BIOINFORMATICS

Vol. 27 ISMB 2011, pages i43–i51
doi:10.1093/bioinformatics/btr240

Piecewise linear approximation of protein structures using the principle of minimum message length

Arun S. Konagurthu[1,*], Lloyd Allison[1,*], Peter J. Stuckey[2] and Arthur M. Lesk[3]

[1]Clayton School of Information Technology, Monash University, Clayton, VIC 3800, [2]Department of Computer Science and Software Engineering, The University of Melbourne, Parkville, VIC 3010 Australia and [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 article 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.

Availability: http://www.csse.monash.edu.au/~karun/pmml.

Contact: arun.konagurthu@monash.edu; lloyd.allison@monesh.edu

1 INTRODUCTION

With the rapid growth in the corpus of known structures, concise representations are increasingly preferred to inspect and analyze protein folding patterns (Abagyan and Maiorov, 1988; Lesk, 1995; Richardson, 1981; Taylor *et al.*, 1983). 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 (Kamat and Lesk, 2007; Konagurthu and Lesk, 2010; Lesk and Chothia, 1980; Lesk, 1995).

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, π-helix and 3_{10}-helix) and *strand* of sheet (Edsall *et al.*, 1966). Ideally, the spatial trace of α-Carbon (C_α) 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 C_α 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 Maiorov, 1988; Konagurthu *et al.*, 2008; Mizuguchi and Go, 1995; Shi *et al.*, 2007).

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_α 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

*To whom correspondence should be addressed.

(Dupuis *et al.*, 2004; Labesse *et al.*, 1997; Levitt and Greer, 1977; Majumdar *et al.*, 2005; Richards and Kundrot, 1988; Sklenar *et al.*, 1989; Srinivasan and Rose, 1999; Taylor, 2001). (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 (Colloc'h *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_α 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 article generalizes to three dimensions the work of Banerjee *et al.* (1996), who described a polygonal approximation method on general two dimensional sequence of points.[1] 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.

[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.

2 THE MINIMUM MESSAGE LENGTH FRAMEWORK

Wallace and Boulton (1968) first proposed the theory of MML, 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

$$p(H\&D) = p(H) \times p(D|H) = p(D) \times p(H|D)$$

where $p(H\&D)$ is the joint probability of data D and the hypothesis H, $p(H)$ is the *prior* probability of hypothesis H, $p(D)$ is the prior probability of data D, $p(H|D)$ is the *posterior* probability of H given D, and $p(D|H)$ 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 $l(E) = -\log_2\big(p(E)\big)$ 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) 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 transmitting the data raw. If a stated hypothesis takes longer than what is required by a null-model, then clearly such an hypothesis is unacceptable. A more complex hypothesis fits the data better than a simpler model, in general. We see that MML encoding gives a trade-off between hypothesis complexity ($l(H)$), and its goodness of fit to the data ($l(D|H)$). Therefore, MML criterion formally justifies and realises Occam's razor.

An important aspect of MML framework is that it is tolerant to measurement accuracy and noise in the underlying data. For a justification of this and a comprehensive study of the principle of MML, refer (Wallace, 2005).

3 FORMULATING THE PROBLEM USING MINIMUM MESSAGE LENGTH

A protein $\mathcal{P}=\{P_1,\cdots,P_n\}$ is a sequence[2] of n three-dimensional points corresponding to the coordinates (in $\mathbb{R}^3$) of C_α atoms along the protein backbone, from its N- to C- terminus.[3]

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

Given some subsequence $\mathcal{Q}$ of sequence of points $\mathcal{P}$, 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 $\mathcal{Q}$. This is because any pair of consecutive points, $Q_{r'} \equiv P_{i'_r}$ and $Q_{r'+1} \equiv P_{i_{r'+1}}$, form endpoints for abstracting the points between $P_{i_{r'}}$ and $P_{i_{r'+1}}$ (inclusive) in the protein with a line segment. Note that a subsequence $\mathcal{Q}$ with k endpoints yields a delineation containing $k-1$ line segments between successive endpoints.

The goal this article is to find the *best* delineation of a given set of coordinate data, where the objective 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 $\mathcal{P}$ with a hypothesis $\mathcal{Q}$ 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 $\mathcal{P}$ from the received message to a reasonable level of precision using the general rules they have agreed upon as a part of the code book.

[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 $\mathcal{P}$ 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.)

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 $\mathcal{Q}$ which denotes the delineation of $\mathcal{P}$. This is equivalent to transmitting the hypothesis $\mathcal{Q}$ in $l(\mathcal{Q})$ bits.

(2) The second part will contain the remainder of points in $\mathcal{P}$ (that is, $\mathcal{P}-\mathcal{Q}$) that weren't sent in the first part. In other words, these are the points in $\mathcal{P}$ that are between the endpoints stated in $\mathcal{Q}$. 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 $\mathcal{P}$ given the hypothesis $\mathcal{Q}$ over $l(\mathcal{P}|\mathcal{Q})$ bits.

Therefore, as a part of the codebook, the transmitter and receiver must have agreed upon the encoding of the endpoints in $\mathcal{Q}$ and the encoding of deviations of points $\mathcal{P}-\mathcal{Q}$ explained by line segments between successive endpoints in $\mathcal{Q}$. 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 $\mathcal{Q}$ 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, $(x_{\min},y_{\min},z_{\min})$ and $(x_{\max},y_{\max},z_{\max})$ 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 $\log(x_{\max}-x_{\min})+\log(y_{\max}-y_{\min})+\log(z_{\max}-z_{\min})=\log\big((x_{\max}-x_{\min})(y_{\max}-y_{\min})(z_{\max}-z_{\min})\big)=\log V$ bits, where V is the volume of the bounding rectangular box. It follows from here that all the k endpoints in $\mathcal{Q}$ can be stated in $k\log V$ bits.[5]

Therefore, the message length to state the first part of the transmission requires $\log^* k + k\log V$ bits.

[4]$\log^* x = \log x + \log\log x + \cdots$ (over all positive terms)

[5]Note that the coordinates of the bounding rectangular box is a constant given the data, so it can be ignored at least for the purposes of comparing two hypotheses.

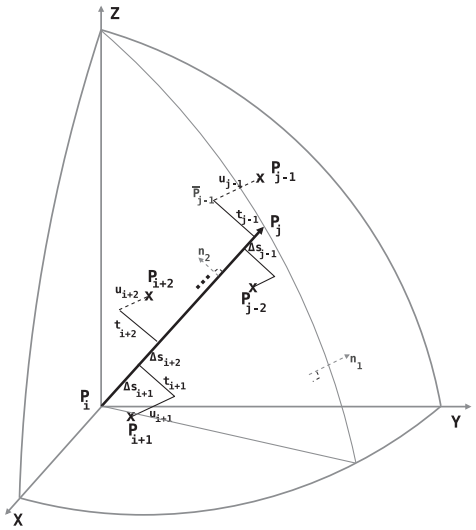

Fig. 1. Deviations $\Delta s, t$ and u of intermediate points $P_{i+1} \cdots P_{j-1}$ to a line segment between two endpoints P_i and P_j. (Refer main text.)

4.2 Encoding the second part of the message

In the second part, the transmitter has to encode the data, $\mathcal{P}-\mathcal{Q}$, between endpoints stated in the first part of the message. For a successive pair of endpoints $\left(Q_{r'} \equiv P_i, Q_{r'+1} \equiv P_j\right), 1 \leq i < j \leq n, 1 \leq r' \leq k$ in $\mathcal{Q}$, there are $j-i-1$ intermediate points between P_i and P_j in $\mathcal{P}$. 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 $\mathcal{Q}$ 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 $\mathcal{Q}$, $Q_{r'} \equiv P_i$ and $Q_{r'+1} \equiv P_j$. This line will be used to explain the intermediate points $P_{i+1} \cdots P_{j-1} \in \mathcal{P}$. For any intermediate point P_r, $i+1 \leq r \leq j-1$, define three spatial deviations $\Delta s_r, t_r$ and u_r. In the reverse order, u_r is the signed distance of P_r to the plane defined by vectors $P_j - P_i$ and z-axis. To define t_r, first project P_r to the plane defined above. Call this projection point $\overline{P}_r$. Given this projection, t_r is the signed perpendicular distance of $\overline{P}_r$ to the line L_{ij}. Finally, the deviation Δs_r is the (unsigned) lateral distance along the line L_{ij} between points of projection of $\overline{P}_{r-1}$ and $\overline{P}_r$ onto the line (Fig. 1). (Refer the supplementary note containing a discussion on these deviations under arbitrary rotation of the coordinates.) Note that once the endpoints P_i and P_j are specified, and given the sets of spatial deviations Δs_r's, t_r's and u_r's for the intermediate points $P_r, \forall i < r < j$, the receiver can entirely recover the coordinates of all intermediate points.

In this work, we assume the three spatial deviations Δs's, t's and u's of the intermediate points to be independent and normally distributed. Individual variables of each distribution are considered independent and random. (See supplementary note for a discussion on these assumptions.) Given these assumptions we have three distributions of the form: $\Delta s \sim \mathcal{N}(\mu_{\Delta s}, \sigma^2_{\Delta s})$, $t \sim \mathcal{N}(\mu_t, \sigma^2_t)$, and $u \sim \mathcal{N}(\mu_u, \sigma^2_u)$, where μ and σ^2 are the *mean* and *variance* of the respective normal distributions. For the structural coordinate data, we assume that the mean of the distributions of t's and u's is zero: $t \sim \mathcal{N}(0, \sigma^2_t)$, and $u \sim \mathcal{N}(0, \sigma^2_u)$. Therefore, to communicate the three distribution, the transmitter has to state the following four parameters: $\left(\mu_{\Delta s}, \sigma^2_{\Delta s}, \sigma^2_t, \sigma^2_u\right)$.

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: $\left(\sigma^2_{\Delta s}, \sigma^2_t, \sigma^2_u\right)$.

We will now compute the code lengths to state the variance of three normal distributions. Variance for a Gaussian distribution is simply 'mean squared minus squared mean':

$$\sigma^2_{\Delta s} = \frac{\sum_{r=i+1}^{j-1} (\Delta s_r - \mu_{\Delta s})^2}{m_{ij}} = \frac{\sum_{r=i+1}^{j-1} \Delta s_r^2}{m_{ij}} - {\mu_{\Delta s}}^2$$

Similarly, we have $\sigma^2_t = \frac{\sum_{r=i+1}^{j-1} t_r^2}{m_{ij}}$ and $\sigma^2_u = \frac{\sum_{r=i+1}^{j-1} u_r^2}{m_{ij}}$, since $\mu_t = \mu_u = 0$. 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 Δs's. Since we have assumed that the distribution is a Gaussian, the probability distribution of a random variable Δs_r with parameters $\mu_{\Delta s}$ and ${\sigma_{\Delta s}}^2$ is given by:

$$\Delta s_r \sim \mathcal{N}(\mu_{\Delta s}, \sigma_{\Delta s}) = \frac{1}{\sqrt{2\pi}\sigma_{\Delta s}} e^{-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma^2_{\Delta s}}}$$

Since we assumed that variables are independent, we have

$$p\left(\Delta s_{i+1}, \ldots, \Delta s_{j-1} | \mathcal{N}(\mu_{\Delta s}, \sigma^2_{\Delta s})\right) = \prod_{r=i+1}^{j-1} \frac{1}{\sqrt{2\pi}\sigma_{\Delta s}} e^{-\frac{(\Delta s_r - \mu_{\Delta s})^2}{2\sigma^2_{\Delta s}}}$$

This implies,

$$p\left(\Delta s_{i+1},\ldots,\Delta s_{j-1}|\mathcal{N}(\mu_{\Delta s},\sigma^2_{\Delta s})\right)=\left(\frac{1}{\sqrt{2\pi}\sigma_{\Delta s}}\right)^{m_{ij}} e^{\frac{-m_{ij}}{2}}.$$

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

Following a similar expansion, we can show that the code lengths for the deviation t_r's and u_r's are $\frac{m_{ij}}{2}\log\left(2\pi e\sigma_t^2\right)$ and $\frac{m_{ij}}{2}\log\left(2\pi e\sigma_u^2\right)$, 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 $(P_{i_r},P_{j_r}), 1\le r<k$. (In practice, for a delineation, P_{i_r} of the r-th line segment is equivalent to $P_{j_{r-1}}$ of $(r-1)$th line segment.) Then the total code length of the second part is the sum of the following terms:

(1) $\sum_{r=1}^{k-1}\log^* m^r_{ij}$, where $m^r_{ij}=j_r-i_r-1$, representing the total code length to encode the number of intermediate points described by all line segments in the delineation put together.

(2) $\sum_{r=1}^{k-1}\frac{3}{2}\log m^r_{ij}$ bits to encode the parameters (three per line segment) corresponding to the distribution of spatial deviations for all lines.

(3) $\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_{\Delta s}}^2\right)$ bits to encode Δs_r's over all line segments

(4) $\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_t}^2\right)$ bits to encode t_r's over all line segments

(5) $\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_u}^2\right)$ bits to encode u_r's over all line segments

4.3 Problem statement

Given a delineation $\mathcal{Q}$ (hypothesis) of coordinates $\mathcal{P}$ (data), denote the total message length required to explain the data by the hypothesis as $\mathcal{H}(\mathcal{Q})$. Combining the code lengths to state the two part message described in Sections 4.1 and 4.2, the total message length is:

$$\begin{aligned}\mathcal{H}(\mathcal{Q})=\log^* k+k\log V+\sum_{r=1}^{k-1}\log^* m^r_{ij}+\sum_{r=1}^{k-1}\frac{3}{2}\log m^r_{ij}\\ +\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_{\Delta s}}^2\right)+\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_t}^2\right)\\ +\sum_{r=1}^{k-1}\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_u}^2\right)\end{aligned}\quad(1)$$

Since $\log^* k\ll k\log V$, the transmitter can ignore stating that term in the code length. Assume

$$\begin{aligned}\mathcal{H}^r_{ij}=\log V+\log^* m^r_{ij}+\frac{3}{2}\log m^r_{ij}+\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_{\Delta s}}^2\right)\\ +\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_t}^2\right)+\frac{m^r_{ij}}{2}\log\left(2\pi e{\sigma^r_u}^2\right)\end{aligned}\quad(2)$$

$\mathcal{H}^r_{ij}$ denotes the component code length to express each line segment L^r_{ij} with endpoints P_{i^r} and P_{j^r}, given a delineation $\mathcal{Q}$. This implies $\mathcal{H}(\mathcal{Q})=\sum_{r=1}^{k-1}\mathcal{H}^r_{ij}$

This allows us to formally define the delineation problem as follows:

The problem:

Given $\mathcal{P}$ containing a sequence of n points, find a subsequence $\mathcal{Q}\in\mathcal{P}$ containing $k\le n$ points such that the total message length to explain $\mathcal{P}$ with $\mathcal{Q}$, $\mathcal{H}(\mathcal{Q})=\sum_{r=1}^{k-1}\mathcal{H}^r_{ij}$, is globally minimum.

5 FINDING THE OPTIMAL DELINEATION

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

Potentially every pair of points P_i and P_j, $1\le i<j\le n$ can be a part of the delineation in $\mathcal{Q}^*$. (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 $\mathcal{H}=(\mathcal{H}_{ij})_{1\le i<j\le n}$ of code lengths for all possible pairs of points in $\mathcal{P}$.

Then, the matrix $\mathcal{H}$ will be used to find a subsequence of points $\mathcal{Q}^*$ such that the total code length $\mathcal{H}(\mathcal{Q}^*)$ 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 $\mathcal{P}$, there are ${}^nC_2=\frac{n\times(n-1)}{2}$ 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 $\mathcal{P}$. Next, for each line segment, there are three 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 relatively complex part is to compute the code lengths of the spatial deviations of the line, Δs's t's and u's. Each of these three deviations have code lengths that depend on their respective variance, $\sigma^2_{\Delta s},\sigma^2_t$ and σ^2_u. 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 näive 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 $\mathcal{H}$.

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 $\mathcal{H}$ of code length between all possible endpoints to find the optimal delineation $\mathcal{Q}^*$ that minimizes $\mathcal{H}(\mathcal{Q}^*)$ using a one-dimensional dynamic program.

Let C_i be an array that stores the optimal code length of delineating points $P_1,\ldots,P_i$, $\forall 1 \le i \le n$. 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:

$$C_1 = 0,$$
$$C_j = \min_{i=1}^{j-1} \left\{ \mathcal{H}_{1j}, (C_i + \mathcal{H}_{ij}) \right\}, \forall 1 \le j \le n$$

In other words, the optimal code length to delineate the points $P_1,\ldots,P_j$ $(1 \le j \le n)$ 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 $\mathcal{H}_{ij}$ is minimum, over all $1 \le i < j$.

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 $\mathcal{Q}^*$ of $\mathcal{P}$, where $\mathcal{H}(\mathcal{Q}^*) \equiv C_n$ 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_j 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 $\mathcal{Q}^*$.

6 EFFICIENT COMPUTATION OF MATRIX $\mathcal{H}$

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

For the matrix $\mathcal{H}$ to be computable in $O(n^2)$ operations, each element $\mathcal{H}_{ij}$ in the matrix should be computable in constant time. However terms ${\sigma^r_{\Delta s}}^2$, ${\sigma^r_t}^2$, and ${\sigma^r_u}^2$ in Equation (2) cannot be computed in constant time. For a line segment L_{ij}, näively, 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 $\mathcal{H}$. Below we will show that each of ${\sigma^r_{\Delta s}}^2$, ${\sigma^r_t}^2$, and ${\sigma^r_u}^2$ 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 $\sigma^2_{\Delta s}$'s

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

By the definitions of the spatial deviations in Section 4.2, any $\Delta s_r, 1 \le i < r < j \le n$ is the scalar associated with the projection of the vector $(P_r - P_{r-1})$ onto the vector $L_{ij} \equiv (P_j - P_i)$. (Refer Fig. 1.) Let $\hat{L}_{ij} = \left\langle \hat{L}^x_{ij}, \hat{L}^y_{ij}, \hat{L}^z_{ij} \right\rangle$ represent the direction cosines of the vector L_{ij}, where $\hat{L}^x_{ij} = \frac{(x_j - x_i)}{||L_{ij}||}$, $\hat{L}^y_{ij} = \frac{(y_j - y_i)}{||L_{ij}||}$ and $\hat{L}^z_{ij} = \frac{(z_j - z_i)}{||L_{ij}||}$. Then Δs_r is the dot product of $(P_r - P_{r-1})$ and $\hat{L}_{ij}$: $\Delta s_r = (P_r - P_{r-1}) \cdot \hat{L}_{ij}$. Expanding this we get,

$$\Delta s_r = (x_r - x_{r-1})\hat{L}^x_{ij} + (y_r - y_{r-1})\hat{L}^y_{ij} + (z_r - z_{r-1})\hat{L}^z_{ij}$$

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

$S_{ij} = \sum_{r=i+1}^{j-1} \left((x_r - x_{r-1})\hat{L}^x_{ij} + (y_r - y_{r-1})\hat{L}^y_{ij} + (z_r - z_{r-1})\hat{L}^z_{ij} \right)^2$ Expanding S_{ij},

$$\begin{aligned} S_{ij} = {} & \hat{L}^{x\,2}_{ij} \sum_{r=i+1}^{j-1} (x_r - x_{r-1})^2 + \hat{L}^{y\,2}_{ij} \sum_{r=i+1}^{j-1} (y_r - y_{r-1})^2 \\ & + \hat{L}^{z\,2}_{ij} \sum_{r=i+1}^{j-1} (z_r - z_{r-1})^2 \\ & + 2\hat{L}^x_{ij}\hat{L}^y_{ij} \sum_{r=i+1}^{j-1} (x_r - x_{r-1})(y_r - y_{r-1}) \\ & + 2\hat{L}^y_{ij}\hat{L}^z_{ij} \sum_{r=i+1}^{j-1} (y_r - y_{r-1})(z_r - z_{r-1}) \\ & + 2\hat{L}^x_{ij}\hat{L}^z_{ij} \sum_{r=i+1}^{j-1} (x_r - x_{r-1})(z_r - z_{r-1}) \end{aligned} \quad (3)$$

Now, let $S^{xx}_{ij}, S^{yy}_{ij}, S^{zz}_{ij}, S^{xy}_{ij}, S^{yz}_{ij}, S^{xz}_{ij}$ be a set of variables which we will call here *sufficient statistics*. These variables are of the form: $S^{AB}_{ij} = \sum_{r=i+1}^{j-1} (A_r - A_{r-1})(B_r - B_{r-1})$, where A and B take the values $\{x,y,z\}$.

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

$$\begin{aligned} S_{ij} = {} & \hat{L}^{x\,2}_{ij} S^{xx}_{ij} + \hat{L}^{y\,2}_{ij} S^{yy}_{ij} + \hat{L}^{z\,2}_{ij} S^{zz}_{ij} + 2\hat{L}^x_{ij}\hat{L}^y_{ij} S^{xy}_{ij} \\ & + 2\hat{L}^y_{ij}\hat{L}^z_{ij} S^{yz}_{ij} + 2\hat{L}^x_{ij}\hat{L}^z_{ij} S^{xz}_{ij} \end{aligned} \quad (4)$$

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 $S^{AB}_{ij+1} = S^{AB}_{ij} + (A_j - A_{j-1})(B_j - B_{j-1})$, where $\{A,B\} \in \{x,y,z\}$.

Therefore, using the sufficient statistics the computation of $\sigma_{\Delta s}$ for a line segment can be computed incrementally in constant time.

6.2 Constant-time update of σ_t^2's

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

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:

$$\left\langle -(x_j - x_i)(z_j - z_i), -(y_j - y_i)(z_j - z_i), (x_j - x_i)^2 + (y_j - y_i)^2 \right\rangle$$

Let $\hat{n}_2 = \langle \hat{n}_2^x, \hat{n}_2^y, \hat{n}_2^z \rangle$ represent the direction cosines of $\vec{n}_2$, where $\hat{n}_2^x = \frac{-(x_j - x_i)(z_j - z_i)}{||\vec{n}_2||}$, $\hat{n}_2^y = \frac{-(y_j - y_i)(z_j - z_i)}{||\vec{n}_2||}$ and $\hat{n}_2^z = \frac{(x_j - x_i)^2 + (y_j - y_i)^2}{||\vec{n}_2||}$.

Then $t_r = (P_r - P_i) \cdot \hat{n}_2$. (Refer Fig. 1.) This implies

$$\Delta t_r = (x_r - x_i)\hat{n}_2^x + (y_r - y_i)\hat{n}_2^y + (z_r - z_i)\hat{n}_2^z$$

Assume $T_{ij} = \sum_{r=i+1}^{j-1} t_r^2$ and expanding along the steps we took in the previous section, we get

$$\begin{aligned} T_{ij} &= \hat{n}_2^{x\,2} T_{ij}^{xx} + \hat{n}_2^{y\,2} T_{ij}^{yy} + \hat{n}_2^{z\,2} T_{ij}^{zz} + 2\hat{n}_2^x \hat{n}_2^y T_{ij}^{xy} \\ &\quad + 2\hat{n}_2^y \hat{n}_2^z T_{ij}^{yz} + 2\hat{n}_2^x \hat{n}_2^z T_{ij}^{xz} \end{aligned} \tag{5}$$

where computation of any T_{ij+1} can be updated from T_{ij} in constant time.

6.3 Constant-time update of σ_u^2's

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

Then $u_r = (P_r - P_i) \cdot \hat{n}_1$. (Refer Fig 1.) Expanding as before we get

$$U_{ij} = \hat{n}_1^{x\,2} U_{ij}^{xx} + \hat{n}_1^{y\,2} U_{ij}^{yy} + 2\hat{n}_1^x \hat{n}_1^y U_{ij}^{xy} \tag{6}$$

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 $\mathcal{H}$ 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 (in C++). Our implementation is available from http://www.csse.monash.edu.au/~karun/pmml/.

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 >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.

Figure 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

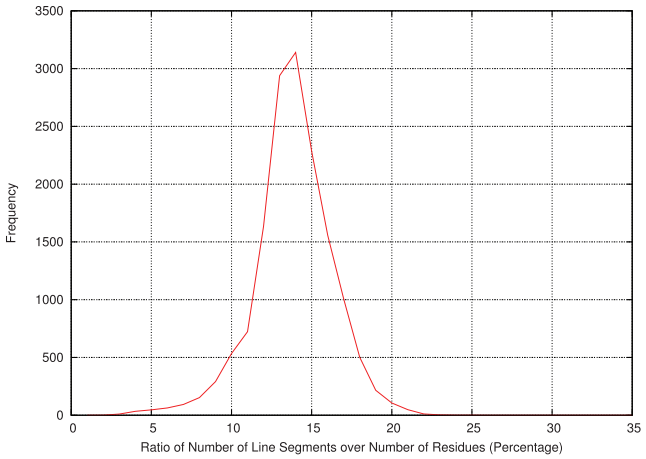

Fig. 2. Distribution of ratios of number of line segments over number of residues per structure in the dataset. Ratios are expressed in percentages and rounded to the nearest integral value.

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.

It is of considerable interest to evaluate the agreement of standard secondary structural elements—helices and strands of sheets—with the delineation identified by the program. We note that an ideal delineation of a structure must encompass these elements since they are ideal candidates for approximation with lines or vectors given the linear spatial trend in their geometry. In order to evaluate the agreement, we coarsely classify each segment to one of three secondary structure states: 'Helix', 'Strand' and 'Other'. This three-state classification is based on certain geometric characteristics of the segments in the delineation. Specifically, we compute the following geometric profiles for each segment: 'rise', 'pitch' and backbone dihedral angles ϕ and ψ. The *rise* (ρ) of the segment with endpoints P_i and P_j is $\rho = D_{ij}/(j - i + 1)$, where D_{ij} is the Euclidean distance between the endpoints. In other words, the rise gives the average translation of points along the line between endpoints. The rise of a standard secondary structure is directly related to the pitch (p) of the segment. For a substructure with a geometry that repeats itself every n residues, the relationship between rise and pitch is given by $p = n\rho$. Table 1 summarizes the geometric profiles of ideal secondary structures (Taylor, 2001). Inspecting these profiles per segment, a coarse characterisation for each segment in the delineation is achieved.

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

Table 1. Geometric profiles of ideal secondary structures used to classify coarsely the delineation identified by the program. ϕ and ψ are average backbone dihedral angles. n is the periodicity of the local structure. ρ is the rise. p is the pitch

Type	ϕ	ψ	n	ρ	p
3_{10}-Helix	−57.1	−69.7	3.0	2.0	6.0
α-Helix	−57.8	−47.0	3.6	1.5	5.5
π-Helix	−74.0	−4.0	4.4	1.1	5.0
β-Strand	−139.0	135.0	2.0	3.4	6.8

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

Comparison	Helices (%)	Strands (%)
PMML (coarse)_vs_DSSP	79.0	83.3
PMML (coarse)_vs_STRIDE	79.3	83.1
PMML (refine)_vs_DSSP	92.6	92.4
PMML (refine)_vs_STRIDE	91.3	92.1
STRIDE_vs_DSSP	95.7	96.9

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 segment level assignment by our method produced a satisfactory concordance with DSSP and STRIDE, there is still a disagreement of ∼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 stand 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.

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

[6]We do not distinguish between the three types of helices and treat them as one state.

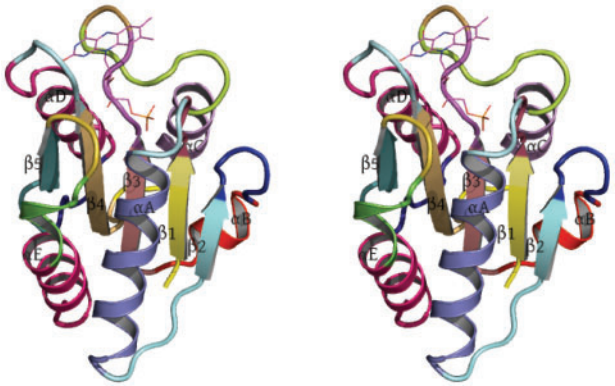

Fig. 3. Wall-eye stereo image of 1.8 Å crystal structure of oxidized *Clostridium beijerinckii* flavodoxin. Each delineated segment produced by PMML is shown in a different color. The elements of secondary structures, of helices and strands of sheet, were derived from the wwPDB file, 5NLL, and are shown 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.

Table 3. The residue ranges of secondary structural elements (SSEs) in the structure of flavodoxin shown in Fig. 3

SSE	wwPDB	PMML
β_1	Lys2-Trp6	Met1-Tyr5
α_A	Asn11-Glu25	Asn11-Glu27
β_2	Asn31–Asn34	Gly27-Ile33
α_B	Ile40-Asn45	Asn39-Glu46
β_3	Ile48–Cys53	Asp47-Cys53
α_C	Phe66-Lys76	Glu65-Thr75
β_4	Lys81–Tyr88	Gly79-Ser87
α_D	Lys94-Gly105	Gly91-Gly107
β_5	Leu115–Gln118	Glu112-Gln118
α_E	Asp122-Ile136	Glu120-Gln126,Gln126-Ile136

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.

of secondary structure elements as ascertained by an expert. To highlight these differences consider the following example of the delineation produced by PMML. Figure 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 ($\beta_1,\ldots,\beta_5$), with two helices packed against each face of the sheet (α_A,α_E and α_C,α_D). There is also a short helix (α_B) located near the N-terminus of the protein. (Fig. 3.) Different segments produced by PMML are shown in different colors. 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. 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. The segmentation in the regions around the SSEs α_E, β_2 and β_5 show some discrepancies. The residue range from wwPDB corresponding to α_E was approximated by PMML using 2 segments instead of one. The first segment is composed of roughly one turn of the helix at α_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 β_2, the start location identified by PMML precedes the start location identfied in the wwPDB file by four 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 β_1), so the β_2 strand may well start at residue Lys28 or Asp29. Similarly, for β_5, the start location of the segment from PMML was identified to be three residues before the location identified in the wwPDB file, and inspecting the structure, we note the $\beta-$bulge in strand β_5, and hydrogen bonds between atoms 80O$\cdots$109N and 82N$\cdots$109O; assignment of the start of the strand β_5 to residue 109 is not indefensible.

8 CONCLUSION

We have presented a novel and efficient method to delineate protein structures using the 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/.

ACKNOWLEDGEMENTS

We thank the anonymous referees for comments that improved the manuscript. L.A. and A.S.K. thank Nathan Hurst for useful pointers during the development of this work.

Funding: ASK's research is supported by Monash University's Talent Enhancement and Larkins Fellowship. NICTA is funded by the Australian Government as represented by the Department of Broadband, Communications and the Digital Economy and the Australian Research Council.

Conflict of Interest: none declared.

REFERENCES

Abagyan,R.A. and Maiorov,V.N. (1988) A simple qualitative representation of polypeptide chain folds: comparison of protein tertiary structures. *J. Biomol. Struct. Dyn.*, **5**, 1267–1279.

Banerjee,S. *et al*. (1996) A minimum description length polygonal approximation method. *IBM Tech. Rep.*, **RJ 10007**, 1–19.

Bellman,R. (1957) *Dynamic Programming*. Princeton University Press, Princeton, New Jersey.

Berman,H.M. *et al*. (2002) The protein data bank. *Acta Crystallogr. D Biol. Crystallogr.*, **58**(Pt 6 No 1), 899–907.

Chothia,C. *et al*. (1981) Helix to helix packing in proteins. *Proc. Natl Acad. Sci. USA*, **78**, 4146–4150.

Colloc'h,N. *et al*. (1993) Comparison of three algorithms for the assignment of secondary structure in proteins. *Protein Eng.*, **6**, 377–382.

Cuff,J.A. and Barton,G.J. (1999) Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins*, **34**, 508–519.

Dupuis,F. *et al*. (2004) Protein secondary structure assignment through Voronoi tessellation. *Proteins*, **55**, 519–528.

Edsall,J.T. *et al*. (1966) A proposal of standard conventions and nomenclature for the description of polypeptide conformations. *J. Mol. Biol.*, **15**, 399–407.

Elias,P. (1975) Universal codeword sets and representations of the integers. *IEEE Trans. Inf. Theory*, **21**, 194–203.

Frishman,D. and Argos,P. (1995) Knowledge-based protein secondary structure assignment. *Proteins*, **23**, 566–579.

Kabsch,W. and Sander,C. (1983) Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, **22**, 2577–2637.

Kamat,A.P. and Lesk,A.M. (2007) Contact patterns between helices and strands of sheet define protein folding patterns. *Proteins: Structure, Function, and Bioinformatics*, **66**, 869–876.

Konagurthu,A.S. and Lesk,A.M. (2010) Concise tableau representation of protein folding patterns. *J. Mol. Recogn.*, **23**, 253–257.

Konagurthu,A.S. *et al*. (2008) Structural search and retreival using tableau representation of protein folding patterns. *Bioinformatics*, **24**, 645–651.

Labesse,G. *et al*. (1997) P-SEA: a new efficient assignment of secondary structure from c alpha trace of proteins. *Comput. Appl. Bio. Sci.*, **13**, 291–295.

Lesk,A.M. and Chothia,C. (1980) How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins. *J. Mol. Biol.*, **136**, 225–230.

Lesk,A.M. (1995) Systematic representation of protein folding patterns. *J. Mol. Graphics*, **13**, 159–164.

Levitt,M. and Greer,J. (1977) Automatic identification of secondary structure in globular proteins. *J. Mol. Biol.*, **114**, 181–239.

Majumdar,I. *et al*. (2005) PALSSE: A program to delineate linear secondary structural elements from protein structures. *BMC Bioinformatics*, **6**, 202.

Mizuguchi,K. and Go,N. (1995) Comparison of spatial arrangements of secondary structural elements in proteins. *Protein Eng.*, **8**, 353–362.

Richardson,J.S. (1981) The anatomy and taxonomy of protein structure. *Adv. Protein Chem.*, **34**, 167–339.

Richards,F.M. and Kundrot,C.E. (1988) Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. *Proteins*, **3**, 71–78.

Shannon,C.E. (1948) A mathematical theory of communication. *Bell Syst. Tech. Jrnl.*, **27**, 379–423.

Shi,S. *et al*. (2007) Searching for three-dimensional secondary structural patterns in proteins with ProSMoS. *Bioinformatics*, **23**, 1331–1338.

Sklenar,H. *et al*. (1989) Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. *Proteins*, **6**, 46–60.

Srinivasan,R. and Rose,G.D. (1999) A physical basis for protein secondary structure. *Proc. Natl Acad. Sci. USA*, **96**, 14258–14263.

Taylor,W.R. *et al*. (1983) A elipsoidal approximation of protein shape. *J. Mol. Graphics*, **1**, 30–38.

Taylor,W.R. (2001) Defining linear segments in protein structures. *J. Mol. Biol.*, **310**, 1135–1150.

Wallace,C.S. and Boulton,D.M. (1968) An information measure for classification. *Comp. J.*, **11**, 185–194.

Wallace,C.S. (2005) *Statistical and Inductive Inference using Minimum Message Length*. Information Science and Statistics. Springer, New York.

Wang,G. and Dunbrack,R. L. Jr. (2003) PISCES: a protein sequence culling server. *Bioinformatics*, **19**, 1589–1591.