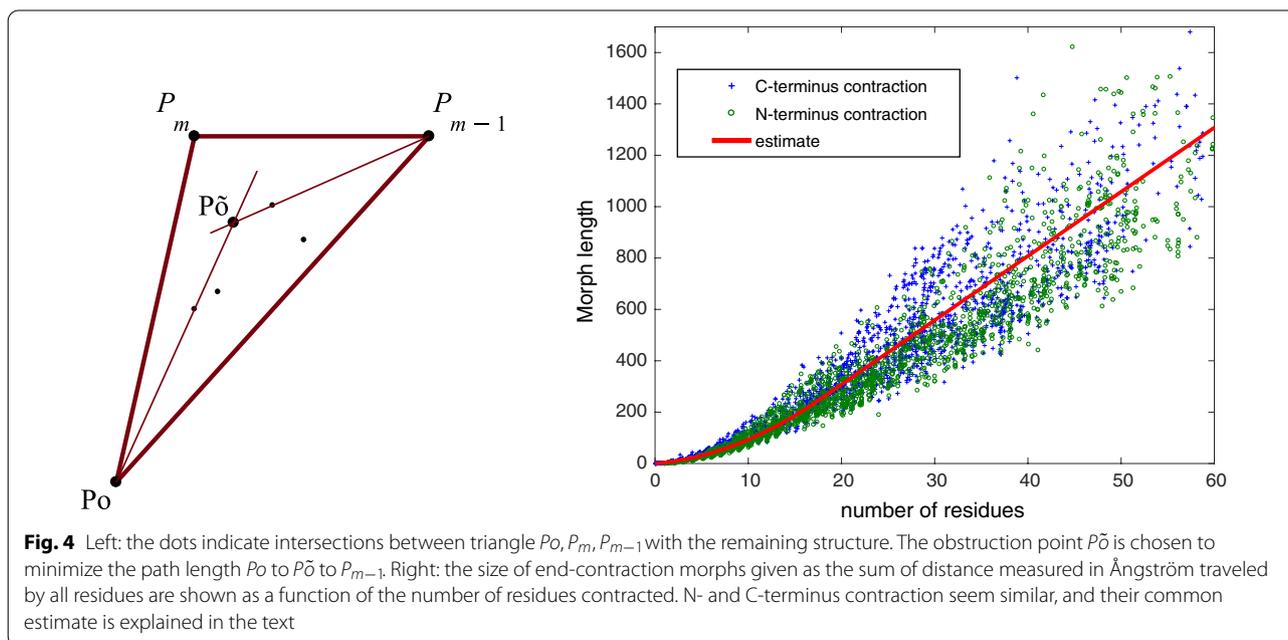
The end-contractions is performed on the original protein structures, and we omit writing the constant interpolation parameter $t = 0$ or 1 . That is P_{a_j} is shorthand for either $P_{a_j}(0)$ or $P_{a_j}(1)$. Let the first obstruction point be $P_o = P_{a_j}$ and let $m = \text{floor}(a_j)$. We have to move the

entire curve P_1, \dots, P_m to the point P_{a_j} . P_m may always be moved along the straight line to P_{a_j} . If the triangle with corners P_o, P_m, P_{m-1} is disjoint from the curve downstream of $P(a_j)$ the shortcut from P_{m-1} to P_o may be used. Otherwise a new obstruction point $P\tilde{o}$ inside the triangle is chosen such that the length of the path P_o to $P\tilde{o}$ to P_{m-1} is minimized and all downstream line segments are avoided. See Fig. 4. The algorithm is greedy and accepts the path from $P\tilde{o}$ to P_{a_j} as the shortest found and all upstream points will use this as part of their path to P_{a_j} . The last encountered obstruction point is moved to $P\tilde{o}$. Then this shortening of the end-contraction is continued for all the upstream line segments. The cost of avoiding the self-intersection $(a_j, b_j, sign_j, t_j^*)$ is thus the total distance traveled by the points in the end-contraction both for $t = 0$ and for $t = 1$ together with m times the distance traveled by $P(a_j)$ in the original linear interpolation, because all the m contracted points travel this distance.

If $a_1 \leq a_2 \leq \dots \leq a_n$ and b_1, b_2, \dots, b_n are the parameter values of the essential self-intersections the combined smallest N and C terminus end-contractions removing them all solve:

$$\begin{aligned} \min_{1 \leq a < b \leq L} & \quad \text{N - Contraction}(a) + \text{C - Contraction}(b) \\ \text{s.t. } & \quad a_i \leq a \text{ or } b \leq b_i \text{ for all } i = 1, \dots, n. \end{aligned}$$

If no N-contraction is used $a = 1$ and $b = \min(b_1, b_2, \dots, b_n, L)$ in order to remove all the self-intersections. If $a = a_1$ the first self-intersection is removed from the N-terminus end and $b = \min(b_2, \dots, b_n, L)$, removes the



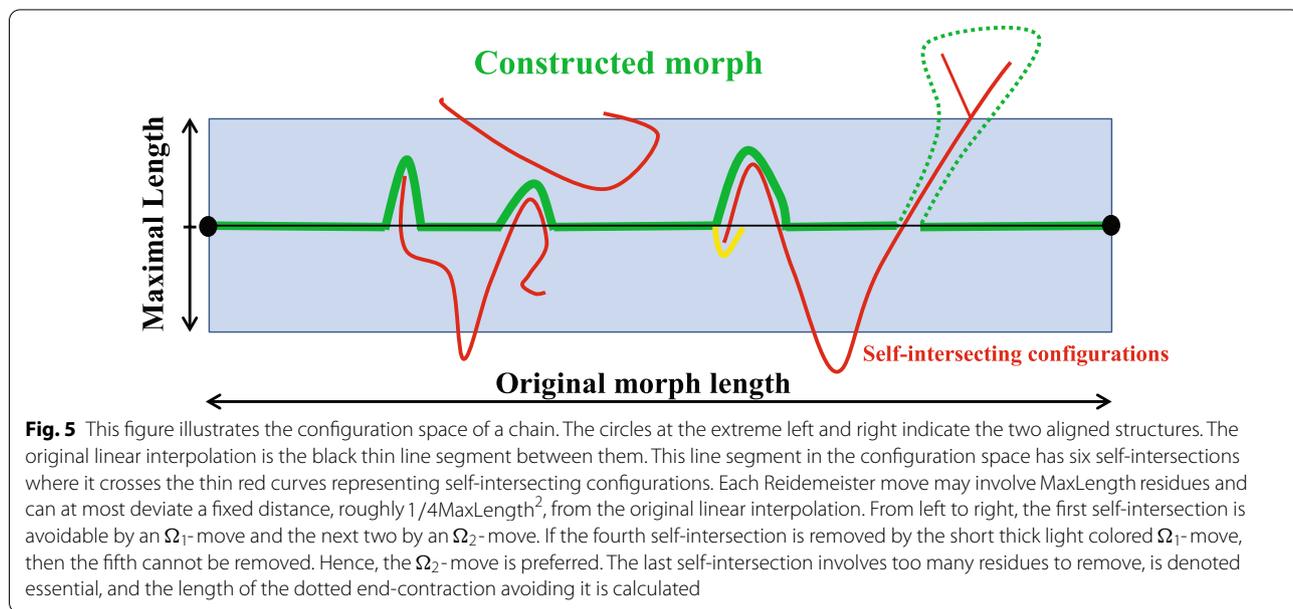
remaining self-intersections. Thus in general $a = a_j$ and $b = \min(b_{j+1}, \dots, b_n, L)$ fulfill the constraints and we only have to check $n + 1$ cases to find this minimum. To speed this calculation up, we use an estimate of the end-contraction based on the observation that the distance traveled by residues up to 17 residues away from the self-intersection point grow roughly linearly to 25\AA and is fluctuating around 25\AA when further apart along the backbone. This gives a reasonable estimate of the sizes of the chosen end-contractions as shown on Fig. 4. The end-contractions are the computationally most costly part of all the calculations. As they only serve to quantify the distance between essentially different folds, we offer the possibility to omit this calculation and instead use the estimated size of the end-contractions. Finally, we offer the possibility to consider a self-intersection as Ω_1 -removable if it is sufficiently close to a protein terminus. One natural choice is to set the longest allowed distance from the point of self-intersection to an end to half of the user-defined MaxLength. In case a self-intersection also

is Ω_1 -removable the price is set to the minimum of the Ω_1 price and the end-contraction price. Figure 5 illustrates the full self-avoiding path that lies in a tubular neighborhood of the original linear interpolation if essential self-intersections are absent.

Treating alignments with gaps

Optimal structural superposition of two protein chains includes gaps in the alignments of the chains to compensate for the naturally occurring insertions and deletions in homologous protein chains. A canonical example is that a loop may be longer in one structure than in the other; but such that the long loop can be contracted into the shorter loop without causing self-intersections of the entire structure. A sequence or structural alignment determines a pairing of only a part of the residues of the two proteins. We choose to fill out the alignment gaps using linear interpolation as explained by the following example involving a 12 residue long Chain 0 and a 10 residue long Chain 1 using the notation of the program TM-align:

Backbone 0	1	2	3	4	5	6	7	8	9	-	-	-	10	11	12
TM align			.	:		:	:						:	:	
Backbone 1			1	2	-	3	4	-	-	5	6	7	8	9	10



The first aligned residue pairs between Chain 0 and Chain 1 are (3, 1), (4, 2) and (6, 3). We treat weakly aligned pairs, indicated with “” by TM-align, as aligned pairs indicated by “:”. Residue 5 of Chain 0 is not aligned and we choose to align it with the midpoint $P_{2.5}^1$ between Chain 1’s second and third residue. The next two aligned pairs (7, 4) and (10, 8) leave 3 intervals open on Chain 0 and 4 intervals on Chain 1. By linear interpolation, Curve 0 needs to be traversed at 3/4 of the speed Curve 1 is traversed resulting in the re-parameterizations:

New param.	1	2	3	4	5	6	7	8	9	10
Re-param. 0	3	4	5	6	7	7.75	8.5	9.25	10	11
IsAligned	1	1	0	1	1	0	0	0	1	1
Re-param. 1	1	2	2.5	3	4	5	6	7	8	9

The re-parameterized Curve 1 is the piecewise linear curve connecting the points:

$$\tilde{P}_1^1 = P_1^1 \quad \tilde{P}_2^1 = P_2^1 \quad \tilde{P}_3^1 = P_{2.5}^1 \quad \tilde{P}_4^1 = P_3^1 \quad \dots \quad \tilde{P}_{10}^1 = P_9^1$$

The re-parameterized Curve 0 connects the points:

$$\begin{matrix} \tilde{P}_1^0 = P_3^0 & \tilde{P}_2^0 = P_4^0 & \tilde{P}_3^0 = P_5^0 & \tilde{P}_4^0 = P_6^0 & \tilde{P}_5^0 = P_7^0 \\ \tilde{P}_6^0 = P_{7.75}^0 & \tilde{P}_7^0 = P_{8.5}^0 & \tilde{P}_8^0 = P_{9.25}^0 & \tilde{P}_9^0 = P_{10}^0 & \tilde{P}_{10}^0 = P_{11}^0 \end{matrix}$$

The piecewise linear curve connecting the \tilde{P}_i^0 's differs slightly from the original curve as e.g. a corner around P_8^0 is cut when connecting $\tilde{P}_6^0 = P_{7.75}^0$ to $\tilde{P}_7^0 = P_{8.5}^0$. However, it is trivial to morph the one curve into the other and the thickness of the protein backbone guarantees that no self-intersections can occur when doing this. The structural alignment thus gives rise to studying the self-intersections of the linear interpolation

$$P(t, a) = (1 - t)P_{\text{RePar}^0(a)}^0 + tP_{\text{RePar}^1(a)}^1,$$

for $t \in [0, 1]$ and $a \in [1; m]$, where m is determined by the alignment. On each line segment the function IsAligned(a) is given by linear interpolation of the values zero (for not aligned) and one (for aligned) at the endpoints of the line segments. In particular, it is one on line segments connecting aligned residues and zero when connecting non-aligned residues.

A self-intersection for parameter values (a_k, b_k) gives the sum $\text{AlignedSum}(k) = \text{IsAligned}(a_k) + \text{IsAligned}(b_k)$. We call the self-intersection aligned-aligned if $\text{AlignedSum}(k) \geq 1.5$, aligned-gap if $1.5 > \text{AlignedSum}(k) > 0.5$ and gap-gap if $\text{AlignedSum}(k) \leq 0.5$. The minimal allowed distance, d_{min} , between points along the backbone curve used to define the overlap are given by linear interpolation of the discrete d_{min} values after the

re-parametrization. E.g., d_{min} for two points 4.80 line segments apart along the backbone curve is set to 3.51Å found by interpolating $d_{\text{min}} = 3.47\text{Å}$ and 3.52Å for points being 4 and 5 line segments apart respectively.

A loop-contraction combined with other deformation between two structures likely lead to a pair of self-intersections of type Ω_2 . If just one of these self-intersections involves the aligned parts and the other not, then the aligned-aligned self-intersection is probably essential when only considering aligned parts of the morph whereas it is removable if all self-intersections are considered. Beside the goal to tell if alignment gaps are topologically similar, this is a reason not to restrict the morph to the aligned parts.

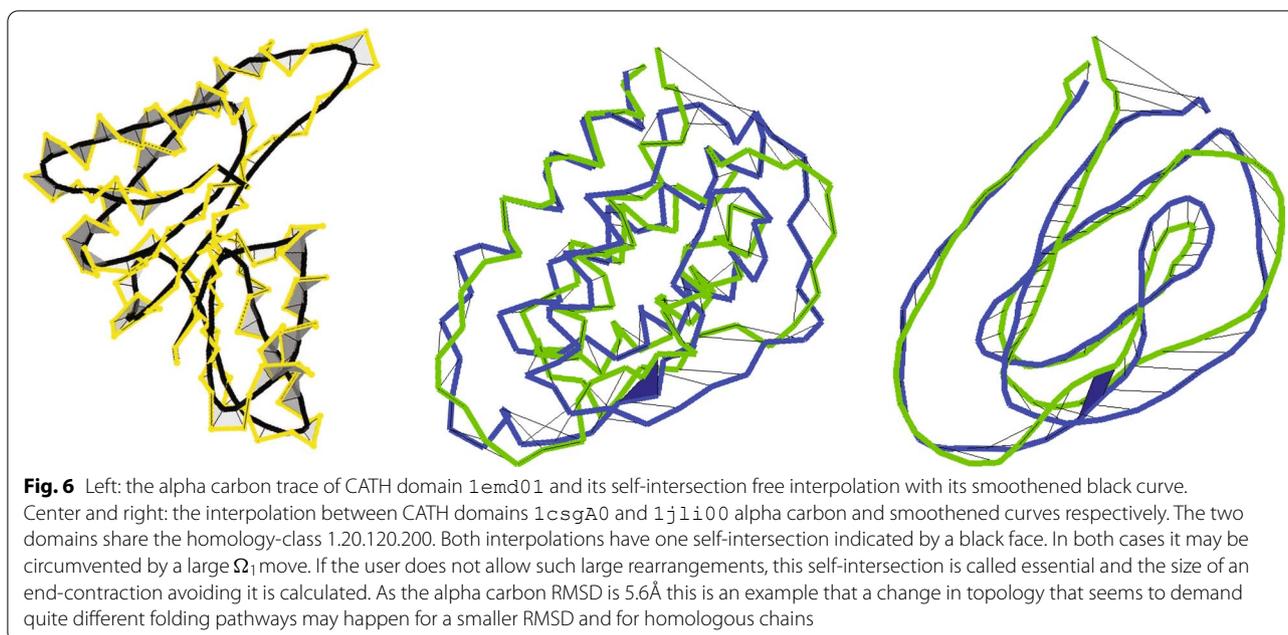
Applying curve smoothening

We represent each protein structure by either the positions of its alpha carbon atoms C_i^α or by a smoothened curve. In the smoothening all alpha carbons, except for the first two and the last two are replaced by the fixed convex combination $P_i = \frac{C_{i-2}^\alpha + aC_{i-1}^\alpha + bC_i^\alpha + aC_{i+1}^\alpha + C_{i+2}^\alpha}{2+2a+b}$. The

constants $a = 2.4$ and $b = 2.1$ are chosen to minimize the total curvature of a collection of protein structures [18]. All linear interpolations from an alpha carbon curve to its smoothened curve are self-intersection free for the protein structures used here. The smoothening straightens both alpha helices and beta-strands, see Fig. 6. The smoothened representation resembles typical cartoon pictures of protein structures and has the advantage often to make it possible to follow an interpolation visually, see Fig. 6. Furthermore, it results in fewer interpolation self-intersections. For the smoothened representation, the overlap is based on the minimal distances 1.0, 2.1, 3.0, 3.4, 3.6Å for residues with indices $|i - j| = 1, \dots, 5$ apart and 3.7Å otherwise. A self-intersection check may be avoided if the line-segments are shorter than 3.5Å and the sum of the endpoints overlap is $< 2.1\text{Å}$.

Results

When performing linear interpolation between more and more distant structures, overlap will start to grow and eventually self-intersections that can be removed using Ω_1 and Ω_2 moves will emerge. At first, each Ω move involves only a small part of the backbone. For a self-intersection with data $(a_k; b_k; \text{sign}_k; t_k)$ we defined the backbone length of the corresponding Ω_1 move as $b_k - a_k$ and for an Ω_2 move it's backbone length is also the (real) number of line segments it involves. When interpolating structures that are more distant these backbone lengths may grow and eventually obstructions to the Ω_1 and Ω_2 moves may occur. This we now illustrate with a few examples with high sequence identity. The *-marked examples have 100%

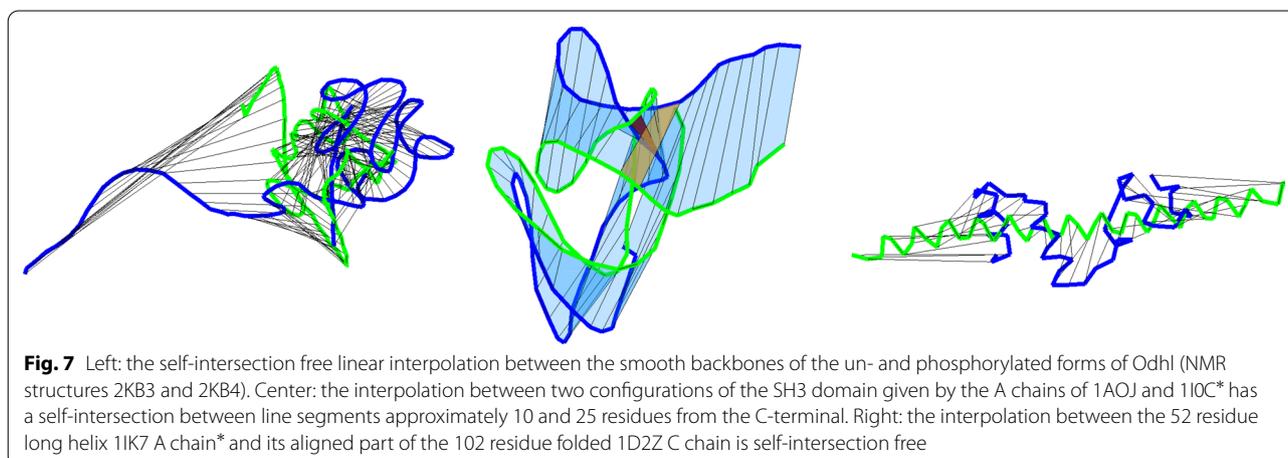


sequence identity and are taken from the Sequence-Similar, Structure-Dissimilar Protein Pairs in the PDB found by Kosloff and Kolodny [2].

Examples with high sequence identity

The A and B chains of the open and closed forms of adenylate kinase, 4AKE and 1ANK, both have $\text{RMSD} \approx 7\text{\AA}$. The interpolation cause no overlap of the alpha carbon curves, and next to no mean overlap of 10^{-3}\AA for the smooth curves and consequently self-intersection checks are not needed. We may conclude that both the A and B chain interpolations are easy to perform. We expect the same conclusion for a hinge-motion of relatively ridged sub-domains since neighboring residues perform almost identical motions. The 5.3Å RMSD

motion between the glutamine binding protein's structures 1GGG and 1WDN with and without ligand is an example of this. The $\text{RMSD} = 8\text{\AA}$ interpolation of hepatitis C virus between the A chain configurations 1A1Q and 1JXP* has a mainly local overlap of mean 0.11\AA and only 5 self-intersection checks are needed for the alpha carbon curve interpolation. The alpha carbon curve interpolation is self-avoiding. The large, $\text{RMSD} = 23\text{\AA}$ motion between the (un-) and phosphorylated forms of Odhl* is shown in Fig. 7 left. The free arm performs a long motion and the remaining compact domain rotates close to a third of a full rotation in the RMSD superposition. The linear interpolation of this large rotation contracts the structure and causes 1.0\AA mean overlap (0.6\AA for the smooth curve). Only



the interpolation of the alpha carbon curves has self-intersections. Its 3 self-intersections may be removed by one Ω_1 and one Ω_2 , move each of backbone length less than 4 line segments. Note, that all interpolations considered so far are 'easy to perform' even if some of them involve larger motions.

The RMSD = 14Å linear interpolation between two configurations of the 60 residue SH3 domain is shown in the middle of Fig. 7. The alpha carbon and smooth backbone interpolations both have one self-intersection, that in both cases may be resolved by an Ω_1 move involving about 15 residues or by an end-contraction of backbone length counted as twice the 10 residues it involves. Hence, this self-intersection is considered as non-essential if MaxLength e.g. is 16 which may be larger than some users will allow. Finally the RMSD = 13Å interpolation from the folded 1D2Z C chain to the 52 residue long helix 1IK7 A chain* is shown to the right in Fig. 7. The folded structure is monotone in the direction of the helix and consequently no self-intersections occur even though most consider the topology changed.

TM-align seldom aligns the entire shorter chain and introduces generally more gaps than the global RMSD alignment. The resulting interpolations therefore are different from the RMSD based above. TM-align aligns 181/183 of the 214 residues of the A/B chains of 4AKE and 1ANK and the re-parameterized curve has 220/220 vertices covering the entire original chains. For the alpha carbon curves the gaps cause mainly local overlap of mean 0.79/0.74Å and one Ω_1 removable self-intersection. The same is found in the A chain interpolation between 1A1Q and 1JXP* and between 1GGG'a A chain and 1WDN, where apart from the local overlap of the alpha carbon interpolation, both the alpha carbon and smooth curve interpolations have non-local overlap. The situation is quite different in the case of Odhl where TM-align aligns only the compact and almost unchanged part of the structure. For the SH3 domain, 35 of the first 36 residues are aligned by TM-align together with the weakly aligned 47th residue. The one self-intersection is again present; this time in the alignment gap and its presence is thus due to the weak alignment of residue 47. In the 1IK7-1D2Z C chain example TM-align aligns only 18 almost consecutive residues identifying two helical segments. These examples fall in two main cases. In the first, the TM-alignment covers most of the aligned chains. Here the gaps in the TM-alignment, that are not taken into account in the TM-superposition, cause more overlap especially for the alpha carbon curve interpolations of which some have a local Ω_1 removable self-intersection. The other main case is when TM-align identifies a structurally conserved region whose interpolation is trivial but where the global RMSD alignment shows that the full structures are connected by relatively long but obstruction free interpolations.

Detecting knottedness

The D chain of 2FG6 contains a right handed trefoil knot located in the interval from the 171th to the 238th residue [26]. We align the 83 residues 162 to 244 of 2FG6's D chain that contain the knot to 408 sequence class representatives of CATH 2.4 domains of compatible lengths from 75 to 100 residues. Using global RMSD superposition, all these interpolations have at least 1 self-intersection with an average of 8.4 intersections (3.6 for the smooth representation). In the following $n(m)$ is short hand for n for the alpha carbon curve and m for the smooth representation. Depending on MaxLength almost all morphs have at least one essential self-intersection. E.g. for MaxLength = 10 only 3(1) domains of topologies 3.40.30 and 3.66.20 are morphed into the trefoil knot without causing essential self-intersections. For MaxLength = 20 there are 11(11) such cases. We have cut 2FG6's D chain relatively close to its trefoil knot, and it may become unknotted by a similar size end-contraction. This effect is very pronounced as self-avoiding morphs are found in 6(5), 34(25), 109(81), 177(151), 325(318) of the 408 cases for MaxLength = 5, 10, 14, 16, 20 respectively when end-contractions up to backbone-length MaxLength/2 are allowed. Without end-contractions, a self-avoiding morph is found only in cases that firstly, are unknotted by the original interpolation by moving an end through an enclosing loop, and that secondly also have a sufficiently large MaxLength to avoid other self-intersections locally. For MaxLength = 20 and with end-contraction of at most 10 residues, the end-contraction may unknot the knot and hereafter the 20 residue long Reidemeister moves are powerful enough to find a self-avoiding morph to most of the other folds. Hence, to capture the characteristics or topology of a fold; one has to set the maximal allowed end-contraction in concordance with how tightly the representative domain is cut. Similarly, many TM-alignment windows are too short to contain the knot, in which case fewer morph obstructions are found.

Data

We pick a representative set of protein structures by taking the sequence family representative CATH2.4 domains [37] restricting to cases where it is gap free and has between 75 and 150 residues. These are clustered at 60% sequence homology and give 1034 domains representing 1034 sequence families, 521 homology families (H-classes), 281 topologies T-classes, 23 architectures and the 4 main classes. If the longest of a pair of domains is at most 10% longer than the shorter domain, we find the alignment of the entire shorter domain with minimal RMSD allowing one inner gap in this alignment. The choice of limited differences in domain lengths together

with shorter but frequent domains is done to make global alignment in the best possible agreement with the structural classification. The global RMSD alignment use exhaustive search and is time consuming. Hence, the choice of domain pairs is also necessary to make the global alignment practical. For each of these 142068 alignments we calculate the RMSD together with overlap and topological obstructions found in the corresponding linear interpolations. We do the same based on the TM-alignments using its standard parameters. We find 135698 inter T-class pairs, 6370 intra T-class pairs, and 1304 intra H-class pairs giving 5066 intra T-class pairs that are not inter H-class pairs. All data from these alignments are provided as Additional files 1, 2, 3, 4 that generate the figures on CATH alignments below.

On inter and intra fold class morphs

This section concerns the 142,068 structural alignments of CATH domain pairs of similar chain length. For the alpha carbon curves 29% of the total overlap is local caused by residues with indices $|i - j| \leq 4$. Relatively large local metric changes when alpha helices are involved is the main contributor to the local overlap. For comparison, only 7% of the total overlap is local for the smooth curves. The non-local contributions coming from aligned regions are highly correlated (0.97) between the alpha carbon and smoothed curves.

Self-intersections and alignment gaps

A structure with $n + 1$ alpha carbons has n line segments and $n(n - 1)/2$ line segment pairs. If a self-intersection has $\text{AlignedSum} \geq 1.5$ then we say it is between two aligned parts of the structure. Otherwise, a least one alignment gap is involved. Our global RMSD alignment allows one gap that usually is too short to make intersections with itself in the linear interpolation. One thousand aligned line segment pairs on average cause 2.2 self-intersections and gap-involving line-segment pairs cause 1.6. The smooth representation generally cause fewer self-intersections and the corresponding numbers are 0.81 and 0.52 respectively. When using TM-align we connect all inner gaps in the alignments to check if they cause topological changes. We use the Aligned Window Fraction (AWF) given by the shorter aligned window length divided by the shorter chain length to characterize the coverage of the aligned windows. On average 55% of the residues in the shorter chain are aligned and AWF is 0.8 on average. From allowing similar size end-contractions on the knotted example above, one hereby expects and finds significantly fewer self-intersections. One thousand aligned line segment pairs on average cause 0.17 (0.048 smooth) self-intersection where gap-involving

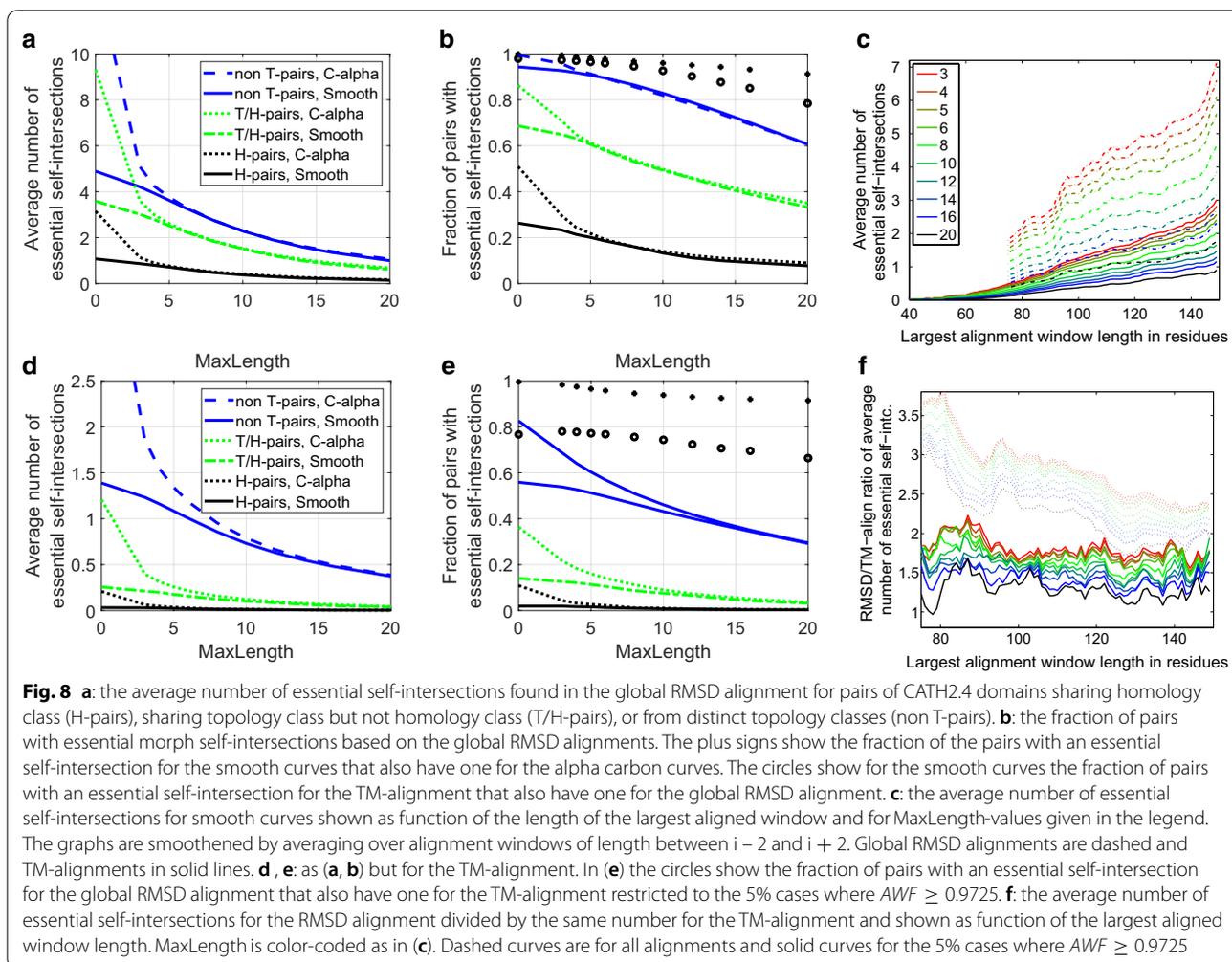
line-segment pairs cause 0.73 (0.22 smooth). TM-align aligns pairs of alpha carbons that in the final superposition are at distance at most 5\AA . This short distance does not prevent self-intersections when interpolating the superimposed structures. In 142068 TM-aligned domain pairs, we find 42165 (12103 smooth) self-intersections of aligned parts. For both alignment methods, the alpha carbon curves cause almost 3 times as many self-intersections as the smooth curves.

Essential self-intersections

Figure 8a shows the average number of essential self-intersections as function of the maximal allowed backbone length, MaxLength , of Ω_1 and Ω_2 moves. The original number of self-intersection is found for $\text{MaxLength} = 0$. There are clearly more self-intersections for the alpha carbon representation, e.g. on average 13 between fold classes compared to the 4.9 for the smooth representation, and there is a relatively low correlation of 0.68 between the numbers of self-intersections in the two representations. However, the essential self-intersection numbers of the two representations are closely related for MaxLength at least four. See Fig. 8a, b. Their correlation is 0.92 for $4 \leq \text{MaxLength} \leq 6$ and drops slowly down to 0.79 for $\text{MaxLength} = 20$.

The shorter TM-alignments give fewer essential self-intersections than the global RMSD alignments as expected. For the TM-alignments MaxLength needs to be larger to remove the additional self-intersections of the alpha carbon curves. See Fig. 8d, e. For both alignment methods, an essential self-intersection for smooth curves generally implies one for the alpha carbon curves and the notion of essential self-intersections therefore seems robust when restricted to one alignment method. Both the alignments and superpositions provided by the two alignment methods may be very different. One should thus not expect essential self-intersections to agree between them. However, if there is an essential self-intersection in the TM-aligned window, then it is more likely than average to find one in the RMSD alignment, Fig. 8b. If the TM-aligned window covers most of the structure also the opposite tendency is found, Fig. 8e.

Figure 8c shows the average number of essential self-intersections as function of the aligned window length and Fig. 8f shows the ratio between them. TM-align can avoid self-intersections by choosing a shorter aligned window. But also in cases where the TM-alignment window is almost global, it results in fewer essential self-intersection. See Fig. 8f. Possibly both TM-aligns freedom to optimize the alignment using multiple gaps and its choice of superposition result in fewer essential self-intersections. Figure 8f shows that for larger values of MaxLength the local Reidemeister

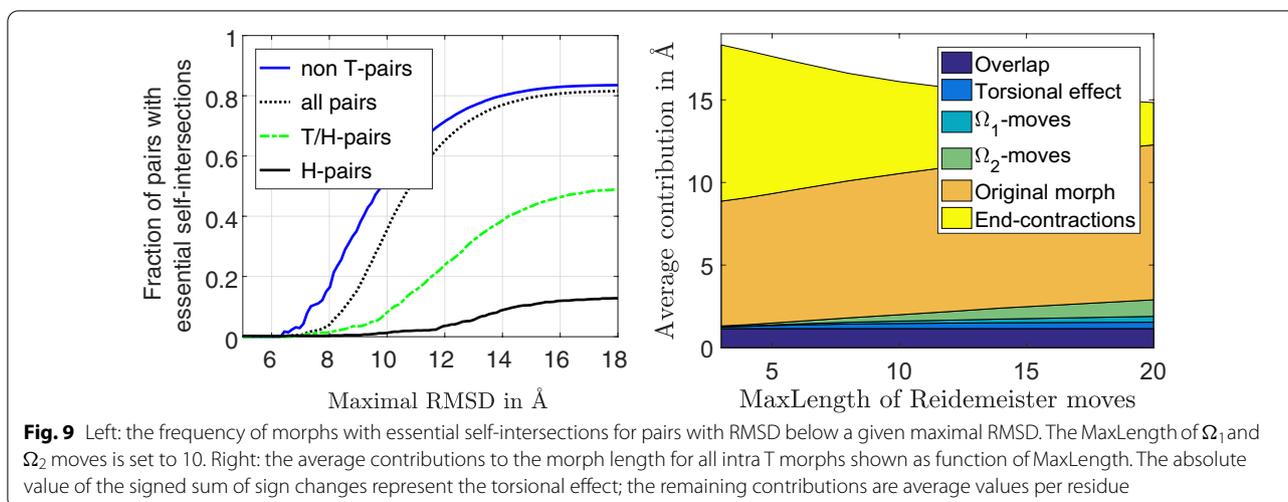


moves manage to compensate for many of these additional self-intersections caused by the lack of gaps in the global alignments.

Morphability and fold classification

Interpolations between pairs of homological structures have essential self-intersections relatively often. Figure 6 shows an example where the sole self-intersection may be removed by an Ω_1 move involving half of the chain or by a relatively large movement of one of the long ends. This author finds it debatable if these two domains should share fold class. For MaxLength = 10 as many as 13% of all H-pair morphs have essential self-intersections, but usually for greater RMSD than this example as seen from Fig. 9. The fraction of H-pair morphs with essential self-intersections varies between the classes. E.g., 5 out of the 14 alignments in class 1.20.120.200 self-intersects as seen on Fig. 6. In total 47 of the 512 H-classes with intra H-class morphs have

essential self-intersections for MaxLength = 10, but most classes require more data to be studied in detail. See the included supplementary data. Due to the large global RMSD values of interpolations with essential self-intersections, TM-align typically aligns too few residues to capture the eventual topological dissimilarity. The most striking finding is not that more than half of all intra T-class global RMSD morphs have essential self-intersections, but that many inter T-pair morphs do not. For smooth representation and MaxLength = 10, 868 of 879 H-pairs, 1189 of 1295 non-H but T-pairs and 1831 of 3852 inter T-pairs with $RMSD \leq 10$ do not cause essential self-intersections. Hence, under this RMSD restriction, most intra T-pairs, in total 2057 of 2174, are morphable, but there is a compatible number of self-avoiding morphs of similar length into other fold classes emphasizing the continuous nature of protein folds. An example is the 7.1Å global RMSD interpolation between the 6 helix domain 2ygsA0 and the 2 helix



and 4 beta-strand domain 1ha101 which is self-intersection free both in the alpha carbon and smooth representation. The 0.49\AA (0.03\AA smooth curve) mean overlap has very little 0.03\AA (0.001\AA) non-local contribution. Hence, even if their local geometry and hydrogen bonding patterns are very different, their full backbones are of typical H-pair distance in our data set, and the linear interpolation causes only minor local steric problems.

Morph lengths

In case no essential self-intersection is found, the original L^1 morph length plus the price of the Ω_1 and Ω_2 moves is an estimate of the length of a self-avoiding morph for the piecewise linear backbone curve not taking into account the thickness of a protein chain. The shortest overlap free distance is $d = 3.7\text{\AA}$. Hence, each point on the backbone curve has to traverse half the circumference of a circle with radius $d/2$ to go from over sliding to under sliding and staying at an overlap free distance. One may therefore add the penalty $\pi * d/2 * |b_k - a_k + 1|$ for Ω_1 moves and $\pi * d/2 * |b_j - a_j + b_k - a_k + 2|$ for Ω_2 moves. The total overlap is the sum of the distances each pair of amino acids need to get further apart during the morph in order to get an overlap free morph. One may therefore wish to add and possibly weight the AverageOverlap to the morph-length estimate. Similarly, one may introduce a weight for the end-contraction length and for the penalty for net torsional effects. Figure 9 shows the average contribution from one residue to morph lengths on intra T-morphs as function of MaxLength for the smoothed backbone curves. Clearly shorter morphs are found for larger MaxLength as end-contractions

are longer than local Reidemeister moves. Inter T-class morphs are on average 26\AA for MaxLength = 3 and 21\AA for MaxLength = 20, that is only slightly longer than the intra T-class morphs. The global RMSD alignment thus do not find a drastic difference in the lengths of inter and intra fold class self-avoiding morphs. TM aligns approximately half of the residues, namely those superimposed within 5\AA distance. The morphs lengths of the aligned windows are therefore considerably shorter than those obtained using the global RMSD alignment. The TM-alignment based morph lengths increase mainly due to the gaps in the alignment and to the topological obstructions of the morph. For MaxLength = 3 the average morph lengths are 3.2 , 6.4 and 13.2\AA per residue for intra H, inter H but intra T, and inter T class morphs respectively. For MaxLength = 20 the similar numbers drop to 3.0 , 5.5 and 11.0\AA .

Discussion

We are handling a paradox by trying to establish a “topological” distinction in the path connected space of protein structures, and we emphasize that the set of essential self-intersections we find are only local obstructions in the non-convex but connected space of protein embeddings. If a current alignment score function shows high similarity then the two configurations are close, at least, in the space of immersions, which allows self-intersections. We offer to test if the linear interpolation between the two structures is self-avoiding and if not then we offer an algorithmic search for alternative morphs through embeddings in a neighborhood of the linear interpolation. We divide the self-intersections into two sets. The

set of non-essential self-intersections contains the maximal number of self-intersections that can be avoided using a sequence of Ω_1 and Ω_2 moves. The remaining self-intersections are called essential. There are three possible reasons why a given self-intersection is marked essential. The first is that a needed Ω move involves larger parts of the protein chain than permitted by the user. The second is that there is a “topological” obstruction to a needed Ω move. The third is that the self-intersection can be avoided but it is cheaper to avoid other self-intersections. The “topological” obstruction is a case where the constructed Ω move will cause at least one new self-intersection and is technically found as an intersection between a constructed disk-surface and the protein structure. However considering protein structures as open curves any such intersection can be avoided by deforming the disk-surface. In most cases these intersections need to be pushed past at least one chain terminal, and this results in a large move that is more efficiently resolved by an end-contraction. In some cases two backbone-surface intersections lie close along the backbone and a smaller alteration of the surface may avoid them both - think of untying a slip-knot. In this study, there has most often been only one backbone-surface intersection. Hence, it's a relatively rare situation that the obstructions to an Ω move can be avoided locally and this lies outside the scope of this work, but it may become more important if longer MaxLength are wanted. Algorithmically we have decided to keep our untying moves local along the backbone to ensure they can be treated in one-pass. Mathematically, our novel notion of essential self-intersection is hereby not a topological but a geometric notion saying that any self-avoiding morph between the two configurations has to go further than a given threshold away from the linear morph path. It is our hope and main aim that this tunable threshold reflects a general perception of when two structures have different folding paths.

In the example shown on Fig. 1 *1csgA0* is resolved by x-ray diffraction and *1jli00* by NMR. From the alignment it's clear that the two structures have similar distance matrices and residue contacts leaving the possibility that the morph self-intersection points to a modeling error in one of the two structures. The smoothened representation makes visual inspection of morphs significantly easier. The speedup of calculations due to fewer self-intersections in this representation was however neutralized as the filtering of potential self-intersections by overlap is less efficient. The overlap is the sum of the steric clashes during an imagined linear interpolation between two aligned and superimposed structures. It is important algorithmically and may be interesting structural biologically. Even in

a case where the backbone curve morph is self-avoiding, the overlap holds information in addition to the morph length. We saw examples of long overlap free morphs between sequence identical highly flexible structures. In these morphs, a neighborhood of most residues undergo a motion close to a rigid motion. This cause both little overlap and that the shorter internal distances are subject to only smaller changes. Hence, a structural comparison based on local internal distances as FlexE will also detect that neighboring residues perform similar motions. Thus, the combined information of RMSD and overlap may be similar that of RMSD and FlexE in such cases.

Conclusion and future work

We present a number of measures that quantify obstacles to deform a protein structure to an aligned and superimposed structure and show they give significant additional information to the distances usually used in protein structure alignment score functions. We study the linear interpolation between two superimposed structures. First, we quantify the steric problems of this linear interpolation by finding the shortest distance between each amino acid pair during the interpolation and compare it to distance constraints found in native protein structures. Larger deformations between sequence similar native structures cause little overlap and are recognized as longer but simple morphs. Next, we find all self-intersections of the protein backbone during linear interpolation between two aligned structures. We introduce 3-dimensional versions of Reidemeister moves each capable of avoiding one or two self-intersections by altering the original linear morph. Near a protein terminal an end-contraction can also be used to avoid a self-intersection. We calculate the length of each of these morph alterations and solve the optimization problem to avoid the maximal number of self-intersections at the lowest additional morph length. If this results in a self-avoiding morph the structural similarity found by the alignment program is verified for the entire aligned windows including all inner alignment gaps. Otherwise we find a smallest set of essential self-intersections that cannot be avoided. We demonstrate that this novel notion of essential self-intersections is relatively robust, e.g., under a change in the representation of protein chains. The user may input the maximal allowed chain length of Reidemeister moves and of end-contractions. We hope this allows users to tailor an appropriate notion of when two structures have different threading for the application at hand.

We find as expected that morphs from unknotted proteins to a given knotted protein have essential self-intersections except for a few cases where the morphs actually untie the

knot. Using a global RMSD alignment and a set of sequence representative CATH domains, we find a significant fraction of all homologous protein structure pairs separated by at least one essential self-intersection and give examples of homologous pairs that most will be considered as different threaded despite having lower RMSD. Our perhaps most interesting finding is that many representatives of distinct folds may be morphed into each other using only smaller alterations of the direct linear interpolation. This supports the continuous view on parts of protein fold space.

The residues that are aligned by TM-align are superimposed within 5 Å. We find 42165 (12103 smooth curve) self-intersections between aligned parts in 142068 TM-alignments. Thus this close distance still allows topological dissimilarity of the aligned parts. In the TM-alignments performed in this work approximately half of the residues are aligned on average. The TM-alignments result in fewer topological obstructions than expected alone from their shorter alignment length. An interesting subject for future research is therefore to develop methods that combine the superior structural alignment search of e.g. TM-align with the restriction to local topological similarity defined by the absence of essential morph self-intersections. The current algorithm is in Matlab and requires two superimposed structures and the alignment in TM-align format as input. It is available on request to the author. When a matching alignment method is developed, the software can stand-alone and will be released.

With fixed sequence alignment, protein structure prediction is a natural application of our present method as structural comparison is done for the entire structure. Capable of detecting differences in threading and pointing out where in the structure these problems occur, we expect our method to give an interesting supplement to the distance measures used to assess protein models as e.g. done in the Critical Assessment of protein Structure Prediction (CASP) experiments.

When combined with TM-align, our method can detect cases where structurally similar proteins have different folding pathways, which is important for understanding protein folding. In addition, alternate threading of a similar core structure may change dynamical and thereby functional properties. Further, we hope our focus on self-avoiding morphing may contribute to the development of alignment methods and to a more detailed picture of sequence to structure relationship for proteins and RNA.

Supplementary information

The online version contains supplementary material available at <https://doi.org/10.1186/s13015-020-00180-3>.

Additional file 1. SupplementaryDataCATH2_4Domains.m. Matlab script that loads and explains all the raw data from all CATH2.4 alignments. It produces figures and may be altered to investigate the data further.

Additional file 2. Data from aligning CATH2.4 domains. NeamTM3to-20NoEndContraction.mat Data from global RMSD and TM-align alignments of alpha carbon backbone curves. The maximal length of Reidemeister moves is varied from 3 to 20 and end-contractions are not allowed. Similar data for the smooth representation of the backbone curve are contained in NeamTMSmooth3to20NoEndContraction.mat and in NeamTMSmooth3to20EndContraction.mat when allowing end-contractions of MaxLength=2/residues.

Additional file 3. Data from aligning CATH2.4 domains. NeamTM3to-20NoEndContraction.mat Data from global RMSD and TM-align alignments of alpha carbon backbone curves. The maximal length of Reidemeister moves is varied from 3 to 20 and end-contractions are not allowed. Similar data for the smooth representation of the backbone curve are contained in NeamTMSmooth3to20NoEndContraction.mat and in NeamTMSmooth3to20EndContraction.mat when allowing end-contractions of MaxLength=2/residues.

Additional file 4. Data from aligning CATH2.4 domains. NeamTM3to-20NoEndContraction.mat Data from global RMSD and TM-align alignments of alpha carbon backbone curves. The maximal length of Reidemeister moves is varied from 3 to 20 and end-contractions are not allowed. Similar data for the smooth representation of the backbone curve are contained in NeamTMSmooth3to20NoEndContraction.mat and in NeamTMSmooth3to20EndContraction.mat when allowing end-contractions of MaxLength = 2/residues.

Acknowledgements

The author is pleased to thank Kresten Lindorff-Larsen for suggesting examples of flexible proteins and Christian Henriksen for discussions on how to remove the maximal number of self-intersections of a morph such that the resulting additional morph length is minimal and appreciates his elegant formulation presented here. In addition, I would like to thank Rachel Kolodny for interesting discussions on protein fold space causing me to start working on measuring interpolation overlap while enjoying her hospitality in Haifa, and finally the anonymous reviewers for their very careful reading and thoughtful suggestions for improvements of the manuscript.

Authors' contributions

The author read and approved the final manuscript.

Funding

Not applicable.

Availability of data and materials

All data on the alignments of CATH2.4 domains are supplied as supporting material together with a Matlab-script for reading and handling the data.

Ethics approval and consent to participate

Not applicable.

Competing interests

The author declares not to have competing interests.

Received: 2 March 2018 Accepted: 17 December 2020
Published online: 27 February 2021

References

- Hasegawa H, Holm L. Advances and pitfalls of protein structural alignment. *Curr Opin Struct Biol*. 2009;19(3):341–8. <https://doi.org/10.1016/j.sbi.2009.04.003>.
- Kosloff M, Kolodny R. Sequence-similar, structure-dissimilar protein pairs in the PDB. *Prot Struct Funct Bioinform*. 2008;71(2):891–902. <https://doi.org/10.1002/prot.21770>.
- Nepomnyachiy S, Ben-Tal N, Kolodny R. Global view of the protein universe. *Proceed Nat Acad Sci USA*. 2014;111(32):11691–6. <https://doi.org/10.1073/pnas.1403395111>.
- Venkatraman V, Yang YD, Sael L, Kihara D. Protein-protein docking using region-based 3d zernike descriptors. *Bmc Bioinform*. 2009;10(1):407. <https://doi.org/10.1186/1471-2105-10-407>.
- La D, Esquivel-Rodríguez J, Venkatraman V, Li B, Sael L, Ueng S, Ahrendt S, Kihara D. 3d-surfer: software for high-throughput protein surface comparison and analysis. *Bioinformatics*. 2009;25(21):2843–4.
- Nepomnyachiy S, Ben-Tal N, Kolodny R. Complex evolutionary footprints revealed in an analysis of reused protein segments of diverse lengths. *Proceed Nat Acad Sci USA*. 2017;114(44):11703–8. <https://doi.org/10.1073/pnas.1707642114>.
- Zemla A, Venclovas C, Moulton J, Fidelis K. Processing and analysis of CASP3 protein structure predictions. *Prot Struct Funct Bioinform*. 1999;37(53):22–9. [https://doi.org/10.1002/\(SICI\)1097-0134\(1999\)37:3<22::AID-PROT5>3.0.CO;2-W](https://doi.org/10.1002/(SICI)1097-0134(1999)37:3<22::AID-PROT5>3.0.CO;2-W).
- Gamliel R, Kedem K, Kolodny R, Keasar C. A library of protein surface patches discriminates between native structures and decoys generated by structure prediction servers. *BMC Struct Biol*. 2011;11(1):20. <https://doi.org/10.1186/1472-6807-11-20>.
- Marks DS, Colwell LJ, Sheridan R, Hopf TA, Pagnani A, Zecchina R, Sander C. Protein 3d structure computed from evolutionary sequence variation. *Plos ONE*. 2011;6(12):28766. <https://doi.org/10.1371/journal.pone.0028766>.
- Kabsch W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallograph Sect*. 1976;A32(5):922–3.
- Zhang Y, Skolnick J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*. 2005;33(7):2302–9. <https://doi.org/10.1093/nar/gki524>.
- Holm L, Sander C. Protein-structure comparison by alignment of distance matrices. *J Mol Biol*. 1993;233(1):123–38.
- Perez A, Yang Z, Bahar I, Dill KA, MacCallum JL. Flexe: Using elastic network models to compare models of protein structure. *J Chem Theory Comput*. 2012;8(10):3985–91. <https://doi.org/10.1021/ct300148f>.
- Mariani V, Biasini M, Barbato A, Schwede T. Iddt: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics*. 2013;29(21):2722–8. <https://doi.org/10.1093/bioinformatics/btt473>.
- Erdmann M. Protein similarity from knot theory: geometric convolution and line weavings. *J Comput Biol*. 2005;12(6):609–37. <https://doi.org/10.1089/cmb.2005.12.609>.
- Levitt M. Protein folding by restrained energy minimization and molecular-dynamics. *J Mol Biol*. 1983;170(3):723–64.
- Røgen P, Fain B. Automatic classification of protein structure by using Gauss integrals. *Proceed Nat Acad Sci USA*. 2003;100(1):119–24.
- Røgen P. Evaluating protein structure descriptors and tuning Gauss integral based descriptors. *J Phy-Condens Matter*. 2005;17(18):1523–38. <https://doi.org/10.1088/0953-8984/17/18/010>.
- Røgen P, Sinclair R. Computing a new family of shape descriptors for protein structures. *J Chem Inform Comput Sci*. 2003;43(6):1740–7. <https://doi.org/10.1021/ci034095a>.
- Penner RC, Knudsen M, Wiuf C, Andersen JE. Fatgraph models of proteins. *Communicat Pure Appl Mathe*. 2010;63(10):1249–97. <https://doi.org/10.1002/cpa.20340>.
- Penner RC, Knudsen M, Wiuf C, Andersen JE. An algebro-topological description of protein domain structure. *PLoS ONE*. 2011;6(5):19670. <https://doi.org/10.1371/journal.pone.0019670>.
- Krishnamoorthy B, Provan S, Tropsha A. A topological characterization of protein structure. *Springer Optimizat Applicat*. 2007;7:431–54. <https://doi.org/10.1007/978-0-387-69319-4-22>.
- Budowski-Tal I, Nov Y, Kolodny R. FragBag, an accurate representation of protein structure, retrieves structural neighbors from the entire PDB quickly and accurately. *Proceed Nat Acad Sci USA*. 2010;107(8):3481–6. <https://doi.org/10.1073/pnas.0914097107>.
- Røgen P, Karlsson PW. Parabolic section and distance excess of space curves applied to protein structure classification. *Geom Dedic*. 2008;134(1):91–107. <https://doi.org/10.1007/s10711-008-9247-z>.
- Lua RC. PyKnot: a PyMOL tool for the discovery and analysis of knots in proteins. *Bioinformatics*. 2012;28(15):2069–71. <https://doi.org/10.1093/bioinformatics/bts299>.
- Lai YL, Chen CC, Hwang JK. pKNOT vol 2: the protein KNOT web server. *Nucleic Acids Res*. 2012;40(1):228–31. <https://doi.org/10.1093/nar/gks592>.
- Takusagawa F, Kamitori S. A real knot in protein. *J Am Chem Soc*. 1996;118(37):8945–6. <https://doi.org/10.1021/ja961147m>.
- Faisca PFN. Knotted proteins: a tangled tale of structural biology. *Comput Struct Biotechnol J*. 2015;13:459–68. <https://doi.org/10.1016/j.csbj.2015.08.003>.
- Niemyska W, Dabrowski-Tumanski P, Kadlof M, Haglund E, Sulkowski P, Sulkowska JI. Complex lasso: new entangled motifs in proteins. *Scie Rep*. 2016;6(1):36895. <https://doi.org/10.1038/srep36895>.
- Dabrowski-Tumanski P, Sulkowska JI. Topological knots and links in proteins. *Proceed Nat Acad Sci USA*. 2017;114(13):3415–20. <https://doi.org/10.1073/pnas.1615862114>.
- Khatib F, Rohl CA, Karplus K. Pokefind: a novel topological filter for use with protein structure prediction. *Bioinformatics*. 2009;25(12):281–8. <https://doi.org/10.1093/bioinformatics/btp198>.
- Franklin J, Koehl P, Doniach S, Delarue M. Minactionpath: maximum likelihood trajectory for large-scale structural transitions in a coarse-grained locally harmonic energy landscape. *Nucleic Acids Res*. 2007;35(2):477–82. <https://doi.org/10.1093/nar/gkm342>.
- Mishra R, Bhushan S. Knot theory in understanding proteins. *J Mathe Biol*. 2012;65(6–7):1187–213. <https://doi.org/10.1007/s00285-011-0488-3>.
- Randrup T, Røgen P. How to Twist a Knot. *Arch Math*. 1997;68(3):252–64. <https://doi.org/10.1007/s000130050055>.
- Saunders D. Matlab implementation (2013) <http://www.mathworks.com/matlabcentral/fileexchange/42827-weighted-maximum-matching-in-general-graphs>
- Galil Z. Efficient algorithms for finding maximum matching in graphs. *Comput Surv*. 1986;18(1):23–38. <https://doi.org/10.1145/6462.6502>.
- Sillitoe I, Lewis TE, Cuff A, Das S, Ashford P, Dawson NL, Furnham N, Laskowski RA, Lee D, Lees JG, Lehtinen S, Studer RA, Thornton J, Orengo CA. CATH: comprehensive structural and functional annotations for genome sequences. *Nucleic Acids Res*. 2014;43(D1):376. <https://doi.org/10.1093/nar/gku947>.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions

