With the rapid growth in the corpus of known structures, concise representations are increasingly preferred to inspect and analyze protein fold patterns (Abagyan and Maiorov, 1988; Lesk, 1995). Among the standard secondary structure definitions are: helix ($\alpha$-helix, $\beta$-helix and $\beta_1$-helix) and strand of sheet (Eadsall et al., 1966). Ideally, the spatial trace of $\alpha$-Carbon ($C_{\alpha}$) atoms of standard secondary structure shows 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 $C_{\alpha}$ 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 $C_{\alpha}$ 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 interaction of these elements (Kamat and Lesk, 2007; Konagurthu and Lesk, 2010; Lesk and Chothia, 1980; Lesk, 1995).
This method, our results show, the line segments generated by delineating secondary structures. However, as expected from such and thresholds. We emphasize that our method is not a method for the existing methods, our method is completely free of parameters piecewise linear approximation of the structure. In a stark contrast to contributing to this minimum message length result in the best (message) length of the given coordinate data. The code lengths of bits (or code length). Using these code lengths, a globally structure. Each such explanation is encoded in a certain number line segment the points between any pair of residues in the method uses an information-theoretic approach to explain as a (message) length of the given coordinate data. The code lengths

Mathematical Theory of Communication' (Shannon, 1948) that, given an event \( E \) with a probability \( p(E) \), the message length, \( \log_2(p(E)) \) for an optimal code is given by \( \log_2(p(E)) \). Carrying this insight to the Bayes’s theorem, we get the following relationship between conditional probabilities in terms of optimal message lengths.

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 \( H \)) 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.

1Banerjee 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.
A protein MML, refer (Wallace, 2005).

Formulating the problem using minimum message length

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

Define a piecewise linear approximation of $P$ as a subsequence of $k \leq n$ points from $P$ of the form $Q = \{Q_1 \equiv P_{i_1}, \ldots, Q_n \equiv P_{i_n}\}$ such that $1 = i_1 < i_2 < \cdots < i_n = n$, 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 $Q_n = P_n$).

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, $Q_{ij}, 1 \leq i < j \leq k$. 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, $Q_i \equiv P_{i_j}$ and $Q_{i+1} \equiv P_{i_{j+1}}$, form endpoints for abstracting the points between $P_{i_j}$ and $P_{i_{j+1}}$ (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 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 $P$ 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 received 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 $l(P\setminus Q)$ 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.

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.\(^5\)

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 ($\AA$) 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, $(\min_{x}, \min_{y}, \min_{z})$ and $(\max_{x}, \max_{y}, \max_{z})$ over all possible values of $x, y, z$ in the given data. Once this bounding box is specified, any $(x, y, z)$ coordinates within the box can be coded in $\log(\max_x - \min_x) + \log(\max_y - \min_y) + \log(\max_z - \min_z) = \log V$ bits, 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.\(^6\)

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

Piecewise linear approximation of structures using MML

\(^2\)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 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.)

\(^4\)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.
4.2 Encoding the second part of the message

In the second part, the transmitter has to encode the data, \( P = Q \), between endpoints stated in the first part of the message. For a successive pair of endpoints \( Q_i = P_i, Q_{i+1} = P_{j+1} \), \( 1 \leq i < j \leq n \), 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 \( Q \). \( Q_i = P_i \) and \( Q_{i+1} = P_{j+1} \). This line will be used to explain 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 \( \Delta u_r, \Delta t_r, \Delta a_r \).

In the reverse order, \( u_1 \) is the signed distance of \( P_i \) to the plane defined by vectors \( P_{j-1} - P_{j-1} \) and \( z \)-axis. To define \( t_1 \), first project \( P_i \) to the plane defined above. Call this projection point \( P_{ij} \). Given this projection, \( t_1 \) is the signed perpendicular distance of \( P_{ij} \) to the line \( L_{ij} \). Finally, the deviation \( a_1 \) is the (unsigned) lateral distance along the line \( L_{ij} \) between points of projection of \( P_{i+1} \cdots P_{j-1} \) 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 \( \Delta u_i \)'s, \( \Delta t_i \)'s and \( \Delta a_i \)'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 \( \Delta u_i \)'s, \( \Delta t_i \)'s and \( \Delta a_i \)'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: \( \Delta u \sim N(\mu_u, \sigma_u^2) \), \( \Delta t \sim N(\mu_t, \sigma_t^2) \), and \( u \sim N(\mu_u, \sigma_u^2) \), where \( \mu \) and \( \sigma^2 \) are the mean and variance of the respective normal distributions. For the spatial coordinate data, we assume that the mean of the distributions of \( \Delta u \)'s and \( \Delta a \)'s is zero: \( \Delta u \sim N(0, \sigma_u^2) \), and \( u \sim N(0, \sigma_u^2) \). Therefore, to communicate the three distribution, the transmitter has to state the following four parameters: \( \mu_u, \sigma_u^2, \sigma_t^2, \sigma_a^2 \).

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

\[
\mu_{\Delta u} = \frac{1}{m_{ij}} \sum_{r=i+1}^{j-1} \Delta u_r
\]

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

Similarly, we have \( \sigma_{\Delta t}^2 = \frac{1}{m_{ij}} \sum_{r=i+1}^{j-1} (\Delta t_r - \mu_{\Delta t})^2 \) and \( \sigma_{\Delta a}^2 = \frac{1}{m_{ij}} \sum_{r=i+1}^{j-1} (\Delta a_r - \mu_{\Delta a})^2 \), since \( \mu_a = 0 \).

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

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

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

\[
\sigma_{\Delta a}^2 = \frac{1}{m_{ij}} \sum_{r=i+1}^{j-1} (\Delta a_r - \mu_{\Delta a})^2
\]
This implies,

\[ p\left(Δ_{i+1}, \ldots, Δ_{j-1} | N(μ, σ_1^2) \right) = \left(\frac{1}{\sqrt{2πσ_1}}\right)^{m_{ij}} e^{-\frac{1}{2} \left(\frac{m_{ij}}{σ_1^2}\right)} \]

Therefore, using Shannon’s insight, the optimal code length to describe the entire sets of individual deviations of Δ’s for a line \(L_i\) will require \(-log_2 p\left(Δ_{i+1}, \ldots, Δ_{j-1} | N(μ, σ_1^2) \right)\)

\[ = -log\left(\frac{1}{\sqrt{2πσ_1}}\right)^{m_{ij}} e^{-\frac{1}{2} \left(\frac{m_{ij}}{σ_1^2}\right)} = \frac{m_{ij}}{2} log(2πeσ_1^2) \text{ bits.} \]

Following a similar expansion, we can show that the code lengths for the deviation \(t_i\)’s and \(u_i\)’s are \(\frac{n_r}{2} log(2πeσ_1^2)\) and \(\frac{n_u}{2} log(2πeσ_2^2)\), respectively.

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

\[
\begin{align*}
(1) & \sum_{i=1}^{k-1} \log^* m_{i_j} \quad \text{where } m_{i_j} = j_i - i_j - 1, \text{ representing the total code length to encode the number of intermediate points described by all line segments in the delineation put together.} \\
(2) & \sum_{i=1}^{k-1} \frac{1}{2} \log m_{i_{ij}} \text{ bits to encode the parameters (three per line segment) corresponding to the distribution of spatial deviations for all lines.} \\
(3) & \sum_{i=1}^{k-1} \frac{1}{2} \log(2πeσ_1^2) \text{ bits to encode } Δ_i\text{'s over all line segments} \\
(4) & \sum_{i=1}^{k-1} \frac{1}{2} \log(2πeσ_2^2) \text{ bits to encode } t_i\text{'s over all line segments} \\
(5) & \sum_{i=1}^{k-1} \frac{1}{2} \log(2πeσ_2^2) \text{ bits to encode } u_i\text{'s over all line segments}
\end{align*}
\]

4.3 Problem statement

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

\[
H(Q) = \log^* k + k \log V + \sum_{r=1}^{k-1} \log^* m_{i_j} + \sum_{r=1}^{k-1} \frac{1}{2} \log m_{i_{ij}}
\]

\[
+ \sum_{r=1}^{k-1} \frac{1}{2} \log(2πeσ_1^2) + \sum_{r=1}^{k-1} \frac{1}{2} \log(2πeσ_2^2)
\]

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

\[
H'_Q = \log V + \log^* m_{i_j} + \frac{3}{2} \log m_{i_{ij}} + \frac{1}{2} \log\left(\frac{2πeσ_1^2}{m_{i_j}}\right)
\]

\[
+ \frac{1}{2} \log\left(\frac{2πeσ_2^2}{m_{i_{ij}}}\right) \tag{2}
\]

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

The problem:

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

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 \(P_i\) and \(P_j\), \(1 ≤ i < j ≤ n\) 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 \(H'(Q)\) containing the code lengths for all possible pairs of points in \(P\).

Then, the matrix \(H\) will be used to find a subsequence of points \(Q^*\) such that the total code length \(H(Q^*)\) 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 \(H'_Q\) required to describe any line segment \(L_i\) 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 \(\binom{n}{2}\) 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 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, \(Δ_i\)’s \(F\)’s and \(s\)’s. Each of these three deviations have code lengths that depend on their respective variance, \(σ_1^2\), \(σ_2^2\) and \(σ_3^2\). 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 \(H\) 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 \(H\) 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 $P_1, \ldots, P_i$, $1 \leq i \leq n$. 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:

$$C_1 = 0,$$

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

In other words, the optimal code length to delineate the points $P_1, \ldots, P_i$ ($1 \leq i \leq n$) builds on the optimal code length to delineate from $P_1, \ldots, P_j$, if and only if the value of $C_j$ 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 $H(Q^*) = C_n$ is globally minimum. The subsequence of endpoints of this optimal delineation can be computed by storing, for each $i$-th endpoint 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 $\sigma^2$, $\sigma^2_1$, and $\sigma^2_2$ in Equation (2) cannot be computed in constant time. For a line segment $L_{ij}$, naively, these three variances take time proportional to the number of points explained by the line to compute, leading to a $O(n^2)$ algorithm for computing the matrix $H$. Below we will show that each of $\sigma^2$, $\sigma^2_1$, and $\sigma^2_2$ can indeed be computed incrementally in constant time from previous computations resulting in a $O(n^2)$ algorithm.

6.1 Constant-time update of $\sigma^2_\Delta$'s

Consider first these notations: for any vector $\vec{v}$ with direction ratios $(x, y, z)$, let $||\vec{v}|| = \sqrt{x^2+y^2+z^2}$ represents the vector norm of $\vec{v}$. 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 $\Delta_{ij}$, $1 \leq i < j \leq n$ is the scalar associated with the projection of the vector $(P_j - P_{i-1})$ onto the vector $L_{ij} = (P_j - P_i)$. (Refer Fig. 1.) Let $L_{ij} = \{L_{ij}^x, L_{ij}^y, L_{ij}^z\}$ represent the direction cosines of the vector $L_{ij}$, where $L_{ij}^x = \frac{L_{ij}^x}{||L_{ij}||}$, $L_{ij}^y = \frac{L_{ij}^y}{||L_{ij}||}$ and $L_{ij}^z = \frac{L_{ij}^z}{||L_{ij}||}$. Then $\Delta_{ij}$ is the dot product of $(P_j - P_{i-1})$ and $L_{ij}$: $\Delta_{ij} = (P_j - P_{i-1}) \cdot L_{ij}$. Expanding this we get,

$$\Delta_{ij} = (x_i - x_{i-1})L_{ij}^x + (y_i - y_{i-1})L_{ij}^y + (z_i - z_{i-1})L_{ij}^z$$

Denoting $S_j = \sum_{i=1}^{j-1} \Delta_{ij}^2$, $S_j = \sum_{i=1}^{j-1} (x_i - x_{i-1})^2 L_{ij}^x + (y_i - y_{i-1})^2 L_{ij}^y + (z_i - z_{i-1})^2 L_{ij}^z$. Expanding $S_j$,

$$S_j = L_{ij}^x \sum_{i=1}^{j-1} (x_i - x_{i-1})^2 + L_{ij}^y \sum_{i=1}^{j-1} (y_i - y_{i-1})^2 + L_{ij}^z \sum_{i=1}^{j-1} (z_i - z_{i-1})^2$$

$$+ 2L_{ij}^x \sum_{i=1}^{j-1} (x_i - x_{i-1})(y_i - y_{i-1})$$

$$+ 2L_{ij}^y \sum_{i=1}^{j-1} (y_i - y_{i-1})(z_i - z_{i-1})$$

$$+ 2L_{ij}^z \sum_{i=1}^{j-1} (z_i - z_{i-1})(x_i - x_{i-1})$$

(3)

Now, let $S^0_{ij}$, $S^0_{ij}$, $S^0_{ij}$, $S^0_{ij}$, $S^0_{ij}$ be a set of variables which we will call here sufficient statistics. These variables are of the form:

$$S^0_{ij} = \sum_{r=1}^{j-1} (A_i - A_{i-1})(B_j - B_{j-1})$$

Where $A$ and $B$ take the values $(x, y, z)$.

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

$$S_j = L_{ij}^x S^0_{ij} + L_{ij}^y S^0_{ij} + L_{ij}^z S^0_{ij} + 2L_{ij}^x L_{ij}^y S^0_{ij}$$

$$+ 2L_{ij}^x L_{ij}^z S^0_{ij} + 2L_{ij}^y L_{ij}^z S^0_{ij}$$

(4)

From Equation (4) it can be clearly seen that any $S_{ij+1}$ can be updated from $S_j$ in constant time, using the sufficient statistics. This holds because any $S^0_{ij+1} = S^0_{ij} + (A_i - A_{i-1})(B_j - B_{j-1})$, where $(A, B) \in \{x, y, z\}$.

Therefore, using the sufficient statistics the computation of $\sigma^2_\Delta$ for a line segment can be computed incrementally in constant time.
6.2 Constant-time update of $\sigma^2$'s
Let $\hat{n}_1$ be the normal to a plane defined by $\hat{z} \times \hat{e}_j$, where $\hat{z}$ is the unit vector along z-axis with the direction cosines $(0, 0, 1)$. It follows that the direction ratios of $\hat{n}_1$ are $-(y_j - y_i), (x_j - x_i), 0$.

Define $\hat{n}_2$ as a vector which is normal to the plane $\hat{e}_j \times \hat{n}_1$. The direction ratios of $\hat{n}_2$ will be:

$-(x_j - x_i)(y_i - y_j), -(y_j - y_i)(x_j - x_i), (x_j - x_i)^2 + (y_j - y_i)^2$.

Let $\hat{n}_2 = (\hat{n}_2^1, \hat{n}_2^2, \hat{n}_2^3)$ represent the direction cosines of $\hat{n}_2$, where:

$\hat{n}_2^1 = \frac{-(x_j - x_i)(y_i - y_j)}{||\hat{n}_2||}$, $\hat{n}_2^2 = \frac{-(y_j - y_i)(x_j - x_i)}{||\hat{n}_2||}$ and $\hat{n}_2^3 = \frac{(x_j - x_i)^2 + (y_j - y_i)^2}{||\hat{n}_2||}$.

Then $t_j = (P_j - P_i)^T \hat{n}_2$. (Refer Fig. 1.) This implies

$$\Delta_t = (x_j - x_i)\hat{n}_2^1 + (y_j - y_i)\hat{n}_2^2 + (z_j - z_i)\hat{n}_2^3$$

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

$$T_{ij} = \hat{n}_2^1 \Sigma_{r=1}^{j-1} (y_r - y_i)^2 + \hat{n}_2^2 \Sigma_{r=1}^{j-1} (x_r - x_i)^2 + 2\hat{n}_2^3 \Sigma_{r=1}^{j-1} (x_r^2 + y_r^2)$$

$$+ 2\hat{n}_2^1 \hat{n}_2^2 \Sigma_{r=1}^{j-1} (x_r x_i + y_r y_i) + 2\hat{n}_2^2 \hat{n}_2^3 \Sigma_{r=1}^{j-1} (x_r y_i - y_r x_i)$$

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

6.3 Constant-time update of $\sigma^2$’s
We have seen above that $\hat{n}_1 = -(y_j - y_i), (x_j - x_i), 0$. Let $\hat{n}_1 = (\hat{n}_1^1, \hat{n}_1^2, \hat{n}_1^3)$, and $\hat{n}_2 = (\hat{n}_2^1, \hat{n}_2^2, \hat{n}_2^3)$. (Note $\hat{n}_1^3 = 0$.)

Then $u_j = (P_j - P_i)^T \hat{n}_1$. (Refer Fig 1.) Expanding as before we get

$$U_{ij} = \hat{n}_1^1 \Sigma_{r=1}^{j-1} (y_r - y_i)^2 + \hat{n}_1^2 \Sigma_{r=1}^{j-1} (x_r - x_i)^2 + 2\hat{n}_1^3 \Sigma_{r=1}^{j-1} (x_r x_i + y_r y_i)$$

$$+ 2\hat{n}_1^1 \hat{n}_2^2 \Sigma_{r=1}^{j-1} (x_r x_i + y_r y_i) + 2\hat{n}_1^2 \hat{n}_2^3 \Sigma_{r=1}^{j-1} (x_r y_i - y_r x_i)$$

where again the computation of any $U_{ij} + t$ 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 $\Sigma$ 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/pmm2/.

We evaluated our method using a non-redundant dataset containing 15399 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 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 $\phi$ and $\psi$. The rise ($r$) of the segment with endpoints $P_i$ and $P_j$ is $r = D_{ij}/(j - i + 1)$, 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 $p = n\pi$. 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 level segment 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 $\alpha$, $\pi$-helices and $\beta$-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. \( \phi \) and \( \psi \) are average backbone dihedral angles. \( n \) is the periodicity of the local structure. \( \rho \) is the rise. \( \rho' \) is the pitch.

<table>
<thead>
<tr>
<th>Type</th>
<th>( \phi )</th>
<th>( \psi )</th>
<th>( n )</th>
<th>( \rho )</th>
<th>( \rho' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3(_10)-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>5.4</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>

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>

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

<table>
<thead>
<tr>
<th>SSE</th>
<th>wwPDB</th>
<th>PMML</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_1 )</td>
<td>Lys2-Trp6</td>
<td>Met1-Tyr5</td>
</tr>
<tr>
<td>( \alpha_4 )</td>
<td>Asn11-Glu25</td>
<td>Asn11-Glu27</td>
</tr>
<tr>
<td>( \beta_5 )</td>
<td>Asn31–Asn34</td>
<td>Gly27-Ile33</td>
</tr>
<tr>
<td>( \alpha_8 )</td>
<td>Ile40–Asn45</td>
<td>Asn39-Glu46</td>
</tr>
<tr>
<td>( \beta_7 )</td>
<td>Ile48–Cys53</td>
<td>Asp47-Cys53</td>
</tr>
<tr>
<td>( \alpha_4 )</td>
<td>Phe66-Lys76</td>
<td>Gln65-Thr75</td>
</tr>
<tr>
<td>( \alpha_6 )</td>
<td>Lys81–Thr84</td>
<td>Gly79-Ser87</td>
</tr>
<tr>
<td>( \alpha_0 )</td>
<td>Lys94-Gly105</td>
<td>Gly91-Gly107</td>
</tr>
<tr>
<td>( \beta_6 )</td>
<td>Leu115–Glu118</td>
<td>Gln112-Gln118</td>
</tr>
<tr>
<td>( \alpha_9 )</td>
<td>Asp123–Ile136</td>
<td>Gln120-Gln126,Gln126-Ile136</td>
</tr>
</tbody>
</table>

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 SNLL. The column PMML gives the corresponding residue ranges of the segmentation produced by PMML.
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 αE, β5 and β5 show some discrepancies. The residue range from wwPDB corresponding to αE was approximated by PMML using 2 segments instead of one. The first segment is composed of roughly one turn of the helix at αE’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 β5, the start location identified by PMML precedes the start location identified 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 β1), so the β5 strand may well start at residue Lys28 or Asp29. Similarly, for β5, 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 β--bulge in strand β5, and hydrogen bonds between atoms 80O–109N and 82N–109Q; assignment of the start of the strand β5 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 may 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


