Piecewise Linear Approximation of Protein Structures using the Principle of Minimum Message Length

Arun S. Konagurthu, Lloyd Allison, Peter J. Stuckey, and Arthur M. Lesk

1 Clayton School of Information Technology, Monash University, Clayton, VIC 3800 Australia.
2 Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, VIC 3010 Australia.
3 Department of Biochemistry and Molecular Biology and The Huck Institute for Genomics, Proteomics and Bioinformatics, The Pennsylvania State University, University Park, PA 16802 USA.

ABSTRACT

Simple and concise representations of protein folding patterns provide powerful abstractions for visualizations, comparisons, classifications, searching and aligning structural data. Structures are often abstracted by replacing standard secondary structural features – that is, helices and strands of sheet – by vectors or linear segments. Relying solely on standard secondary structure may result in a significant loss of structural information. Further, traditional methods of simplification crucially depend on the consistency and accuracy of external methods to assign secondary structures to protein coordinate data. Although many methods exist automatically to identify secondary structure, the impreciseness of definitions, along with errors and inconsistencies in experimental structure data, drastically limit their applicability to generate reliable simplified representations, especially for structural comparison.

This paper introduces a mathematically rigorous algorithm to delineate protein structure using the elegant statistical and inductive inference framework of Minimum Message Length (MML). Our method generates consistent and statistically robust piecewise linear explanations of protein coordinate data, resulting in a powerful and concise representation of the structure. The delineation is completely independent of the approaches of using hydrogen-bonding patterns or inspecting local substructural geometry that the current methods use. Indeed, as is common with applications of the MML criterion, this method is free of parameters and thresholds, in striking contrast to the existing programs which are often beset by them.

The analysis of results over a large number of proteins suggests that the method produces consistent delineation of structures that encompasses, among others, the segments corresponding to standard secondary structure. 


1 INTRODUCTION

With the rapid growth in the corpus of known structures, concise representations are increasingly preferred to inspect and analyze protein folding patterns [Richardson, 1981, Taylor et al., 1983, Abagyan and Maio, 1988, Lesk, 1995]. At the core of this simplification is the idea that proteins contain repetitive substructural elements and that the essence of a fold lies in the assembly and interaction of these elements [Lesk and Chothia, 1980, Lesk, 1995, Kamat and Lesk, 2007, Konagurthu and Lesk, 2010].

The appearance of some of these elements arises from the periodicity in the patterns of hydrogen bonds between backbone nitrogen and carbonyl groups along the protein polypeptide chain. Among the standard secondary structure definitions are: helix (α-helix, β-helix, and 310-helix) and strand of sheet [Edsall et al., 1966]. Ideally, the spatial trace of α-Carbon (Cα) atoms of standard secondary structure show a linear trend allowing them to be abstracted using vectors or line segments, without much loss of structural information about the fold. The common practice is to fit an axis to a helix and a least-square line to Cα or main chain atoms of strands of sheet [Chothia et al., 1981, Lesk, 1995].

Replacement of secondary structural elements with line segments is therefore one of the common methods to abstract protein structures and construct concise representation of their folding patterns. The number of standard secondary structural elements observed in a protein is typically an order of magnitude smaller than the number of residues in a chain. Therefore methods that utilize concise representations clearly benefit from a massive space and computational saving, especially when comparing and analyzing structures on a large scale [Abagyan and Maio, 1988, Mizuguchi and Go, 1995, Shi et al., 2007, Konagurthu et al., 2008].

Methods that abstract protein structure at the level of secondary structure generally rely on external programs that can automatically assign secondary structures to coordinate data. However, accurate identification and assignment of secondary structure is an inexact process [Cuff and Barton, 1999]. Although definitions based on hydrogen-bonding provides some rigor in assigning secondary structure, the standard definition of what constitutes a hydrogen bond is based on the notion of bond energy whose measurement can be imprecise and acutely sensitive even to small differences in the position of nitrogen and carbonyl atoms, especially the carbonyl oxygen positions. Two popular programs that use hydrogen bonding as a basis for assignment of secondary structure are DSSP [Kabsch and Sander, 1983] and STRIDE [Frishman and Argos, 1995].
On the other hand secondary structure can be defined using geometric features such as distances and dihedral angles of $C_i$ atoms along the backbone in addition to other local structural features. In fact, there is a direct correlation between patterns of hydrogen bonding and the geometry that arise out of them. However, secondary structural elements can deviate substantially from ideal geometry, therefore posing severe challenges to detect such elements using geometric features alone. Among the methods that rely primarily on geometry to assign secondary structure are [Levitt and Greer, 1977, Richards and Kundrot, 1988, Sklenar et al., 1989, Labesse et al., 1997, Srinivasan and Rose, 1999, Dupuis et al., 2004, Taylor, 2001, Majumdar et al., 2005]. (See Majumdar et al. [2005] for details of popular programs that assign secondary structural elements.)

We note that previous comparative studies have highlighted the difficulties of existing programs to assign consistently secondary structure to coordinate data and have proposed using a ‘consensus’ definition – secondary structure assignment that is at the intersection of all the methods – to arrive at a reliable simplification of protein structures [Colloch et al., 1993, Cuff and Barton, 1999].

The main goal for abstracting protein structures must be to achieve maximal economy of description with minimal loss of structural information [Taylor, 2001]. However simplifying structures at the level of standard secondary structure is lossy because the loop regions are ignored. Therefore, a reliable method that achieves the above goal and that is tolerant to measurement error and noise is preferred. Even better would be a method entirely independent of preconceived notions of what substructures are being sought.

Here we describe a method that generates a principled abstractions of protein structures. Our method uses the rigorous statistical framework of Minimum Message Length (MML). In fact, the realization of the goal to maximize economy and minimize loss of information fits squarely into the MML criterion, making it extremely well-suited for this specific problem. In this work we treat a protein as an ordered list of $C_i$ atoms along the backbone in addition to other local structural features. However, secondary structural elements can deviate substantially from ideal geometry, therefore posing severe challenges to detect such elements using geometric features alone. Among the methods that rely primarily on geometry to assign secondary structure are [Levitt and Greer, 1977, Richards and Kundrot, 1988, Sklenar et al., 1989, Labesse et al., 1997, Srinivasan and Rose, 1999, Dupuis et al., 2004, Taylor, 2001, Majumdar et al., 2005]. (See Majumdar et al. [2005] for details of popular programs that assign secondary structural elements.)

We note that previous comparative studies have highlighted the difficulties of existing programs to assign consistently secondary structure to coordinate data and have proposed using a ‘consensus’ definition – secondary structure assignment that is at the intersection of all the methods – to arrive at a reliable simplification of protein structures [Colloch et al., 1993, Cuff and Barton, 1999].

The main goal for abstracting protein structures must be to achieve maximal economy of description with minimal loss of structural information [Taylor, 2001]. However simplifying structures at the level of standard secondary structure is lossy because the loop regions are ignored. Therefore, a reliable method that achieves the above goal and that is tolerant to measurement error and noise is preferred. Even better would be a method entirely independent of preconceived notions of what substructures are being sought.

Here we describe a method that generates a principled abstractions of protein structures. Our method uses the rigorous statistical framework of Minimum Message Length (MML). In fact, the realization of the goal to maximize economy and minimize loss of information fits squarely into the MML criterion, making it extremely well-suited for this specific problem. In this work we treat a protein as an ordered list of $C_i$ coordinates. Our method uses an information-theoretic approach to explain as a line segment the points between any pair of residues in the structure. Each such explanation is encoded in a certain number of bits (or code length). Using these code lengths, a globally optimal explanation is computed which minimizes the total encoded (message) length of the given coordinate data. The code lengths contributing to this minimum message length result in the best piecewise linear approximation of the structure. In a stark contrast to the existing methods, our method is completely free of parameters and thresholds. We emphasize that our method is not a method for delineating secondary structures. However, as expected from such a method, our results show that the line segments generated by this method correspond well with standard secondary structures of proteins.

We note that this paper generalizes to three dimensions the work of Banerjee et al. [1996], who described a polygonal approximation method on general two dimensional sequence of points. Indeed, it can be shown that our method described in this paper can be generalized to arbitrary dimensions and other types of structural data (over and beyond proteins). We have attempted to keep the notations in this paper consistent with those described in the work of Banerjee et al. [1996] for the convenience of the reader.

Section 2 briefly summarizes the MML framework, followed by Sections 3-6 which describe the mechanics of our approach. Section 7 presents an analysis of the results of our method over a large number of protein structures.

2 THE MINIMUM MESSAGE LENGTH FRAMEWORK

Wallace and Boulton [1968] first proposed the theory of Minimum Message Length, where given a set of competing hypotheses (or models) that can explain some observed data, the MML criterion provides a statistically rigorous framework for selecting the best hypothesis to describe the data. In many ways MML is a formal information-theoretic realization of the principle of Occam’s razor.

Assume there are some observed data $D$ and some hypothesis $H$ that explains the data. From Bayes’s theorem we get

$$p(H|D) = \frac{p(H) \times p(D|H)}{p(D)}$$

where $p(H|D)$ is the joint probability of data $D$ and the hypothesis $H$, $p(H)$ is the prior probability of hypothesis $H$, $p(D|H)$ is the prior probability of data $D$, $p(H|D)$ is the posterior probability of $H$ given $D$, and $p(D)$ is the likelihood.

MML applies the remarkable result from Shannon’s ‘Mathematical Theory of Communication’ [Shannon, 1948] that, given an event $E$ with a probability $p(E)$, the message length $l(E)$ for an optimal code is given by $l(E) = -\log_2(p(E))$ bits. Carrying this insight to the Bayes’s theorem, we get the following relationship between conditional probabilities in terms of optimal message lengths.

$$l(H|D) = l(H) + l(D|H) = l(D) + l(H|D).$$

The essence of inductive inference is to fit a model to a mass of observed data. For such an approach it is the hypothesis $H$ with the largest posterior probability $p(H|D)$ that is often preferred. Among the terms in the above equation, $p(H)$ (and hence $l(H)$) can usually be estimated well for some reasonable prior on hypotheses. At the same time, the likelihood $p(D|H)$ can also be estimated. But to estimate the posterior probability distribution $p(H|D)$, the prior of observed data $p(D)$ will be needed. Estimating $p(D)$ can be problematic and even impractical. However, for two competing hypotheses, $H$ and $H'$ we have

$$l(H|D) - l(H'|D) = l(H) + l(D|H) - l(H') - l(D|H'),$$

thereby eliminating the necessity to estimate $p(D)$ completely when comparing hypotheses.

MML is best understood through a communication process where a transmitter and a receiver are connected through one of Shannon’s communication channels. The objective is that a transmitter must send some data $D$ to the receiver. The transmitter and receiver must have previously agreed on a set of rules (that is, a code book) of communication using common knowledge and prior expectations. If the transmitter can find a good hypothesis, $H'$, to fit the data, (s)he will be able to transmit the data economically.

In MML, an explanation of the data comes as a two part message:

---

1 Banerjee et al. [1996] use a related minimum description length principle for their approach, which is a technique that was introduced a decade after Wallace and Boulton [1968] proposed the MML criterion. The two approaches are significantly different. See Wallace [2005] for a comparison.
1. transmit the hypothesis $H^*$ taking $l(H^*)$ bits, and
2. transmit the observed data $D$ given $H^*$ taking $l(D|H^*)$ bits.

Such a message paradigm ensures complete transparency in communication. That is, any information that is not common knowledge cannot be included except as a part of the message sent by the transmitter. Otherwise the message sent will be indecipherable by the receiver. There can be no hidden parameters in this framework of communication. In fact, this issue extends to stating and inferring real-valued parameters to an “appropriate” level of precision, which is pertinent to the current problem on hand.

The MML framework additionally offers “safety” in that if an inefficient code is used to encode a message, it can only make the hypothesis look less attractive than otherwise. Note that MML yields a natural hypothesis test: the null-model corresponds to an inefficient code is used to encode a message, it can only make the hypothesis look less attractive than otherwise.

An important aspect of MML framework is that it is tolerant to inefficiency in code and does not require the code to be asymptotically optimal. Here we use an asymptotically optimal Elias omega code which encodes the integral value $x$ using $\log x$ bits.

3 FORMULATING THE PROBLEM USING MINIMUM MESSAGE LENGTH

A protein $P = \{P_1, \cdots, P_n\}$ is a sequence of $n$ three-dimensional points corresponding to the coordinates (in $\mathbb{R}^3$) of $C_n$ atoms along the protein backbone, from its N- to C- terminus.$^3$

Define a piecewise linear approximation of $P$ as a subsequence of $k \leq n$ points from $P$ of the form $Q = \{Q_1 \equiv P_{i_1}, \cdots, Q_k \equiv P_{i_k}\}$ such that $1 \leq i_1 < \cdots < i_k \leq n$, and the first and last points of $Q$ are the same as the first and last points of $P$ (i.e., $Q_1 = P_1$ and $Q_k = P_k$).

Given some subsequence $Q$ of sequence of points $P$, the protein can be approximated (or simplified) using line segments drawn between every successive pair of points in the subsequence, $Q_{r}$ and $Q_{r+1}$, $1 \leq r < k$. We will use the term delineation to describe this piecewise linear approximation. Further, we will use the term endpoint to describe any point in $Q$. This is because any pair of consecutive points, $Q_r \equiv P_i$ and $Q_{r+1} \equiv P_{i+1}$, form endpoints for abstracting the points between $P_i$ and $P_{i+1}$ (inclusive) in the protein with a line segment. Note that a subsequence $Q$ with $k$ endpoints yields a delineation containing $k – 1$ line segments between successive endpoints.

The goal in this paper is to find the best delineation of a given set of coordinate data, where the objective is to select the best comes from defining the problem using the minimum message length criterion. Consistent to the communication process described in Section 2, the transmitter explains the data in $P$ with a hypothesis $Q$ and sends it as a message whose code length is globally minimum over all possible hypotheses. Receiver will then able to infer the entire data $P$ from the receive message to a reasonable level of precision using the general rules they have agreed upon as a part of the code book.

For the problem of delineating a structure from coordinate data, the transmitter will send the following two part message (refer Section 2):

1. The first part is the subsequence of points $Q$ which denotes the delineation of $P$. This is equivalent to transmitting the hypothesis $Q$ in $l(Q)$ bits.
2. The second part will contain the remainder of points in $P$ (that is, $P – Q$) that weren’t sent in the first part. In other words, these are the points in $P$ that are between the endpoints stated in $Q$. The statement of these points will be encoded as spatial deviations with respect to the line segments between endpoints. This is equivalent to transmitting the observed data $P$ given the hypothesis $Q$ over $l(P|Q)$ bits.

Therefore, as a part of the codebook, the transmitter and receiver must have agreed upon the encoding of the endpoints in $Q$ and the encoding of deviations of points $P – Q$ explained by line segments between successive endpoints in $Q$. Since the coordinate data of proteins is available at some fixed precision, the transmitter and receiver agree on the specific precision at which the data should be sent. We emphasize that the encoding of the above should allow the receiver to decode the message to the agreed precision.

4 CODE LENGTH TO STATE THE DELINEATION AND DATA UNDER MML CRITERION

In this section we will discuss the statement and transmission of the two part message described in Section 3.

4.1 Encoding the first part of the message

The first part pertains to the transmission of the delineation $Q$ containing $k$ endpoints. The transmitter must therefore state the number of points $k$. There are several optimal universal prefix codes available to encode integers. Here we use an asymptotically optimal Elias omega code which encodes the integral value $k$ in $\log^* k$ bits$^4$ [Elias, 1975].

Next, the coordinates of all endpoints are to be encoded. Each endpoint is a set of three real numbers of the form $(x, y, z)$. Published protein coordinate data contain three putatively significant figures after the decimal point, in Angstrom (Å) units. The transmitter can scale this data to one decimal precision and treat the coordinates as integers. Now, an optimal code to send these coordinates is for the transmitter to first send the coordinates of a bounding rectangular box, $(x_{\text{min}}, y_{\text{min}}, z_{\text{min}})$ and $(x_{\text{max}}, y_{\text{max}}, z_{\text{max}})$ over all possible values of $x$, $y$ and $z$ in the given data. Once this bounding box is specified, any $(x, y, z)$ coordinates within the box can be coded in

$^2$ We use the term sequence in this paper to mean an ordered list. This should not be confused with the primary sequence of amino acids of a protein.

$^3$ Assume that the protein $P$ is oriented such that $P_1$ is the origin, $P_2$ lies on the positive $x$-axis, and $P_3$ lies on the $xy$-plane. This is one of the many possible schemes that ensures that our method is invariant to rotation and translation of the frame-of-reference in which the coordinate data is defined. (See supplementary note for a detailed discussion on this issue.)

$^4$ $\log^* x = \log x + \log \log x + \cdots$ (over all positive terms)
Let information:

In the second part, the transmitter has to encode the data, two hypotheses.

Note that the coordinates of the bounding rectangular box is a constant

Deviations

The intermediate points \( P_{i+1} \cdots P_{j-1} \in P \). For any intermediate point \( P_r, i+1 \leq r \leq j-1 \), define three spatial deviations \( \Delta s_r, \Delta t_r \) and \( \Delta u_r \) in the reverse order, \( \Delta u_r \) is the signed perpendicular distance of \( P_r \) to the plane defined by vectors \( P_j - P_i \) and \( z \)-axis. To define \( \Delta t_r \), first project \( P_r \) to the plane defined above. Call this projection point \( \mathbf{P}_r \). Given this projection, \( \Delta t_r \) is the signed perpendicular distance of \( \mathbf{P}_r \) to the line \( L_{ij} \). Finally, the deviation \( \Delta s_r \) is the (unsigned) lateral distance along the line \( L_{ij} \) between points of projection of \( P_{r-1} \) and \( P_r \) onto the line (see Fig. 1). (Refer supplementary note containing a discussion on these deviations due to arbitrary rotation of the coordinates.)

Note that once the endpoints are transmitted (see Section 4.1), the receiver can deduce the value of \( \Delta u_r \)'s and \( \Delta t_r \)'s from four to three:

To explain the encoding of this part more clearly, consider Fig. 1.

Therefore, the message length to state the first part of the transmission requires \( k \log V \) bits.

4.2 Encoding the second part of the message

In the second part, the transmitter has to encode the data, \( P \in Q \), between endpoints stated in the first part of the message. For a successive pair of endpoints \( (Q_{i}, Q_{i+1} \in P) \), \( 1 \leq i < j \leq n, 1 \leq r' \leq k \in Q \), there are \( j-i-1 \) intermediate points between \( P_i \) and \( P_j \) in \( P \). In this work, these intermediate data points will be treated as noisy samples and will be stated as a set of spatial deviations with respect to the line segment between \( P_i \) and \( P_j \).

If such a scheme is used to communicate the second part of the message, for each line segment in \( Q \) between successive endpoints, the second part of the message will encode the following information:

1. the number of points explained by the line segment.
2. three spatial deviations for each intermediate point with respect to the line that will allow the receiver to recover the original location of the intermediate point up to a reasonable approximation.
3. the parameters of the probability distribution associated with each of the three sets of spatial deviations, over all intermediate points.

To explain the encoding of this part more clearly, consider Fig. 1. Let \( L_{ij} \) denote the line segment between two successive endpoints in \( Q \). \( Q_{i} \equiv P_i \) and \( Q_{i+1} \equiv P_j \). This line will be used to explain

\[ \log(x_{\text{max}} - x_{\text{min}}) + \log(y_{\text{max}} - y_{\text{min}}) + \log(z_{\text{max}} - z_{\text{min}}) = \log((x_{\text{max}} - x_{\text{min}})(y_{\text{max}} - y_{\text{min}})(z_{\text{max}} - z_{\text{min}})) \]

where \( V \) is the volume of the bounding rectangular box. It follows from here that all the \( k \) endpoints in \( Q \) can be stated in \( k \log V \) bits.\(^5\)

Therefore the message length to state the first part of the transmission requires \( k \log V \) bits.

Consider the calculations of these parameters. For the line \( L_{ij} \), there are \( j-i-1 \) intermediate points. Represent this quantity by the variable \( m_{ij} \). Then

\[ \mu_{\Delta s} = \frac{\sum_{r=i+1}^{j-1} \Delta s_r}{m_{ij}} \approx \frac{\sum_{r=i+1}^{j} \Delta s_r}{m_{ij}+1} = \frac{D_{ij}}{m_{ij}+1}, \]

where \( D_{ij} \) is the Euclidean distance between \( P_i \) and \( P_j \). Note that once the endpoints are transmitted (see Section 4.1), the receiver can deduce the value of \( \mu_{\Delta s} \) requiring no explicit statement for this parameter in the message. This reduces the number of parameters to be stated from four to three: \( (\mu_{\Delta s}, \sigma_{\Delta s}^2, \sigma_{\Delta u}^2) \).

We will now compute the code lengths to state the variance of three normal distributions. Variance for a Gaussian distribution is simply \( \text{mean squared minus squared mean} \):

\[ \sigma_{\Delta s}^2 = \frac{\sum_{r=i+1}^{j-1} (\Delta s_r - \mu_{\Delta s})^2}{m_{ij} - m_{ij}^2} = \frac{\sum_{r=i+1}^{j-1} \Delta s_r^2}{m_{ij}} - \mu_{\Delta s}^2 \]

Similarly, we have \( \sigma_{\Delta u}^2 = \frac{\sum_{r=i+1}^{j-1} u_r^2}{m_{ij} - m_{ij}^2} \), since \( \mu_{\Delta s} = \mu_{\Delta u} = 0 \). We note that the code length for each parameter varies with \( \frac{1}{2} \log m_{ij} \) bits. (See Chapter 5 of Wallace [2005]).

With the parameters of the distributions encoded, we will now compute the code lengths required to state the individual values of \( \Delta s \)'s. Since we have assumed that the distribution is a Gaussian, the probability distribution of a random variable \( \Delta s_r \) with parameters \( \mu_{\Delta s} \) and \( \sigma_{\Delta s}^2 \) is given by:

\[ \Delta s_r \sim N(\mu_{\Delta s}, \sigma_{\Delta s}^2) = \frac{1}{\sigma_{\Delta s}^2} \exp\left(-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma_{\Delta s}^2}\right) \]

Since we assumed that variables are independent, we have

\[ \Delta s_{s_r} \sim N(\mu_{\Delta s}, \sigma_{\Delta s}^2) = \frac{1}{\sigma_{\Delta s}^2} \exp\left(-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma_{\Delta s}^2}\right) \]

\[ \Delta s_{s_r} \sim N(\mu_{\Delta s}, \sigma_{\Delta s}^2) = \frac{1}{\sigma_{\Delta s}^2} \exp\left(-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma_{\Delta s}^2}\right) \]

\[ \Delta s_{s_r} \sim N(\mu_{\Delta s}, \sigma_{\Delta s}^2) = \frac{1}{\sigma_{\Delta s}^2} \exp\left(-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma_{\Delta s}^2}\right) \]
\[ p(\Delta s_{i+1}, \ldots, \Delta s_j | \mathcal{N}(\mu, \sigma^2)) = \prod_{r=i+1}^{j-1} \frac{1}{\sqrt{2\pi}\sigma_{\Delta r}} e^{-\frac{(\Delta x_r - \mu)^2}{2\sigma_{\Delta r}^2}} \]

This implies,
\[ p(\Delta s_{i+1}, \ldots, \Delta s_j | \mathcal{N}(\mu, \sigma^2)) = \left( \frac{1}{\sqrt{2\pi}\sigma_{\Delta r}} \right)^m_{ij} e^{-m_{ij}^{s}}. \]

Therefore, using Shannon’s insight, the optimal code length to describe the entire sets of individual deviations of \( \Delta s \)'s for a line \( L_{ij} \) will require \( -\log(p(\Delta s_{i+1}, \ldots, \Delta s_j | \mathcal{N}(\mu, \sigma^2))) \) = \( -\log \left( \left( \frac{1}{\sqrt{2\pi}\sigma_{\Delta r}} \right)^m_{ij} e^{-m_{ij}^{s}} \right) = m_{ij}^{s} \log (2\pi e \sigma_{\Delta r}^2) \) bits.

Following a similar expansion, we can show that the code lengths for the deviation \( t_r \)'s and \( u_r \)'s are \( m_{ir}^{t} \log (2\pi e \sigma_{t_r}^2) \) and \( m_{ir}^{u} \log (2\pi e \sigma_{u_r}^2) \) respectively.

So far in this second part we have computed the code lengths required to state intermediate points explained by the line \( L_{ij} \). Note that a delineation of a structure containing \( k \) endpoints defines \( k - 1 \) such line segments. For convenience in notation, assume the endpoints of each line segment is of the form \((P_{kr}, P_{jr})\), \( 1 \leq r < k \). (In practice, for a delineation, \( P_{kr} \) of the \( r \)th line segment is equivalent to \( P_{kr-1} \) of \((r-1)\)th line segment.) Then the total code length of the second part is the sum of the following terms:

\[
\sum_{r=1}^{k-1} \log^* m_{ir}^{t}, \quad m_{ir}^{t} = j_r - i_r - 1, \quad \text{representing the total code length to encode the number of intermediate points described by all line segments in the delineation put together.}
\]
\[
\sum_{r=1}^{k-1} 3 \log^* m_{ir}^{t} \quad \text{bits to encode the parameters (three per line segment) corresponding to the distribution of spatial deviations for all lines.}
\]
\[
\sum_{r=1}^{k-1} m_{ir}^{t} \log (2\pi e \sigma_{t_r}^2) \quad \text{bits to encode \( t_r \)'s over all line segments.}
\]
\[
\sum_{r=1}^{k-1} m_{ir}^{u} \log (2\pi e \sigma_{u_r}^2) \quad \text{bits to encode \( u_r \)'s over all line segments.}
\]

5 FINDING THE OPTIMAL DELINEATION

This section will describe the procedure to compute the optimal delineation \( Q^{*} \) for a given coordinate data. Broadly, the search for the optimal delineation has two steps.

Potentially every pair of points \( P_{i}, P_{j} \), \( 1 \leq i < j \leq n \) can be a part of the delineation in \( Q^{*} \). (We note here that the segments in the delineation must not overlap, except for successive regions, and those only at their endpoints.) Therefore we will first build a matrix \( \mathcal{H} = (\mathcal{H}_{ij})_{1 \leq i < j \leq n} \) of code lengths for all possible pairs of points in \( P \).

Then, the matrix \( \mathcal{H} \) will be used to find a subsequence of points \( Q^{*} \) such that the total code length \( \mathcal{H}(Q^{*}) \) of the delineation is minimized, using a one-dimensional dynamic program.

5.1 Computation of code length over all possible segments

Equation 2 expresses the message length \( \mathcal{H}_{ij} \) required to describe any line segment \( L_{ij} \) between two points \( P_{i} \) and \( P_{j} \). We will examine the complexity of computing each of the components that constitute Equation 2.

For the \( n \) points in \( P \), there are \( nC_{2} = \frac{n(n-1)}{2} \) possible line segments. The \( \log V \) term in Equation 2 is constant across all possible segments and is computed once while reading the data points of \( P \). Next, for each line segment, there are 3 parameters whose code lengths depend on the number of points in between the endpoints. This is trivially computed in constant time as \( j - i - 1 \).

The relative complex part is to compute the code lengths of the spatial deviations of the line, \( \Delta s \)'s \( t_r \)'s and \( u_r \)'s. Each of these three deviations have code lengths that depend on their respective variance, \( \sigma_{\Delta r}^2, \sigma_{t_r}^2 \) and \( \sigma_{u_r}^2 \). While one can compute the variance of each set of deviations from the coordinate data, such a computation is linear in the number of points that each line segment explains. If this naïve approach is followed, the computation of \( \mathcal{H} \) requires \( O(n^3) \) operations. We will show in the later Section 6 that this is redundant and that the total time required to compute \( \mathcal{H} \) can be achieved in \( O(n^2) \) operations, by computing the variances of all
three spatial deviations incrementally from previous computations using a set of sufficient statistics. But before that we will describe the method to compute the optimal delineation given the matrix $H$.

## 5.2 Optimal delineation as a one-dimensional dynamic program

Dynamic programming is perfectly suited when dealing with problems that contain sequential constraints, where the solutions to the subproblems have a recursive overlapping substructure [Bellman, 1957]. The problem statement in Section 4.3 is an ideal candidate for the search strategy of dynamic programming. Since a delineation is a subsequence which preserves the linear ordering of its elements, the optimal delineation of the given data can be derived by computing and memoizing (i.e., caching) the optimal delineation of its subproblems.

We will use the matrix $H$ of code length between all possible endpoints to find the optimal delineation $Q^*$ that minimizes $H(Q^*)$ using a one-dimensional dynamic program.

Let $C_i$ be an array that stores the optimal code length of delineating points $P_1, \ldots, P_i$, $\forall 1 \leq i \leq n$. The objective is to find the delineation of the given points where $C_n$ is minimum over all possible subsequences of the given points. Therefore, the recurrence relationship of optimal costs using a one-dimensional dynamic program is as follows:

$$
C_1 = 0,
C_j = \min_{1 \leq i < j} \{ H_{ij}, (C_i + H_{ij}) \}, \forall 1 \leq j \leq n
$$

In other words, the optimal code length to delineate the points $P_1, \ldots, P_j$ $(1 \leq j \leq n)$ builds on the optimal code length to delineate from $P_1, \ldots, P_i$, if and only if the value of $C_i$ plus the code length to state a new line segment $H_{ij}$ is minimum, over all $1 \leq i < j$.

Using the above relationship, the array $C$ is filled iteratively from 1 to $n$. Upon completion, the value $C_n$ gives the optimal message length corresponding to the best delineation $Q^*$ of $P$, where $H(Q^*) \equiv C_n$ is globally minimum. The subsequence of endpoints of this optimal delineation can be computed by storing, for each $j$, the back pointer $i < j$ of the array from which the optimal value $C_i$ was derived. With these back pointers, a simple traceback from $C_n$ (until $C_1$ is reached) gives the set endpoints (in reverse order) that form the best delineation $Q^*$.

## 6 EFFICIENT COMPUTATION OF MATRIX $H$

As mentioned in Section 5.1 the matrix of code lengths $H$ can be computed efficiently in $O(n^2)$ operations and this section will show how this can be achieved.

For the matrix $H$ to be computable in $O(n^2)$ operations, each element $H_{ij}$ in the matrix should be computable in constant time. However terms $\sigma^2_{x_i}$, $\sigma^2_{y_i}$, and $\sigma^2_{z_i}$ in Equation 2 cannot be computed in constant time. For a line segment $L_{ij}$, naïvely, these three variances take time proportional to the number of points explained by the line to compute, leading to a $O(n^3)$ algorithm for computing the matrix $H$. Below we will show that each of $\sigma^2_{x_i}$, $\sigma^2_{y_i}$, and $\sigma^2_{z_i}$ can indeed be computed incrementally and in constant time from previous computations resulting in a $O(n^2)$ algorithm.

### 6.1 Constant-time update of $\sigma^2_{x_i}$’s

Consider first these notations: for any vector $\bar{v}$ with direction ratios $(x, y, z)$, let $||v|| = \sqrt{x^2 + y^2 + z^2}$ represents the vector norm of $\bar{v}$. Let any point $P_i \in P$ have the direction ratios of the form $(x_i, y_i, z_i)$.

By the definitions of the spatial deviations in Section 4.2, any $\Delta s_{ij}, 1 \leq i < j \leq n$ is the scalar associated with the projection of the vector $(P_{j} - P_{i-1})$ onto the vector $L_{ij} \equiv (P_j - P_i)$. (Refer Fig 1) Let $\hat{L}_{ij} = (\hat{L}_{ij}^x, \hat{L}_{ij}^y, \hat{L}_{ij}^z)$ represent the direction cosines of the vector $L_{ij}$, where $\hat{L}_{ij}^x = (x_j-x_i)/||L_{ij}||$, $\hat{L}_{ij}^y = (y_j-y_i)/||L_{ij}||$, and $\hat{L}_{ij}^z = (z_j-z_i)/||L_{ij}||$. Then $\Delta s_{ij}$ is the dot product of $(P_j - P_{i-1})$ and $\hat{L}_{ij}$:

$$
\Delta s_{ij} = (P_j - P_{i-1}) \cdot \hat{L}_{ij},
$$

Expanding this we get,

$$
\Delta s_{ij} = (x_j-x_{i-1})L_{ij}^x + (y_j-y_{i-1})L_{ij}^y + (z_j-z_{i-1})L_{ij}^z
$$

Denoting $S_{ij} = \sum_{r=i+1}^{j-1} \Delta s_{ir}^2$

Expanding $S_{ij}$,

$$
S_{ij} = \sum_{r=i+1}^{j-1} ((x_r - x_{i-1})L_{ij}^x + (y_r - y_{i-1})L_{ij}^y + (z_r - z_{i-1})L_{ij}^z)^2
$$

Now, let $S_{ij}^x, S_{ij}^y, S_{ij}^z, S_{ij}^{xy}, S_{ij}^{yz}, S_{ij}^{zz}$ be a set of variables which we will call here sufficient statistics. These variables are of the form:

$$
S_{ij}^{AB} = \sum_{r=i+1}^{j-1} (A_r - A_{r-1})(B_r - B_{r-1}), \text{ where } A \text{ and } B \text{ take the values } \{x, y, z\}.
$$

Expressing Equation 3 in terms of the sufficient statistics, we get

$$
S_{ij} = L_{ij}^x^2 S_{ij}^{xx} + L_{ij}^y^2 S_{ij}^{yy} + L_{ij}^z^2 S_{ij}^{zz} + 2L_{ij}^x L_{ij}^y S_{ij}^{xy} + 2L_{ij}^x L_{ij}^z S_{ij}^{xz} + 2L_{ij}^y L_{ij}^z S_{ij}^{yz} + 2L_{ij}^x S_{ij}^{zy} + 2L_{ij}^y S_{ij}^{yx} + 2L_{ij}^z S_{ij}^{zx}
$$

From Equation 4 it can be clearly seen that any $S_{ij+1}$ can be updated from $S_{ij}$ in constant time, using the sufficient statistics. This holds because any $S_{ij+1}^{AB} = S_{ij}^{AB} + (A_j - A_{j-1})(B_j - B_{j-1})$, where \{A, B\} $\in \{x, y, z\}$.

Therefore, using the sufficient statistics the computation of $\sigma^2_{x_i}$ for a line segment can be computed incrementally in constant time.

### 6.2 Constant-time update of $\sigma^2_{z_i}$’s

Let $\hat{n}_i$ be the normal to a plane defined by $\hat{z} \times \hat{L}_{ij}$, where $\hat{z}$ is the unit vector along z-axis with the direction
cosines $(0, 0, 1)$. It follows that the direction ratios of $\vec{n}_1$ are 
$\langle -(y_i - y_j), (x_j - x_i), 0 \rangle$.

Define $\vec{n}_2$ as a vector which is normal to the plane $\vec{L}_{ij} \times \vec{n}_1$. The
direction ratios of $\vec{n}_2$ will be:
$\langle -(x_j - x_i)(x_j - x_i), -(y_j - y_i)(x_j - x_i), (y_j - y_i)^2 \rangle$
Let $\hat{n}_2 = \langle \hat{n}_{2x}, \hat{n}_{2y}, \hat{n}_{2z} \rangle$ represent the direction cosines of $\vec{n}_2$,
where $\hat{n}_{2x} = \frac{(x_j - x_i)(x_j - x_i)}{||\vec{n}_2||}$, $\hat{n}_{2y} = \frac{-(y_j - y_i)(x_j - x_i)}{||\vec{n}_2||}$ and $\hat{n}_{2z} = \frac{(y_j - y_i)^2}{||\vec{n}_2||}$.

Then $t_e = (P_e - P_1) \cdot \hat{n}_2$. (Refer Fig 1.) This implies
$$\Delta T_{ij} = \sum_{r=i+1}^{j-1} \hat{G} \text{ and expanding along the steps we took in}
$$
the previous section, we get
\[
T_{ij} = \hat{n}_1^2 T_{xx} + \hat{n}_2^2 T_{yy} + \hat{n}_3^2 T_{zz} + 2 \hat{n}_1 \hat{n}_2 T_{xy}
+ 2 \hat{n}_1 \hat{n}_3 T_{xz} + 2 \hat{n}_2 \hat{n}_3 T_{yz} \tag{5}
\]
where computation of any $T_{ij+1}$ can be updated from $T_{ij}$ in
constant time.

6.3 Constant-time update of $\sigma^2$’s

We have seen above that $\vec{n}_1 = \langle -(y_j - y_i), (x_j - x_i), 0 \rangle$. Let $\vec{n}_1 = \langle \hat{n}_{1x}, \hat{n}_{1y}, 0 \rangle$ represent the direction cosines of $\vec{n}_1$, where
$\hat{n}_{1x} = \frac{(y_j - y_i)}{||\vec{n}_1||}$ and $\hat{n}_{1y} = \frac{(x_j - x_i)}{||\vec{n}_1||}$. (Note $\hat{n}_{1z} = 0$).

Then $u_e = (P_e - P_1) \cdot \hat{n}_1$. (Refer Fig 1.) Expanding as before we get
\[
U_{ij} = \hat{n}_1^2 U_{xx} + \hat{n}_2^2 U_{yy} + \hat{n}_3^2 U_{zz} + 2 \hat{n}_1 \hat{n}_2 U_{xy}
+ 2 \hat{n}_1 \hat{n}_3 U_{xz} + 2 \hat{n}_2 \hat{n}_3 U_{yz} \tag{6}
\]
where again the computation of any $U_{ij+1}$ can be updated from $U_{ij}$ in
constant time, when sufficient statistics are maintained.

Therefore, the update rules in Equations 4-6 allows an efficient
computation of the matrix $H$ of code lengths in $O(n^2)$ operations.

7 RESULTS

In the previous sections we have demonstrated an efficient and
statistically robust algorithm to simplify a protein structure with
piecewise linear segments. We implemented the described algorithm

We evaluated our method using a non-redundant dataset
containing 15,399 protein structures obtained from the protein data
bank [Berman et al., 2002]. (The non-redundancy here implies
that no two structures in this dataset share a sequence identity greater
than 65%.) This dataset was culled using the program
PISCES [Wang and Dunbrack, 2003]. The list of proteins structures
in the dataset and the results of their delineation produced by our
method can be obtained from the aforementioned link.

Fig. 2 gives the distribution of the measure of simplification of
structures over the entire dataset. For a structure, the measure of
simplification is the ratio of number of line segments identified by
the program over the number of residues in the structure. On
an average over the entire dataset the delineation size (that is, the
number of line segments in the delineation) constitutes 13.85% of
the total size of structure (in residues). In addition, the average
segment length over the entire dataset is observed to be 8.11
residues. In general the number of segments is correlated to total
size of the protein structure.

<table>
<thead>
<tr>
<th>Type</th>
<th>$\phi$</th>
<th>$\psi$</th>
<th>$n$</th>
<th>$\rho$</th>
<th>$p$</th>
</tr>
</thead>
<tbody>
<tr>
<td>310-Helix</td>
<td>-57.1</td>
<td>-69.7</td>
<td>3.0</td>
<td>2.0</td>
<td>6.0</td>
</tr>
<tr>
<td>$\alpha$-Helix</td>
<td>-57.8</td>
<td>-47.0</td>
<td>3.6</td>
<td>1.5</td>
<td>5.5</td>
</tr>
<tr>
<td>$\pi$-Helix</td>
<td>-74.0</td>
<td>-4.0</td>
<td>4.4</td>
<td>1.1</td>
<td>5.0</td>
</tr>
<tr>
<td>$\beta$-Strand</td>
<td>-139.0</td>
<td>135.0</td>
<td>2.0</td>
<td>3.4</td>
<td>6.8</td>
</tr>
</tbody>
</table>
Examining the coarse segment level assignment for the structures in the dataset, we note that the average length of segments assigned as ‘Helix’ is 13.01 residues while the same for those assigned as ‘Strand’ is 7.33 residues.

To evaluate our coarse assignment, we choose two popular and extensively used secondary structure assignment programs, DSSP [Kabsch and Sander, 1983] and STRIDE [Frishman and Argos, 1995]. DSSP and STRIDE assign each residue to one of multiple secondary structural states, including 3_10-, α-, π-helices and β-strands of sheet. For the structures in our dataset, we generate the respective secondary structure assignments using DSSP and STRIDE. We note that both these programs assign secondary structure definitions at a residue level, while the coarse assignment for our method described above is at a segment level. Therefore, to enable a comparison between the methods we assign all residues within a segment to the segment level secondary structure state.

Table 2 gives the concordance of Helix\(^6\) and Strand assignments between DSSP, STRIDE, and our method, PMML.

Although even a coarse level segment assignment by our method produced a satisfactory concordance with DSSP and STRIDE, there is still a disagreement of \(\sim 15\%\) between PMML and the other two methods. Inspecting these differences we note that the majority of them came from the terminal parts of the segments delineated by our program. Therefore we refine the coarse level assignment produced by PMML using the hydrogen bonding patterns of residues within each segment to reassign the secondary structure state at a residue level. We use a simple proximity (of backbone nitrogen and carbonyl groups) and angle (of N,O,C atoms) based computation of hydrogen bonds. Comparing our refined assignments at a residue level with DSSP and STRIDE we notice a substantial improvement in the concordance of helix and strand assignments with DSSP and STRIDE. (See rows 3 and 4 in Table 2.) We emphasize that although PMML can be used to generate protein secondary structure assignments, its real aim is to generate concise representations of structures, irrespective of the nature of the segments of which they are composed. For instance, PMML could be applied to RNA structures without needing any appeal to the types of substructure anticipated.

Table 2. Percentage agreement of Helix and Strand assignments between various methods

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Helices</th>
<th>Strands</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMML (coarse)_vs_DSSP</td>
<td>79.0%</td>
<td>83.3%</td>
</tr>
<tr>
<td>PMML (coarse)_vs_STRIDE</td>
<td>79.3%</td>
<td>83.1%</td>
</tr>
<tr>
<td>PMML (refine)_vs_DSSP</td>
<td>92.6%</td>
<td>92.4%</td>
</tr>
<tr>
<td>PMML (refine)_vs_STRIDE</td>
<td>91.3%</td>
<td>92.1%</td>
</tr>
<tr>
<td>STRIDE_vs_DSSP</td>
<td>95.7%</td>
<td>96.9%</td>
</tr>
</tbody>
</table>

Manually evaluating the delineation of a large number of structures we notice that although PMML’s delineation identifies the regions of helix and strand consistently, there remain small discrepancies in assigning precise beginning and end residues of secondary structure elements as ascertained by an expert. To highlight these differences consider the following example of the delineation produced by PMML. Fig. 3 shows the structure of oxidized *Clostridium beijerinckii* flavodoxin. This protein binds a cofactor, flavin mononucleotide (FMN). Flavodoxin is a small α/β protein, containing a 5-stranded parallel β-sheet (β\(_1\), · · · , β\(_5\)) with two helices packed against each face of the sheet (α\(_A\), α\(_C\), and α\(_E\)). There is also a short helix (α\(_D\)) located near the N-terminus of the protein. (Fig. 3.) Different segments produced by PMML are shown in different colours. The elements of secondary structure shown as thick ribbons are the secondary structure assignments taken from the structure’s wwPDB file 5NLL. Table 3 gives the residue ranges (that is, start and end residues) for each secondary structural element (SSE) of the flavodoxin structure listed in its wwPDB file. The residue ranges of the corresponding segmentation produced by PMML is also presented in the table. Broadly, the program correctly assigns segments to the SSEs.

![Fig. 3. Wall-eye stereo image of 1.8 Å crystal structure of oxidized *Clostridium beijerinckii* flavodoxin. Each delineated segment produced by PMML is shown in a different color. The elements of secondary structures, of helices and strands of sheet, were derived from the wwPDB file 5NLL, and are shown in this figure as thick ribbons. The labels of various secondary structures are also shown. The bound FMN co-factor is shown at the top of the structure as thin lines.](image)

Table 3. The residue ranges of secondary structural elements (SSEs) in the structure of flavodoxin shown in Fig. 3. The SSEs in the rows follow the order of their appearance along the chain of the protein from its N- to C- terminus. The column wwPDB gives the residue ranges of various SSEs as indicated in the wwPDB file 5NLL. The column PMML gives the corresponding residue ranges of the segmentation produced by PMML.

<table>
<thead>
<tr>
<th>SSE</th>
<th>wwPDB</th>
<th>PMML</th>
</tr>
</thead>
<tbody>
<tr>
<td>β(_1)</td>
<td>Lys2-Trp6</td>
<td>Met1-Tyr5</td>
</tr>
<tr>
<td>α(_A)</td>
<td>Asn11-Glu25</td>
<td>Asn11-Glu27</td>
</tr>
<tr>
<td>β(_2)</td>
<td>Asn31–Asn34</td>
<td>Gly27-Ile33</td>
</tr>
<tr>
<td>α(_B)</td>
<td>Ile40–Asn45</td>
<td>Asn39–Glu46</td>
</tr>
<tr>
<td>β(_3)</td>
<td>Ile48–Cys53</td>
<td>Asp47–Cys53</td>
</tr>
<tr>
<td>α(_C)</td>
<td>Phe66–Lys76</td>
<td>Glu65–Thr75</td>
</tr>
<tr>
<td>β(_4)</td>
<td>Lys81–Tyr88</td>
<td>Gly79–Ser87</td>
</tr>
<tr>
<td>α(_D)</td>
<td>Lys94–Gly105</td>
<td>Gly91–Gly107</td>
</tr>
<tr>
<td>β(_5)</td>
<td>Leu115–Gln118</td>
<td>Glu112–Gln118</td>
</tr>
<tr>
<td>α(_E)</td>
<td>Asp122–Ile136</td>
<td>Glu120–Gln126,Gln126–Ile136</td>
</tr>
</tbody>
</table>

\(^6\) We do not distinguish between the three types of helices and treat them as one state.
However, minor differences can be observed in the locations of their start and end residues. In most cases we notice an absolute difference of 1 or 2 residues in the N- or C-terminal regions of these SSEs $\alpha E$, $\beta_2$ and $\beta_3$ show some discrepancies. The residue range from wwPDB corresponding to $\alpha E$ was approximated by PMML using 2 segments instead of one. The first segment is composed of roughly one turn of the helix at $\alpha E$'s N-terminal end. This is understandable as this turn is substantially skewed from the main helical axis and, indeed, there is an interruption in the hydrogen bonding. However, the second segment composed of 11 residues in this region is consistent with the assignment in the wwPDB file. In the case of $\beta_2$, the start location identified by PMML precedes the start location identified in the wwPDB file by 4 residues. On inspecting the flavodoxin structure, there appears to be a backbone hydrogen bond between the carbonyl group of residue Asp29 and the nitrogen of Met1 (of strand $\beta_2$), so the $\beta_2$ strand may well start at residue Lys28 or Asp29. Similarly, for $\beta_3$, the start location of the segment from PMML was identified to be 3 residues before the location identified in the wwPDB file, and inspecting the structure, we note the $\beta$-bulge in strand $\beta_3$, and hydrogen bonds between atoms 800-...109N and 82N-...109O; assignment of the start of the strand $\beta_3$ to residue 109 is not indefensible.

8 CONCLUSION

We have presented a novel and efficient method to delineate protein structures using the minimum message length (MML) framework; MML is tolerant to measurement error and other inaccuracies. The model used in this work is independent of preconceived notions of what substructures are being sought to simplify the observed coordinate data. Our method maximizes the economy of representation while minimizing the loss of information, taking into account even the loop regions of proteins. Analysis of the delineations of a large number of protein structures suggests that the method is consistent in, among others, delineating standard secondary structures. The concise representations produced by this method have a potential use for rapid and accurate structure comparison and lookup. An implementation of our program is available from http://www.csse.monash.edu.au/~karun/pmml/.

9 ACKNOWLEDGEMENTS

ASk’s research is supported by Monash Talent Enhancement and Larks Fellowship program. LA and ASK thank Nathan Hurst for useful pointers during the development of this work. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council. We thank the anonymous referees for comments that improved the manuscript.

REFERENCES

Discussion on the effect of rotation of the coordinates on the method

In this supplementary section, we discuss the effect of rotation of the coordinate data on the message lengths and the resulting segmentation. (It is easy to see that the method described in the paper is invariant to translation.)

A protein is specified in some arbitrary orientation using \((x, y, z)\) coordinates. In our method, the hypothesis (i.e., the collection of endpoints of the segmentation) is transmitted using \(\log^k k + k \log V\) bits, where \(V\) is the volume of a bounding box (see Section 4.1 in the main text). However, when the protein is rotated, the bounding box changes, and hence \(V\), resulting in different message lengths. This can be trivially avoided by defining the bounding box in some canonical way.

Again, in our method, each intermediate point corresponding to a line segment is transmitted as spatial deviations, i.e., in \((\Delta s, t, u)\) coordinates, where \(\Delta s\) is along the line segment, and \(t\) and \(u\) are in the plane that contains the point, and is normal to the line segment. For the receiver to decode the coordinate (to a reasonable precision) from the spatial deviations, the \(t\) and \(u\) axes must lie in the perpendicular plane but their choice is otherwise “arbitrary”.

The intermediate point is transmitted as \(\Delta s, t\) and \(u\) deviations from a predicted location. This location is the centre of a probability distribution, the spread of which shows the uncertainty.

For a protein, it is straightforward to model \(\Delta s\) as independent of \(t\) and \(u\) and the choice of the perpendicular plane, as the lateral distances along the line, \(\Delta s\)’s, are invariant to rotation. However, \(t\)’s and \(u\)’s change under rotation of the coordinates.

The simplest assumption with \(t\) and \(u\) deviations is to also model them independently of each other (as in the main text), since there is no obvious reason to believe that any observed deviations in the \(t\) and \(u\) directions are correlated, thus leading to 3 independent standard deviations as described in the main text: \(\sigma_{\Delta s}, \sigma_t, \text{ and } \sigma_u\).

Since the choice of the \(t\) and \(u\) axes is arbitrary except that they must lie in the perpendicular plane arbitrarily chosen, if \(\sigma_t\) and \(\sigma_u\) differ, a random rotation of the axes, \((t, u)\) to \((t', u')\) will result in a different probability distribution, making the method sensitive to rotation of the coordinates. A principled scheme to overcome this sensitivity is to choose the rotationally invariant bivariate normal distribution where the model assumes \(\sigma_t \equiv \sigma_u\). Under this assumption, if the coordinates are rotated by an arbitrary angle, the distribution still remains exactly the same, making this procedure completely rotation-invariant.

Alternatively, ‘less principled’ strategies to overcome the arbitrariness of the choice of \(t\) and \(u\) axes is to choose these spatial deviations in some rotation-independent way. For example, this can be done with respect to the previous transmitted segment, or by transmitting in the second part of the two part
message, a reference point for every line segment, based on which deviations $t$ and $u$ can be computed.

In practice, based on the experiments we conducted, the choice of the strategy resulted in the message lengths to change, if only slightly (by up to 2%) between the methods. This indeed can cause minor differences in the endpoints of the resulting segmentation.