BIOINFORMATICS

Vol. 28 ISMB 2012, pages i97–i105
doi:10.1093/bioinformatics/bts223

# Minimum message length inference of secondary structure from protein coordinate data

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

[1]Clayton School of Information Technology, Monash University, Clayton VIC 3800, Australia and [2]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**

**Motivation:** Secondary structure underpins the folding pattern and architecture of most proteins. Accurate assignment of the secondary structure elements is therefore an important problem. Although many approximate solutions of the secondary structure assignment problem exist, the statement of the problem has resisted a consistent and mathematically rigorous definition. A variety of comparative studies have highlighted major disagreements in the way the available methods define and assign secondary structure to coordinate data.

**Results:** We report a new method to infer secondary structure based on the Bayesian method of minimum message length inference. It treats assignments of secondary structure as hypotheses that explain the given coordinate data. The method seeks to maximize the joint probability of a hypothesis and the data. There is a natural null hypothesis and any assignment that cannot better it is unacceptable. We developed a program `SST` based on this approach and compared it with popular programs, such as `DSSP` and `STRIDE` among others. Our evaluation suggests that `SST` gives reliable assignments even on low-resolution structures.

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

**Contact:** arun.konagurthu@monash.edu (or lloyd.allison@monash.edu)

## 1 INTRODUCTION

Periodic hydrogen-bonding patterns in globular proteins give rise to elements of secondary structure—helices and sheets. The $\alpha$-helix and $\beta$-sheets were among the first structural motifs predicted from first principles of stereochemistry by Pauling and Corey (1951). We now know these specific motifs are almost ubiquitous across the corpus of known structures. Eventually, other repetitive motifs were also identified, and the alphabet of secondary structures was expanded to include $3_{10}$-helix, $\pi$-helix, $\beta$-turn, $\gamma$-turn, $\Omega$-turn and $\beta$-bulges, among other minor elements. In what follows, we use the term secondary structure to include both the classical helices and sheets, and other common substructural elements.

Accurate assignment of secondary structure of proteins from coordinate data is an important and a challenging problem (Andersen and Rost, 2009). Secondary structure underpins the architectural organization in proteins. It simplifies the complex atom-level description of proteins and is therefore the key to generation of schematic diagrams of their three dimensional (3D) folding patterns (Lesk and Hardman, 1982; Richardson, 1981). They are central in training methods geared for predicting secondary structure from amino acid sequence (Andersen and Rost, 2009). They form a linchpin to efficient methods for structural comparison and analysis (Kamat and Lesk, 2007; Konagurthu *et al.*, 2008).

Over the last 30 years, many programs were developed to address the problem of assigning secondary structure to protein coordinate data. A broad classification can be made of the assignment strategies: (i) methods that use distance and angle profiles of local fragments; (ii) methods that detect hydrogen bonds between backbone atoms; (iii) methods that use 3D geometry of local fragments; and (iv) methods that approximate the backbone trace with a set of straight lines.

The following reviews some of the major earlier contributions to the literature of this problem. Levitt and Greer (1977) were the first to generate an automatic method for secondary structure assignment, based on distance and dihedral angle profiles of $C_\alpha$ atoms over a sliding window of four residues. P-SEA (Labesse *et al.*, 1997) is another method in this category which assigns secondary structural states using a short $C_\alpha$ distance mask and two $C_\alpha$ dihedral angle criteria. PROSS (Srinivasan and Rose, 1999) proposes an assignment based solely on backbone dihedral angles. Xtlsstr (King and Johnson, 1999) calculates backbone dihedral angles and distances and assigns secondary structural types that would be consistent with interactions of amide-amide groups observed from circular dichroism of a protein in ultraviolet range (Andersen and Rost, 2009). More recently, PALSSE (Majumdar *et al.*, 2005) was designed to delineate protein structure into helices and strands, mainly using distance and torsion angle constraints to identify core elements which are later extended to longer segments. KAKSI (Martin *et al.*, 2005) is based on $C_\alpha$ distances and backbone dihedral angles and designed primarily to show concordance with the manual assignments found in the protein data bank (PDB).

The most popular method in this space is 'Dictionary of Secondary Structure of Proteins' (DSSP) developed by Kabsch and Sander (1983). DSSP is based on detecting hydrogen bonds between nitrogen and carbonyl groups along the protein polypeptide chain using a Coulomb approximation of the hydrogen-bond energy function (Andersen and Rost, 2009). Many now consider this method a standard for secondary structural assignment (Martin *et al.*, 2005). Since DSSP was published, several methods have been designed that rely on computing the hydrogen-bond energy between backbone atoms. STRIDE (Frishman and Argos, 1995) is among the successful variants of DSSP which uses a modified hydrogen-bond energy function as well as backbone dihedral angles to compute its assignment. SECSTR (Fodje and Al-Karadaghi, 2002) is another variant which improves the detection and assignment of $\pi$-helices which both DSSP and STRIDE have difficulty characterizing (Martin *et al.*, 2005).

*To whom correspondence should be addressed.

There are other methods which assign secondary structure using 3D features in a protein structure. Richards and Kundrot (1988) describe a method, DEFINE-S, to assign secondary structure using local geometry of ideal secondary structures. The P-CURVE (Sklenar *et al.*, 1989) algorithm uses an helicoidal axis approach derived from a series of peptide planes to assign secondary structure. The VoTAP (Dupuis *et al.*, 2004) algorithm relies on Voronoi tessellation of a residue contact map and then matching the contact map profiles to a consensus assignment of secondary structures by methods like DSSP and STRIDE.

In the last category are indirect methods, such as STICK (Taylor, 2001) and PMML (Konagurthu *et al.*, 2011) which work by approximating the $C_\alpha$ spatial trace using a set of lines. These methods seek the best approximation of the protein backbone using piecewise lines. Only as a post-process to this approximation, each line segment is indirectly attributed a secondary structural type based on criteria such as the average rise and pitch of the $C_\alpha$ atoms within the segment. These approaches solve a related yet different problem, namely 'the best line approximation of the protein chain'.

Consistent with this large number of proposed methods, assignment of secondary structure has been recognized to be an 'inexact process' (Cuff and Barton, 1999). Previous comparative studies have highlighted the difficulties of existing programs to assign secondary structure consistently (Andersen and Rost, 2009; Colloc'h *et al.*, 1993; Cuff and Barton, 1999; Martin *et al.*, 2005; Zhang *et al.*, 2008). These disagreements can be major as shown by Colloc'h *et al.* (1993) where the percentage of agreement between DSSP, DEFINE-S and P-CURVE was only 63% on a residue basis. It has been observed that most disagreements arise in the terminal regions of the assigned secondary structural elements. Reflecting on this problem Robson and Garnier (1986) comment [as quoted by Martin *et al.* (2005)]: 'In looking at a model of a protein, it is often easy to recognize helix and to a lesser extent sheet strands, but it is not easy to say whether the residues at the ends of these features be included in them or not. In addition, there are many distortions within such structures so that it is difficult to assess whether this represents merely a distortion, or a break in the structure. In fact, the problem is essentially that helices and sheets in globular proteins lack the regularity and clear definition found in the Pauling and Corey models.'

Given the complexity of the details of individual protein structures, it is not surprising that the secondary structure assignment problem has resisted a mathematically rigorous definition. The effect can be seen in the use of a variety of definitions by the existing tools, although all of them are reasonable. In this study, we describe an approach, `SST`, to the secondary structure assignment problem using minimum message length (MML) inference (Wallace and Boulton, 1968). Linking statistical inference with data compression, the goal is to communicate losslessly the coordinates of a protein using a two-part message. The first part transmits the secondary structure assignment as a *hypothesis* about the coordinates. The second-part transmits the details of coordinates not explained by the hypothesis. This gives rise to statistically robust objective function to optimize: find the best hypothesis on the coordinate data that minimizes the total two-part message length.

`SST` assigns secondary structure segments of the following types: $\alpha$, $3_{10}$ and $\pi$-helix (including left-handed versions of all these helices when they occur), sharp turns, $\beta$-strands and others (coil). `SST` in a post-processing step merges consecutive structures where appropriate, and groups all strands of a sheet, identifies $\beta$-bulges, to convert the results to a molecular biologist's conventional secondary structure description, and produces a PyMol script to visualize the secondary structural assignments. (Fig. 1.)

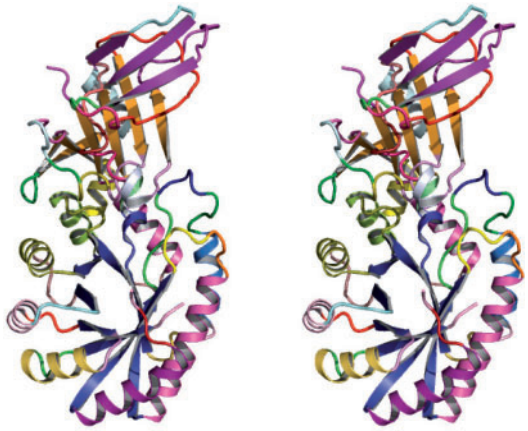

**Fig. 1.** SST assigned secondary structure to coordinates of a 1.6Å crystal structure, Ornithine decarboxylase from mouse.

## 2 OVERVIEW OF MML CRITERION

The MML criterion provides an information–theoretic objective for problems of inference where the goal is to find the best *explanation* (or *theory, hypothesis, model*) for a set of observed data (Wallace and Boulton, 1968). MML relies on quantifying the amount of information *required* to convey losslessly the observed data in an *explanation message*. The best hypothesis is the one which can convey the entire data set in the shortest possible explanation message.

More formally, for some observed data $D$ and a hypothesis $H$ that offers an explanation of the data $D$, Bayes's theorem (Bayes and Price, 1763) gives

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

where $P(H)$ 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*.

Using Shannon's mathematical theory of communication (Shannon, 1948), the amount of information for an explanation of the data $D$ with the hypothesis $H$ is given by

$$I(H\&D) = I(H) + I(D|H) = I(D) + I(H|D)$$

where $I(x) = -\log_2(P(x))$ gives the optimal code length to convey some event $x$ whose probability is $P(x)$.

This immediately gives an objective means to compare competing hypotheses. For hypotheses $H_1$ and $H_2$ on the same data $D$, we have

$$I(H_1|D) - I(H_2|D) = I(H_1) + I(D|H_1) - I(H_2) - I(D|H_2)$$

It follows that the best hypothesis $H^*$ over all competing hypotheses is the one where the expression $I(H^*) + I(D|H^*)$ is minimized.

A concrete realization of the MML framework comes from describing it as a communication process between an imaginary transmitter (Alice) and receiver (Bob) connected over

a Shannon channel. Alice's objective is to send the observed data $D$ using an explanation message in a form such that Bob can receive and decode the data $D$ precisely as Alice sees it. Alice and Bob agree on a *codebook* containing the general rules of communication composed solely of common knowledge about typical, hypothetical data. Anything that is not a part of the codebook must be strictly transmitted as a part of the message. If Alice can find the best hypothesis $H^*$ on the data, Bob will receive a decodable explanation message most economically: The best inference about the data is the hypothesis that minimizes the total message length.

Alice sends the explanation message of $D$ in two parts. In the first part, she transmits the best hypothesis, $H^*$, she could find on the data $D$ taking $I(H^*)$ bits. In the second, she transmits the details of the observed data $D$ not explained by $H^*$, taking $I(D|H^*)$ bits (i.e. the deviations from $H^*$). Notice that MML inference gives a natural trade-off between hypothesis complexity ($I(H^*)$), and its goodness of fit to the data ($I(D|H^*)$).

For a comprehensive resource on MML see Wallace (2005).

## 3 THE DESIGN OF THE COMMUNICATION FRAMEWORK

Protein coordinates for a single-polypeptide chain are represented as an ordered set of 3D points of the form $\mathcal{P}=\{p_1,\cdots,p_n\}$, where any $p_i$ corresponds to the $i$th $C_\alpha$ coordinate along N- to C-terminus of the protein chain. Each $p_i$ defines a 3D real-valued vector $(p_i^x,p_i^y,p_i^z)$ in Angstrom (Å) units, where each component of the vector comes specified (in the PDB) to three positions after the decimal place. Therefore, in this work, we treat the accuracy of measurement of the data as $\epsilon=0.001$ Å (independent of the actual accuracy of the experimental structure determination). The transmitter (Alice) has to send a message to the receiver (Bob) who will then be able to reconstruct the original data from the encoded explanation message *exactly*. For coordinate data from the PDB, Bob will reconstruct each coordinate of each atom to the original precision of three digits after the decimal point.

### 3.1 Null model description of a protein coordinate data

MML gives a natural hypothesis test: The null-model corresponds to transmitting the data raw. If any hypothesis $H$ on the data takes longer than the null model, then clearly $H$ is unacceptable. However, the statement of the raw null model message (without any hypothesis) has to be economical; it must not be willfully inefficient.

The construction of an efficient null model for protein coordinates relies on the observation that the distance between successive $C_\alpha$ atoms in a protein chain is highly constrained at about 3.8Å with only small deviations from this value. The method starts with the transmission of the first $C_\alpha$ coordinate $p_1$ in any choice of encoding that both transmitter and receiver agree on. (Stating $p_1$ simply adds a constant overhead to the message length, whether transmitted via a null model message or an explanation using a hypothesis. A simple way to do away with this overhead is for Alice to translate $\mathcal{P}$ such that $p_1$ becomes the origin. $p_1$ then need not be transmitted explicitly in the message and can be treated implicitly a part of the codebook.) Alice then computes the observed distance $r$ between the successive $C_\alpha$ coordinates $p_1$ and $p_2$. This distance $r$ can be communicated efficiently using an encoding over a normal distribution $\mathcal{N}(\mu,\sigma)$ with a certain fixed mean ($\mu$) and a small standard deviation ($\sigma$) around it. Based on the prior knowledge of $C_\alpha$–$C_\alpha$ distances between successive atoms, these values are set to $\mu=3.8$Å and $\sigma=0.4$Å and are considered to be part of the codebook.

The probability density of a random variable $x$ over a normal distribution with mean $\mu$ and a standard deviation $\sigma$ is given by:

$$\mathcal{N}(x;\mu,\sigma)=\frac{1}{\sqrt{(2\pi)}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Therefore, the probability of stating any distance $r$ to an accuracy of $\epsilon$ (given $\epsilon\ll\sigma$) using the above normal distribution is $P(r)=\epsilon\times\mathcal{N}(x=r;\mu,\sigma)$. This implies

$$P(r)=\frac{\epsilon}{\sqrt{(2\pi)}\sigma}e^{-\frac{(r-\mu)^2}{2\sigma^2}}$$

The optimal code length to transmit $r$ is given by $-\log_2(P(r))$ bits:

$$I(r)=\log_2\left(\frac{\sqrt{(2\pi)}\sigma}{\epsilon}\right)+\frac{(r-\mu)^2}{2\sigma^2}\log_2 e \quad \text{bits.} \qquad (1)$$

Note that Bob will not be able to recover $p_2$ simply from the transmitted information of $p_1$ and the distance $r$ between $p_1$ to $p_2$. $p_2$ can lie anywhere on the surface of a sphere of radius $r$ centered on $p_1$. The precise location of $p_2$ stated to $\epsilon$ can be transmitted by first dividing the surface area of this sphere into cells each of area $\epsilon^2$. This results in $4\pi r^2/\epsilon^2$ such cells distributed uniformly on the surface. These cells can be numbered using a convention that Alice and Bob both agree upon (as a part of the codebook). On the basis of this discretization of the sphere's surface area, Alice transmits the cell number $c$ in which the observed $p_2$ falls within. Assuming uniform probabilities, the probability that the point $p_2$ falls in a cell number $c$ is given by $P(c)=\epsilon^2/4\pi r^2$. Following from this, the code length to state the cell number is:

$$I(c)=-\log_2\left(\frac{\epsilon^2}{4\pi r^2}\right)=\log_2(4\pi r^2)-2\log_2\epsilon \quad \text{bits.} \qquad (2)$$

Bob now has all the information to reconstruct $p_2$ to the precision of $\epsilon$ using the information he has received.

With $p_2$ known at Bob's end, Alice can proceed to encode in the same fashion $p_3$ with respect to $p_2$, then $p_4$ with respect to $p_3$ and so on until all the points in $\mathcal{P}$ are transmitted.

Let $r_i$ ($\forall 1\le i<n$) denote the observed distance between any two successive $C_\alpha$ coordinates $p_i$ and $p_{i+1}$. Let $c_i$ ($\forall 1\le i<n$) denote the cell number on the surface of the sphere of radius $r_i$ centered on the point $p_i$ in which the point $p_{i+1}$ falls. The message length to transmit the entire $C_\alpha$ coordinate data in $\mathcal{P}$ is therefore

$$I_{\text{null}}(p_1,\cdots,p_n)=O(1)+\sum_{i=1}^{n-1}\big(I(r_i)+I(c_i)\big) \quad \text{bits.} \qquad (3)$$

where $O(1)$ denotes the constant number of bits to state $p_1$ (0 bits if Alice translates the coordinates such that $p_1$ lies on the origin).

### 3.2 Models to describe segments of proteins

The secondary structure elements are used as a hypothesis to explain the coordinates. Here, we consider eight models to describe any contiguous stretch of $C_\alpha$atoms (of arbitrary length) along the protein chain: (i) a right-handed $\alpha$-helix; (ii) a left-handed

$\alpha$-helix; (iii) a right-handed $3_{10}$-helix; (iv) a left-handed $3_{10}$-helix; (v) a right-handed $\pi$-helix; (vi) a left-handed $\pi$-helix; (vii) an extended $\beta$-strand; and (8) coil.

The Helical (1–6) and strand (7) models follow ideal Pauling–Corey geometry (Pauling and Corey, 1951) and are of arbitrary length. We term these seven collectively *ideal models*. (Pauling–Corey models are common knowledge and taken to be in the codebook.)

The coil model (8) is treated simply as a model that describes a segment of a protein raw, using the null model approach described above in Section 3.1.

### 3.3 Describing a protein segment using an Ideal model

Assume that at some stage of the transmission Bob has received $C_\alpha$ coordinates up to an intermediate point $p_i$, that is he has received coordinates $(p_1, p_2, \cdots, p_i)$ $(i < n)$. Alice now will transmit a contiguous segment of coordinates $p_i$ to $p_j$ $(1 \leq i < j \leq n)$ using one of the ideal (helical or strand) models. If it is a good model, then the coordinates can be transmitted cheaply. (The discussion of the optimal choice is given in Section 4.)

The number of points to be transmitted in this segment is $j-i$ since the start point of the segment $p_i$ is already known at the receiver's end. The remaining points $p_{i+1}, \cdots, p_j$ are transmitted as follows:

*3.3.1 Transmitting the end point of the segment* The end point of the segment $(p_j)$ is transmitted using the sphere approach similar to the one described in Section 3.1. Instead of the distance between successive $C_\alpha$ coordinates, Alice transmits the distance $d_{\mathrm{ij}}$ between the start $(p_{\mathrm{i}})$ and end $(p_{\mathrm{j}})$ points. This is encoded using a normal distribution where the mean $(\mu)$ is taken as the distance $(d^*)$ between the start and end points from the ideal model containing $j-i+1$ points. The standard deviation $\sigma$ of the end point is set to $\min\left((j-i)\times 0.2\text{Å}, 3\text{Å}\right)$ based on the length of the segment being transmitted and this rule is taken to be a part of the codebook.

On the basis of Equation 1, the code length to state $d_{\mathrm{ij}}$ to the accuracy of $\epsilon$ is given by

$$I(d_{ij}) = \log_2\left(\frac{\sqrt{(2\pi)}\sigma}{\epsilon}\right) + \frac{(d_{ij}-d^*)^2}{2\sigma^2}\log_2 e \qquad \text{bits.} \qquad (4)$$

On the basis of equation 2, given the start point of the segment $p_i$ and distance $d_{ij}$ of the end point $p_j$, the end point can lie anywhere on a sphere with radius $d_{ij}$. $p_j$ can therefore be stated by specifying the cell number $c_{ij}$ on the surface of this sphere in

$$I(c_{ij}) = \log_2(4\pi d_{ij}^2) - 2\log_2\epsilon \quad \text{bits.} \qquad (5)$$

*3.3.2 Encoding the interior points* With the start and end points already known, there are $j-i-1$ interior points of the segment, $p_{i+1}, \cdots, p_{j-1}$, yet to be transmitted. These points can be transmitted cheaply if the chosen ideal model agrees with the observed points in the segment. Alice uses the following procedure to transmit the interior points given a chosen ideal model. (Details of how the optimal choice is made appear in Section 4.)

Consider an ideal model containing $l = j-i+1$ points, denoted formally as $\mathcal{Q} = \{q_1, q_1, \cdots, q_l\}$. The coordinates in $\mathcal{Q}$ are orthogonally transformed to $\mathcal{Q}' = \{q'_1, q'_2, \cdots, q'_l\}$ such that:

(1) $q'_1$ is same as the start point $p_i$ of the segment;

(2) the direction cosines of the vector connecting the start and end points of the ideal model $q'_l - q'_1$ and the direction cosines of the vector connecting the start and end points of the observed segment $p_j - p_i$ are the same; and

(3) the sum of the squared error of the $(l-2)$ interior points of the segment with the corresponding interior points of the ideal model is minimized. That is, $\sum_{1\leq k\leq l-2}\left|p_{i+k} - q'_{1+k}\right|^2$ is minimized, where $|.|$ denotes the Euclidean vector norm.

Such a spatial transformation is related to the more general superposition problem that minimizes the sum of the squared distance between two corresponding vector sets (Kearsley, 1989). However, the transformation is further constrained such that the first points of the two sets are the same (Constraint 1) and the rotational axis for the ideal model is the vector between the start $(p_i)$ and end $(p_j)$ points of the segment (Constraint 2). The first two constraints can be achieved using elementary translation and rotation of the ideal coordinates.

Once $\mathcal{Q}$ is transformed such that the first two constraints are realized, the best rotation $\theta^*$ of $\mathcal{Q}$ about the $p_j - p_i$ axis has to be found so that Constraint 3 is realized. With an approach similar to the generalized superposition problem between two vector sets (Kearsley, 1989), this minimization problem can be solved analytically as an eigenvalue decomposition of a $2\times 2$ square symmetric matrix in quaternion parameters of the corresponding points. (The detailed proof of the analytical method is too long for the main text and hence is provided as Supplementary Material.)

Once the transformation of $\mathcal{Q}$ to $\mathcal{Q}'$ is achieved as described above, Alice can transmit the interior points of the segment, $p_{i+1}, \cdots, p_{j-1}$ by:

(1) transmitting the best rotation about the $p_j - p_i$ axis of the ideal model. (Note, Bob has already received the start and end points, $p_i$ and $p_j$.);

(2) transmitting the interior points $p_{i+1}\cdots p_{j-1}$ as spatial deviations from their corresponding transformed interior points of the ideal model. (Bob already knows $p_i$ and the coordinates of $\mathcal{Q}$ of the ideal model from the codebook. After he receives the end point of the segment $p_j$ (using the sphere approach described above), the ideal coordinates can be transformed such that Constraints 1 and 2 of the transformation discussed above are realized. After Bob receives the rotation $\theta^*$, the ideal coordinates are rotated by that angle around the axis $p_j - p_i$ whose information he already has. Once Alice sends the spatial deviations of interior points $p_{i+1}, \cdots, p_{j-1}$ with respect to the transformed ideal coordinates, Bob can reconstruct the observed interior points of the segment.)

*3.3.3 Transmitting the rotation* Rotation $\theta^*$ is transmitted using a uniform distribution over a circle whose radius $r_{\theta^*}$ is the farthest distance of an interior point of the ideal model from the axis of rotation. Note that $r_{\theta^*}$ need not be transmitted because it is a property of the coordinates of the ideal model which the receiver already knows as a part of the codebook.

The rotation is transmitted by dividing the circumference of a circle of radius $r_{\theta^*}$ into arc segments of length $\epsilon$ and stating the

segment number in which the rotated coordinate with the farthest radius to the axis falls. Thus, the code length of stating $\theta^*$ is

$$I(\theta^*) = -\log_2\left(\frac{\epsilon}{2\pi r_{\theta^*}}\right) = \log_2(2\pi r_{\theta^*}) - \log_2\epsilon \quad \text{bits.} \tag{6}$$

*3.3.4 Transmitting the interior points as spatial deviations* Let any error vector of an interior point of the segment with respect to the corresponding transformed interior point of the ideal model, $e_k \equiv \left(p_{i+k} - q'_{1+k}\right), 1 \le k \le l-2$ have the vector components $(\Delta x_k, \Delta y_k, \Delta z_k)$.

Each $\Delta x$, $\Delta y$ and $\Delta z$ of an interior point is transmitted using a normal distribution with a $\mu$ of 0 and a standard deviation $\sigma$ set to the sample standard deviation computed from these error components.

Wallace (2005) gives the MML estimate of code length to transmit a set of independent data $(\Delta x_1, \Delta y_1, \Delta z_1)$, $(\Delta x_2, \Delta y_2, \Delta z_2)$, $\cdots$ $(\Delta x_{l-2}, \Delta y_{l-2}, \Delta z_{l-1})$ using a normal distribution as:

$$\begin{aligned} I(\Delta x\text{'s}, \Delta y\text{'s}, \Delta z\text{'s}) = {} & \frac{1}{2}(M-1)\log_2\sigma^2 + \frac{M-1}{2} \\ & + \frac{M}{2}\log_2\left(\frac{2\pi}{\epsilon^2}\right) + \frac{1}{2}\log_2\left(2N^2\right) \\ & + \log_2\left(R_\sigma\right) + 1 + \log_2\kappa_1 \text{ bits.} \end{aligned} \tag{7}$$

where $M = 3\times(l-2)$ is the total number of components of the error vectors $e_k$ being transmitted, $R_\sigma$ gives the prior knowledge of the limits to $\log_2\sigma$ and $\kappa_1 \approx 1/12$ denotes the constant corresponding to quantizing lattices proposed by Conway and Sloane (1984). In this study, we assume that $\sigma$ is bounded by 3Å because this is consistent with the limits of utility of root-mean-squared-deviation (RMSD) in superposition as a measure to estimate protein structural similarity.

Therefore, combining code lengths from equations 4–7, the code length required to transmit coordinates of a segment of a protein using any ideal model is give by

$$\begin{aligned} I_{\text{ideal}}(p_i, \cdots, p_j) = {} & I(d_{ij}) + I(c_{ij}) + I(\theta^*) \\ & + I(\Delta x\text{'s}, \Delta y\text{'s}, \Delta z\text{'s}) \text{ bits.} \end{aligned} \tag{8}$$

### 3.4 Describing a protein segment using the coil model

When transmitting a segment of a protein $p_i, \cdots, p_j$ as a coil, the coordinates are stated raw in that range using a null model (Section 3.1). Therefore, the code length of stating a segment $p_i, \cdots p_j$ as a coil is

$$I_{\text{coil}}(p_i, \cdots, p_j) = \sum_{k=i}^{j-1}\left(I(r_k) + I(c_k)\right) \quad \text{bits.} \tag{9}$$

where $I(r_k)$ and $I(c_k)$ are code lengths given in equations 1 and 2.

### 3.5 Describing the protein as a collection of segments

Having laid the foundations of encoding segments of a protein using one of the ideal models or the coil model (Sections 3.3 and 3.4), this section deals with describing the entire protein coordinates as a collection of segments of a model type.

The main idea here is to find the best decomposition of points $\mathcal{P}$ of a protein into segments where each segment is described using exactly one of eight potential models. Note that the decomposition, with the associated model descriptors, gives a secondary structural hypothesis of a protein.

Formally, a segmentation of $\mathcal{P} = \{p_1, \cdots, p_n\}$ gives an ordered subset of points $\mathcal{P}' = \{p'_1 \equiv p_{i_1}, p'_2 \equiv p_{i_2}, \cdots, p'_m \equiv p_{i_m}\}$ where $1 = i_1 < i_2 < \cdots < i_m = n$. Each successive pair of points in $\mathcal{P}'$, $\langle p'_1, p'_2\rangle$, $\langle p'_2, p'_3\rangle, \cdots \left\langle p'_{m-1}, p'_m\right\rangle$, defines the start and end points of a segment. (Notice that $\mathcal{P}'$ gives $m-1$ segments of P, where end point of one segment is same as the start point of the next.) Associated with each segment $\left\langle p'_k, p'_{k+1}\right\rangle$ of length $l_k = i_{k+1} - i_k + 1$ is a model type $t_k$, $1 \le k \le m-1$. A secondary structural assignment of $\mathcal{P}$ is given by the segmentation $\{p'_1, \cdots, p'_m\}$ and its corresponding model assignment $\{t_1, \cdots, t_{m-1}\}$.

We note that for $n$ points in $\mathcal{P}$, there are $2^{n-2} = \left(\binom{n-2}{0} + \binom{n-2}{1} + \cdots + \binom{n-2}{n-2}\right)$ possible segmentations – the first and last points of $\mathcal{P}'$ are the same as those in $\mathcal{P}$. Since each segment of any segmentation can be assigned to any of the eight possible model types, the total possible secondary structural assignments is given by the formula: $\left(8\times\binom{n-2}{0} + 8^2\times\binom{n-2}{1} + \cdots + 8^{n-1}\times\binom{n-2}{n-2}\right)$. For an average protein, this gives a massive search space. (An efficient dynamic programming method to find the best secondary structural assignment is detailed in Section 4.)

Any given segmentation $\mathcal{P}'$ of $\mathcal{P}$ and its associated model types acts as a secondary structural hypothesis of the given coordinate data. Alice can describe and transmit the coordinates in $\mathcal{P}$ using this hypothesis over a two-part message.

*3.5.1 First part of the explanation message* In the first part of the message, Alice communicates the segmentation $\mathcal{P}' = \{p'_1, \cdots, p'_m\}$ and its corresponding model assignments $\{t_1, \cdots, t_{m-1}\}$ as the hypothesis on the observed coordinate data in $\mathcal{P}$. This part of the message will be composed of the following:

(1) the number of segments $(m-1)$ in $\mathcal{P}'$ and

(2) for each segment $p'_k$ $(1 \le k \le m-1)$, communicate

   (a) the length of the segment $l_k = i_{k+1} - i_k + 1$ and

   (b) the model type $t_k$ to encode the points in that segment.

The number of segments $(m-1)$ is an integer transmitted using a log* distribution assuming a universal prior on the distribution of numbers. Rissanen (1983) gives the code length of transmitting any integer $n > 0$ as

$$I_{\log^*}(n) = \log_2^*(n) + \log_2(2.865) \tag{10}$$

where $\log_2^*(n) = \log_2 n + \log_2\log_2 n + \cdots$ (over all +ve terms).

Next, the lengths of the segments are positive integers. Although these integers can also be transmitted using a log* distribution, it is rather inefficient because in practice the lengths of helices, strands and coils are constrained. Therefore, in this work, we, encode the lengths of the segments using a Poisson distribution with a predefined mean of $\lambda$ for each model type: (The parameters $\lambda$ for each of the eight types of models are treated to be a part of the codebook. In this work we empirically set $\lambda = 4$ for coil and $\lambda = 5$ for strands. The lengths of helices are transmitted using a mixture of two Poisson distributions with means 4 and 8.)

$$f(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!}$$

The code length to state any integer $n>0$ using this distribution is:

$$I_{\text{poisson}}(n) = -\log_2(f(x=n;\lambda)) \quad (11)$$
$$= \lambda\log_2 e - n\log_2\lambda + \log_2 n! \quad \text{bits}$$

Finally, each model type $t$ (encoded as an integer $0\leq t\leq 7$) of any segment is stated using a uniform distribution (uniform is the simplest choice. Since some models as more probable than others (e.g. $\alpha$-helices and strands are significantly more probable than other models), a more elaborate coding scheme can also be considered taking into account the empirical distribution of various models.) in

$$I_{\text{uniform}}(t) = \log_2(8) = 3 \text{ bits.} \quad (12)$$

In summary, combining equations 10–12, the code length of the first part of the message proposing the secondary structural hypothesis on $\mathcal{P}$ is

$$I_{\text{first}}(\mathcal{P}') = I_{\log^*}(m-1) + \sum_{k=1}^{m-1}\left(I_{\text{poisson}}(l_k) + I_{\text{uniform}}(t_k)\right) \quad (13)$$

*3.5.2 Second part of the explanation message* In the second part of the message, Alice sends the actual details of the coordinates in $\mathcal{P}$ economically given the hypothesis $\mathcal{P}'$.

The procedure to transmit coordinate data of a segment of a protein using the ideal and coil models has been discussed in Sections 3.3 and 3.4.

Using the notations in Section 3.5, the hypothesis received in the first part of the message is of the form $\left\langle p'_k, p'_{k+1}\right\rangle$, with each segment of some length $l_k$ and type $t_k$. The message length to transmit the coordinates given the above segmentation is:

$$I_{\text{second}}(\mathcal{P}|\mathcal{P}') = O(1) + \sum_{k=1}^{m-1} I_{\text{model}}(p'_k,\cdots,p'_{k+1}) \quad \text{bits} \quad (14)$$

where $O(1)$ is the constant number of bits to state the first $C_\alpha$ coordinate $p_1$ (Section 3.1), $I_{\text{model}}(p'_k,\cdots,p'_{k+1}) = I_{\text{ideal}}(p'_k,\cdots,p'_{k+1})$ if $0\leq t_k\leq 6$ and $I_{\text{model}} = I_{\text{coil}}(p'_k,\cdots,p'_{k+1})$ if $t_k = 7$.

The total message length of communicating $\mathcal{P}$ with $\mathcal{P}'$ comes from combining equations 13 and 14

$$I_{\text{total}}(\mathcal{P}\&\mathcal{P}') = I_{\text{first}}(\mathcal{P}') + I_{\text{second}}(\mathcal{P}|\mathcal{P}') \quad (15)$$

### 3.6 Problem Statement

From the earlier section, the problem of inferring the best secondary structural assignment can now be stated formally as follows: given $\mathcal{P}$ containing $n$ points, find the secondary structural segmentation $\mathcal{P}'$ and its corresponding model assignment such that the total message length to transmit $\mathcal{P}$ losslessly, $I_{\text{total}}(\mathcal{P}\&\mathcal{P}') = I_{\text{first}}(\mathcal{P}') + I_{\text{second}}(\mathcal{P}|\mathcal{P}')$ is minimized.

## 4 INFERENCE OF SECONDARY STRUCTURE

This section describes the search method to find the best MML secondary structural segmentation from given coordinate data.

### 4.1 Constructing the code length matrices

Equation 15 gives the total message length of communicating coordinates in $\mathcal{P}$ using a segmentation $\mathcal{P}'$:

$$I_{\text{total}}(\mathcal{P}\&\mathcal{P}') = I_{\text{first}}(\mathcal{P}') + I_{\text{second}}(\mathcal{P}|\mathcal{P}')$$
$$= I_{\log^*}(m-1) + \sum_{k=1}^{m-1}\left(\begin{array}{l} I_{\text{poisson}}(l_k) + I_{\text{uniform}}(t_k) \\ \quad + I_{\text{model}}(p'_k,\cdots,p'_{k+1})\end{array}\right)$$

For a given protein, any pair of points can potentially be the start and end points of a segment. At the same time, a segment can be described using any of the eight models considered here. Therefore, the procedure to assign secondary structure to $C_\alpha$ coordinates $\{p_1,\cdots,p_n\}$ in a given protein begins by constructing a set of eight code length matrices, one for each model type $t$ ($0\leq t\leq 7$):

$$H^t(i,j) = I_{\text{poisson}}(j-i) + I_{\text{uniform}}(t) + I_{\text{model}}(p_i,\cdots,p_j) \quad (16)$$

where any cell $(i,j)_{1\leq i<j\leq n}$ of the matrix for type $t$ gives the code length of stating the segment $p_i$ to $p_j$ using the model $t$.

### 4.2 Finding the best secondary structural assigning

The segmentation of $\mathcal{P}$ using various model types enforces a strict ordering constraint, satisfying the requirements for a solution by dynamic programming, even though the search space is huge as discussed in Section 3.5. Let any $D(i)$ store the optimal message length of transmitting points $p_1,\cdots,p_i$, for all $1\leq i\leq n$. With the boundary condition of $D(1)=0$, the dynamic programming recurrence to find the optimal assignment is given by

$$D(j) = \min_{i=1}^{j-1}\left\{\begin{array}{ll}\min_t H^t(1,j) & \\ D(i) + \min_t H^t(i,j) & 1<j\leq n\end{array}\right.$$

The above recurrence is used to fill the array $D$ iteratively from 1 to $n$. On completion, the best secondary structure assignment can be derived by remembering the index $i$ and type $t$ from which the optimal $D(j)$ is computed.

## 5 POST-PROCESSING

In the post-processing step, the above-defined successive segments of same type (helical or strand) from the MML inference are examined for moderate curvature. Further, sharp turns are identified and distinguished from coil assignments. Finally, $\beta$-sheets are identified by grouping together the assigned strands.

*Checking for moderate curvature in helix and strand*: The MML inference automatically gives the best (in the information–theoretic sense) piecewise approximation of a curved helix or strand. In a single pass through the secondary structural assignment generated by our method, we check for such curvatures and merge successive segments to form a larger segment.

Each helix ($1\leq t\leq 6$) and strand ($t=7$) is represented by a vector. For a helical segment, the vector is the axis of the helix. For the strand segment, the vector is the least-square line fitting its $C_\alpha$ atoms. Two successive segments (of the same model type) are joined to form a single segment if the orientation angle of their vectors is within 30°. [For a detailed description of the finding the axes and orientation angle, see Konagurthu *et al.* (2008)]

*Identifying sharp turns*: sharp turns often have geometries that are conformationally similar to a turn of ideal helical models considered

in this work. Any assignment of a helical segment of length less than or equal to four residues that is preceded and succeeded by other assigned segments is relabeled as a sharp turn.

However, $\alpha$-helices often fray or tighten at their N- or C-terminal ends giving a short stretch of $\pi$ or $3_{10}$ helical segment. In order not to incorrectly assign these ends as sharp turns, the orientation of a candidate segment is checked against the preceding and succeeding segments and only assigned a sharp turn if the segment's orientation exceeds 45° relative to its neighbouring segments.

*Grouping strands forming $\beta$-sheet*: all strand segments are extracted from a given assignment. A strand adjacency matrix is computed where two strands are treated to be adjacent if and only if they are in parallel (with orientation angle in the range $\pm 45°$) or anti-parallel (with orientation angle in the range [135° to 180°] or [−135° to −180°]) orientation and there exist Van der Waals interactions between at least two pairs of atoms from the segment.

Strands from a $\beta$-sheet are then identified and grouped using a complete depth-first search on the strand adjacency matrix.

## 6 RESULTS

*Implementation*: a program (SST) implementing the method described in the previous section has been developed in C++ programming language. The program accepts protein coordinates in the Brookhaven PDB format and outputs the secondary structure assignment both at a segment level (stating the start and end point of each secondary structural segment) as well as at a residue level. The program also generates a PyMol script, which allows users to visualize the secondary stucture assignment. (Fig. 1).

*Datasets and comparison methods*: To study the performance of SST, we consider a dataset of 1737 PDB structures. These structures are the same as the dataset considered by Martin *et al.* (2005), excluding the structures which have been deprecated or those structures that failed running on any one of the considered methods. (See below.) These structures are divided into four datasets: high-resolution (HRes) dataset with 631 crystal structures solved to 1.7 Å or better; medium resolution (MRes) dataset with 582 crystal structures with resolution between 1.7 and 3 Å resolution; low-resolution (LRes) dataset with 306 structures with resolution >3 Å and finally, a dataset of 218 NMR structures.

In this work, we mainly compare SST with DSSP (Kabsch and Sander, 1983) and STRIDE (Frishman and Argos, 1995) exhaustively on the large structural dataset described earlier. Although there are other programs for secondary structure assignment, we had difficulty finding distributions that we could download. For those we managed to download, we could not install the programs due to unresolvable dependencies in their source code. However, we use a web server 2Struct (Klose *et al.*, 2010), which allows manual submission of queries with a single point of access to a variety of secondary structure assignment methods beyond DSSP and STRIDE such as KAKSI (Martin *et al.*, 2005), PALSSE (Majumdar *et al.*, 2005), STICK (Taylor, 2001), XtlSStr (King and Johnson, 1999) and P-SEA (Labesse *et al.*, 1997). To facilitate comparisons with other methods, we randomly selected 30 low-resolution structures from the LRes dataset and manually collected the secondary structure assignments from the server. The list of structures used in this experiment can be download from http://www.csse.monash.edu.au/~karun/sst.

**Table 1.** Performance of SST compared with DSSP and STRIDE on four datasets: HRes, MRes, LRes and NMR

| | HRes (% Agreement) | | | MRes (% Agreement) | | |
|---|---|---|---|---|---|---|
| | Helix | Strand | Total | Helix | Strand | Total |
| SST versus DSSP | 97.6 | 81.9 | 84.1 | 97.6 | 83.4 | 83.9 |
| SST versus STRIDE | 97.1 | 80.8 | 84.3 | 97.2 | 82.3 | 84.3 |
| STRIDE versus DSSP | 99.4 | 98.5 | 96.7 | 99.4 | 98.9 | 96.9 |
| | LRes (% Agreement) | | | NMR (% Agreement) | | |
| | Helix | Strand | Total | Helix | Strand | Total |
| SST versus DSSP | 97.7 | 84.3 | 82.7 | 98.4 | 53.8 | 51.5 |
| SST versus STRIDE | 97.2 | 82.3 | 83.8 | 94.4 | 78.9 | 83.9 |
| STRIDE versus DSSP | 99.3 | 98.3 | 96.0 | 99.6 | 64.6 | 68.7 |

Columns labelled 'Helix' and 'Strand' give the percentage agreement of residues assigned as helix and strand, respectively, between the two methods. Column 'Total' gives the percentage agreement over three classes: helix, strand and others.

*Comparison*: we first assess the composition of the assigned regular secondary structures (helices and strands) using DSSP, STRIDE and SST over the four datasets described earlier. Overall DSSP assigns 34.5% of residues to helices and 19.9% of residues to strands. STRIDE assigns 37.3 and 27.4%, and SST assigns 40.4 and 25.3% of residues to helices and strands, respectively. In general, DSSP is conservative in assigning regular secondary structures resulting in shorter elements compared with output from STRIDE and SST. Examining the lengths of various secondary structural segments, we observe that the average length of a helix and strand segment assigned by SST is 12.1 and 6.1 residues, respectively. Unlike SST, which states the residue start and end points of each segment, computing the lengths of secondary structural elements from DSSP and STRIDE's output is problematic and error-prone (Majumdar *et al.*, 2005).

Table 1 shows the comparison between SST, DSSP and STRIDE over high, medium, low resolution and NMR structures. To undertake this comparison, the assignments of the three programs are grouped into three classes: helix (of all types), strand and other. DSSP and STRIDE use similar methods for assignment based on detecting hydrogen bonds. Therefore, as one would expect, their assignments are highly similar. SST largely agrees with DSSP and STRIDE when assigning helices. Strands, however, show some disagreement. Visually examining several instances, we find that in many cases SST assigns longer strands than the other two methods. Agreement between strand assignments on NMR structures among the three methods is rather poor. Surprisingly, even DSSP and STRIDE differ enormously in this class even though their assignment methods are quite similar.

Table 2 extends the comparison of SST with other popular methods for assignment on low-resolution structures. The table shows the percentage agreement of secondary structural assignments between the methods. Consistent with previous comparative studies (Colloc'h *et al.*, 1993; Martin *et al.*, 2005), we see considerable differences in the assignments. In the absence of a universally acknowledged *gold standard* for assignment, it becomes very difficult (if not impossible) to objectively validate one method to be truly better than the other. The observed differences mainly arise from the different criteria used by the methods. However,

**Table 2.** Pairwise comparison between secondary structure assignment methods

| | DSSP (%) | STRIDE (%) | KAKSI (%) | PALSSE (%) | P-SEA (%) | STICK (%) | XtlSStr (%) |
|---|---|---|---|---|---|---|---|
| SST | 77.4 | 75.3 | 74.3 | 80.7 | 53.9 | 69.0 | 74.9 |
| DSSP | | 86.1 | 76.9 | 76.0 | 57.1 | 71.8 | 77.6 |
| STRIDE | | | 67.8 | 74.8 | 48.8 | 64.5 | 73.8 |
| KAKSI | | | | 75.4 | 67.4 | 79.4 | 72.9 |
| PALSSE | | | | | 50.9 | 69.9 | 70.8 |
| P-SEA | | | | | | 66.6 | 54.1 |
| STICK | | | | | | | 66.7 |

Each cell in the upper-triangular matrix gives the percentage agreement of the residue-level assignments between a pair of methods indicated in the first row and column. The agreement is measured over all three classes: helix, strand and other.


**Fig. 2.** Residue-level secondary structure assignment of a 1.75 Å flavodoxin structure from *Clostridium beijerinckii*. SST residue-level assignment and the segment boundaries are shown in addition to the assignments across multiple methods. For details of the secondary structure codes, see http://www.csse.monash.edu.au/ karun/sst/codes.html

manually examining many cases where the methods differ, we find that most disagreements appear at the ends of various (helical or strand) segments. We will use a simple example to highlight the most common type of differences. Figure 2 gives the overall residue-level secondary structure assignment across different methods for a flavodoxin structure from *Clostridium beijerinckii* (wwPDB ID:5NLL). DSSP and STRIDE assignments are nearly identical to each other. From the figure, small disagreements between methods can be seen around the start and end points of various segments demarcated by SSTs segment view (labelled SST(SEG)). A major difference between DSSP and SST is the region Lys28…Asn34, which SST assigns as a strand. DSSP starts the segment three residues further at Asn31. Inspecting the structure, we find a backbone hydrogen bond between Asp29 and Met1. This might suggest the start of the strand at either Asp29 or one residue upstream at Lys28 as identified by SST. Also, in the region Glu62…Ile73, only SST correctly assigns a $\pi$-helical cap (Glu62…Phe66) leading into a $\alpha$-helix.

To evaluate the consistency of SSTs secondary structural assignments on coordinates solved at different resolutions, we randomly selected 15 protein structures for which both the superseded low-resolution coordinates and the new high-resolution coordinates were available. Table 3 gives the list of considered structures along with the percentage agreement between SSTs assignment at different resolutions. The results indicate that SST produces consistent results on structures determined at different resolutions. The <10% differences (Table 3, last column) in agreement on the chosen structures may well represent genuine structural differences rather than shortcomings of the algorithm.

Further, to illustrate the reliable segmentation produced by SST on structures with long, curved helices and strands, we chose two structures: Leucine zipper protein (wwPDB: 1NKP) composed of very long helices and Sucrose-specific porin protein

**Table 3.** SST assignment sensitivity to changes in coordinate resolution. Resolution numbers marked with * are taken from the original papers

| Structure name | LRes PDB ID | HRes PDB ID | %Agree |
|---|---|---|---|
| Lysozyme | 2LZH (6.0 Å) | 2ZQ3 (1.6 Å) | 95.3 |
| Ferrochelatase | 1LD3 (2.6 Å) | 1DOZ (1.8 Å) | 97.4 |
| Glutamate Dehydrogenase | 1AUP (2.5 Å) | 1BGV (1.9 Å) | 90.6 |
| Pseudomonas Cytochrome | 151C (2.0 Å) | 351C (1.6 Å) | 93.9 |
| Bence-Jones Protein | 1BJL (2.9 Å)* | 3BJL (2.3 Å) | 90.2 |
| Concanavalin A | 4CNA (2.9 Å)* | 5CNA (2.0 Å) | 91.8 |
| Endochitinase | 1BAA (2.8 Å) | 2BAA (1.8 Å) | 95.5 |
| Ferredoxin Reductase | 1FNR (2.6 Å) | 1FND (1.7 Å) | 95.9 |
| Endonuclease III | 1ABK (2.0 Å) | 2ABK (1.6 Å) | 97.6 |
| Myohemerythrin | 1MHR (2.9 Å)* | 2MHR (1.3 Å) | 92.4 |
| Phosphofructokinase | 5PFK (7.0 Å)* | 6PFK (2.6 Å) | 95.3 |
| Serine Protease Inhibitor | 1QLP (2.9 Å) | 2PSI (2.0 Å) | 95.4 |
| Dimeric Hemoglobin | 1SDH (2.4 Å)* | 3SDH (1.4 Å) | 98.4 |
| Glutathione Reductase | 1GRS (3.0 Å)* | 3GRS (1.5 Å) | 94.4 |
| Calmodulin Fragment TR2C | 1TRC (3.6 Å) | 1FW4 (1.7 Å) | 93.9 |

(wwPDB: 1A0S) composed of long, curved strands forming $\beta$-barrels. Although SST initially breaks the curved segments into smaller pieces, the post-processing step explained in Section 5 reconstitutes these pieces back correctly into fuller segments. [Fig. 3 gives SSTs assignment on the porin protein (1A0S). The figure shows that the curved strands of the $\beta$-barrels have been reconstituted and grouped reasonably well in the post-processing step.]

Finally, as a difficult case we consider the 10 Å resolution protein coordinates of Elongation Factor Tu (GDB.Kirromycin) from *Escherichia coli* (wwPDB: 1qzd) solved using Cryo-Electron Microscopy. Its wwPDB file contains only $C_\alpha$ coordinate information. DSSP, STRIDE and P-SEA fail to process such

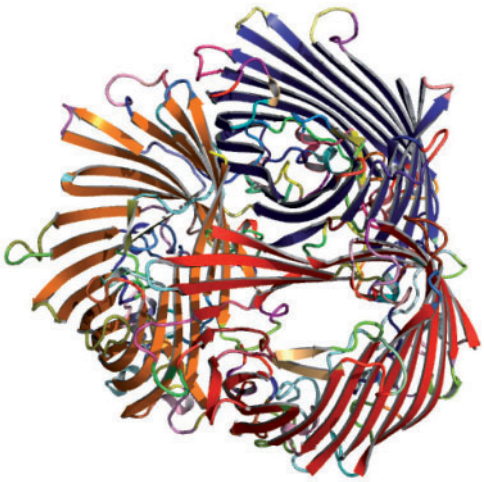

**Fig. 3.** Automatically generated PyMol image of SSTs secondary structural assignment on sucrose-specific porin (ScrY) from *Salmonella typhimurium* (wwPDB: 1A0S)

information as the coordinates of other atoms are needed to decipher Hydrogen bonds. KAKSI and XtlSStr are able to process this structure but assign all residues in the chain to coil. Of the considered methods, only SST, PALSSE and STICK assigned any secondary structure. For lack of space, the overall residue-level assignment across these three methods are presented in Figure 1 of the supplementary text. Examining the structure, PALSSE consistently overestimates the regular secondary structural regions by a large margin. STICK performs well, especially in identifying $\beta$-strands. However, it miscalculates several secondary structural elements. In comparison, SST produced the most reasonable segmentation of the three methods on visual inspection of the structure.

## 7 CONCLUSION

Reliable secondary structure assignment is an important problem. We have developed a novel information theoretic method to address this problem using the Bayesian framework of MML inference. Careful examination of the results over a large number of structures suggests that our method gives consistent assignments even on low-resolution data. We note that our method uses a dictionary of models composed of ideal secondary structural elements. The details of the models are explicit and open to scrutiny. It is likely that these models can be improved. ('Essentially, all models are wrong, but some are useful.'—George Box.) However, modification to the models is an improvement if, and only if, it yields extra compression.

## 8 ACKNOWLEDGEMENTS

LA thanks Sally P. Allison for support. ASK acknowledges helpful discussions with Peter J. Stuckey during the development of this work, and thanks to Rekha Amar for proof reading this manuscript.

*Funding*: ASK's research is supported by Monash Larkins Fellowship.

*Conflict of Interest*: none declared.

## REFERENCES

Andersen,C.A. and Rost,B. (2009) Secondary structure assignment. In Gu,J. and Bourne,P.E. (eds.), *Structural Bioinformatics*, Wiley-Blackwell, pp. 459–484.

Bayes,T. and Price,R. (1763) An essay towards solving a problem in the doctrine of chance. *Philos. Trans. Roy. Soc. Lond.*, **53**, 370–418.

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

Conway,J.H. and Sloane,N.J.A. (1984) On the Voronoi regions of certain lattices. *SIAM Journal on Algebraic and Discrete Methods*, **5**, 294–305.

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.

Fodje,M. and Al-Karadaghi,S. (2002) Occurrence, conformational features and amino acid propensities for the $\pi$-helix. *Protein Eng.*, **15**, 353–358.

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: Struct. Funct. Bioinformatics*, **66**, 869–876.

Kearsley,S.K. (1989) On the orthogonal transformation used for structural comparisons. *Acta. Cryst.*, **A45**, 208–210.

King,S. and Johnson,W. (1999) Assigning secondary structure from protein coordinate data. *Proteins*, **35**, 313–320.

Klose,D.P. *et al.* (2010) 2Struct: the secondary structure server. *Bioinformatics*, **20**, 2624–2625.

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

Konagurthu,A.S. *et al.* (2011) Piecewise linear approximation of protein structures using the principle of minimum message length. **27**, i43i51.

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 Hardman,K.D. (1982) Computer-generated schematic diagrams of protein structures. *Science*, **216**, 539–540.

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.

Martin,J. *et al.* (2005) Protein secondary structure assignment revisited: a detailed analysis of different assignment methods. *BMC Struct. Biol.*, **5**, 17.

Pauling,L. and Corey,R. (1951) Configurations of polypeptide chains with favored orientations around single bonds: Two new pleated sheets. *Proc. Natl. Acad. Sci. USA*, **37**, 729–740.

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.

Rissanen,J. (1983) A universal prior for integers and estimation by minimum description length. *Ann. Stat.*, **11**, 416–431.

Robson,B. and Garnier,J. (1986) *Introduction to Proteins and Protein Engineering*. Elsevier Science Ltd. Amsterdam.

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

Sklenar,H. *et al.* (1989) Describing protein structure: a general algorithm yielding complete helicoidal parameters and 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. (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. *Comput. J.*, **11**, 185–194.

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

Zhang,W. *et al.* (2008) Assessing secondary structure assignment of protein structures by using pairwise sequence-alignment benchmarks. *Proteins*, **71**, 61–67.

# Supplementary text to "Minimum Message Length Inference of Secondary Structure from Protein Coordinate Data"

Arun S. Konagurthu, Arthur M. Lesk and Lloyd Allison

## 1 Analytical solution to solve the constrained least-sqaure superposition problem (See Section 3.3.2 in the main text.)

### 1.1 Preliminaries.

Let $Q = (q_1, q_2, q_3, q_4) \equiv [\![\ q_1\ ,\ \ \vec{\mathbf{q}}\ ]\!]$ be considered a general *quaternion* containing the combination of a scalar and a vector in $\Re^3$. Given two quaternions $P$ and $Q$, the *sum* is given by, $P+Q \equiv [\![\ p_1+q_1\ ,\ \ \vec{\mathbf{p}}+\vec{\mathbf{q}}\ ]\!]$ and the *product* can be expressed as $PQ \equiv [\![\ p_1q_1-\vec{\mathbf{p}}\cdot\vec{\mathbf{q}}\ ,\ \ p_1\vec{\mathbf{q}}+q_1\vec{\mathbf{p}}+\vec{\mathbf{p}}\times\vec{\mathbf{q}}\ ]\!]$. The *norm* of a quaternion $Q$, $\mid Q \mid$ is given by $\sqrt{(q_1^2 + \vec{\mathbf{q}}\cdot\vec{\mathbf{q}})}$.
The *inverse* $Q^{-1}$ is given by $\frac{[\![\ q_1\ ,\ \ -\vec{\mathbf{q}}\ ]\!]}{|Q|^2}$. A vector $\vec{\mathbf{v}}$ can be treated as a quaternion with a zero scalar component, $[\![\ 0\ ,\ \ \vec{\mathbf{v}}\ ]\!]$.

A *pure rotation* about the unit vector $\hat{\mathbf{n}}$ by an angle $\theta$ can be computed using the unit quaternion $\hat{Q}$ (with unit norm) as $\hat{Q} = [\![\ \cos\frac{\theta}{2}\ ,\ \ \hat{\mathbf{n}}\sin\frac{\theta}{2}\ ]\!]$. Then rotation of a vector $\vec{\mathbf{v}}$ to a transformed position $\vec{\mathbf{v}}'$ is given by:

$$[\![\ 0\ ,\ \ \vec{\mathbf{v}}'\ ]\!] = \hat{Q}^{-1}[\![\ 0\ ,\ \ \vec{\mathbf{v}}\ ]\!]\hat{Q}. \tag{1}$$

### 1.2 A solution to the constrained version of the superposition problem.

Let $X = (\vec{\mathbf{x}}_1, \cdots, \vec{\mathbf{x}}_n)$ and $X' = (\vec{\mathbf{x}}'_1, \cdots, \vec{\mathbf{x}}'_n)$ be two corresponding vector sets where $(\forall 1 \leq i \leq n)$ $\vec{\mathbf{x}}_i \equiv x_i\hat{\mathbf{i}} + y_i\hat{\mathbf{j}} + z_i\hat{\mathbf{k}}$ and $\vec{\mathbf{x}}'_i \equiv x'_i\hat{\mathbf{i}} + y'_i\hat{\mathbf{j}} + z'_i\hat{\mathbf{k}}$ where $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ are unit vectors along the coordinate axes in $\Re^3$.

Assume that the set $X$ has been transformed such that $\vec{\mathbf{x}}'_1 = \vec{\mathbf{x}}_1$, and the direction cosines of the vector $\vec{\mathbf{x}}'_n - \vec{\mathbf{x}}'_1 = \vec{\mathbf{x}}_n - \vec{\mathbf{x}}_1 \equiv \hat{\mathbf{n}}$

A unit quaternion
$Q = [\![\ \cos\frac{\theta}{2}\ ,\ \ \hat{\mathbf{n}}\sin\frac{\theta}{2}\ ]\!] \equiv (\cos\frac{\theta}{2}, \hat{n}_x\sin\frac{\theta}{2}, \hat{n}_y\sin\frac{\theta}{2}, \hat{n}_z\sin\frac{\theta}{2}) \equiv (q_1, \hat{n}_xq_2, \hat{n}_xq_2, \hat{n}_xq_2)$ is used to rotate along $\hat{\mathbf{n}}$ (and dilate as $\mid Q \mid$ is not (yet) constrained to 1) the set $X$ to match $X'$. As the rotated $X$ does not exactly match $X'$, a quaternion can be associated with each residual (*i.e.* error) observed between the corresponding vectors in two sets as:

$$[\![\ 0\ ,\ \ \vec{\mathbf{e}}_i\ ]\!] = [\![\ 0\ ,\ \ \vec{\mathbf{x}}'_i\ ]\!] - Q^{-1}[\![\ 0\ ,\ \ \vec{\mathbf{x}}_i\ ]\!]Q, \qquad \forall 1 \leq i \leq n. \tag{2}$$

The rotation about $\hat{\mathbf{n}}$ associated with $Q$ is to be calculated such that the sum of squares of the above residuals, $\sum \mid \vec{\mathbf{e}}_i \mid^2 = \sum \mid [\![\ 0\ ,\ \ \vec{\mathbf{e}}_i\ ]\!] \mid^2$ is *minimised*. (Note: In this entire text, $\sum$ is over all $i \in (1, \cdots, n)$. In fact, the first and last residual can be dropped from the objective function above.)
To make the problem more algebraically elegant, multiply the equation 2 by $Q$.

$$Q[\![\ 0\ ,\ \ \vec{\mathbf{e}}_i\ ]\!] = Q[\![\ 0\ ,\ \ \vec{\mathbf{x}}'_i\ ]\!] - [\![\ 0\ ,\ \ \vec{\mathbf{x}}_i\ ]\!]Q, \qquad \forall i \in (1, \cdots, n). \tag{3}$$

Then, the equation 3 can be used to construct a slightly modified residual function $\xi$.

$$\begin{aligned}
\xi &= \sum \mid Q[\![\ 0\ ,\ \ \vec{\mathbf{e}}_i\ ]\!]\mid^2 \\
&= \sum \mid [\![\ -\vec{\mathbf{q}}\cdot\vec{\mathbf{e}}_i\ ,\ \ q_1\vec{\mathbf{e}}_i+\vec{\mathbf{q}}\times\vec{\mathbf{e}}_i\ ]\!]\mid^2 \\
&= \sum \mid [\![\ -(\hat{n}_x q_2 e_x^i+\hat{n}_y q_2 e_y^i+\hat{n}_z q_2 e_z^i)\ ,\ \ (q_1 e_x^i+\hat{n}_y q_2 e_z^i-\hat{n}_z q_2 e_y^i)\mathbf{i}+(q_1 e_y^i+\hat{n}_z q_2 e_x^i-\hat{n}_x q_2 e_z^i)\mathbf{j} \\
&\qquad +(q_1 e_z^i+\hat{n}_x q_2 e_y^i-\hat{n}_y q_2 e_x^i)\mathbf{k}\ ]\!]\mid^2, \quad (let\ \vec{\mathbf{e}}_i\equiv(e_x^i\mathbf{i}+e_y^i\mathbf{j}+e_z^i\mathbf{k}) \qquad (4) \\
&= \sum \Big((\hat{n}_x q_2 e_x^i+\hat{n}_y q_2 e_y^i+\hat{n}_z q_2 e_z^i)^2+(q_1 e_x^i+\hat{n}_y q_2 e_z^i-\hat{n}_z q_2 e_y^i)^2+(q_1 e_y^i+\hat{n}_z q_2 e_x^i-\hat{n}_x q_2 e_z^i)^2 \\
&\qquad +(q_1 e_z^i+\hat{n}_x q_2 e_y^i-\hat{n}_y q_2 e_x^i)^2\Big) \\
&= \sum (q_1 e_x^i)^2+(q_1 e_y^i)^2+(q_1 e_z^i)^2+(\hat{n}_x q_2 e_x^i)^2+(\hat{n}_x q_2 e_y^i)^2+(\hat{n}_x q_2 e_z^i)^2 \\
&\qquad +(\hat{n}_y q_2 e_x^i)^2+(\hat{n}_y q_2 e_y^i)^2+(\hat{n}_y q_2 e_z^i)^2+(\hat{n}_z q_2 e_x^i)^2+(\hat{n}_z q_2 e_y^i)^2+(\hat{n}_z q_2 e_z^i)^2 \\
&= \sum \mid Q\mid^2 \mid \vec{\mathbf{e}}_i\mid^2 \\
&= \mid Q\mid^2 \sum \mid \vec{\mathbf{e}}_i\mid^2 .
\end{aligned}$$

From above it is clear that the modified residual function reduces to the unmodified residual function ($\sum \mid \vec{\mathbf{e}}_i\mid^2$) when the $\mid Q\mid = 1$.

Again, from equation 3 we have

$$\xi = \sum \mid Q[\![\ 0\ ,\ \ \vec{\mathbf{x}}_i'\ ]\!] - [\![\ 0\ ,\ \ \vec{\mathbf{x}}_i\ ]\!]Q\mid^2$$

Expanding the above using quaternion product formula expressed in a combination of vector products we get,

$$\begin{aligned}
\xi &= \sum \mid [\![\ -\vec{\mathbf{q}}\cdot\vec{\mathbf{x}}_i'\ ,\ \ q_1\vec{\mathbf{x}}_i'+\vec{\mathbf{q}}\times\vec{\mathbf{x}}_i'\ ]\!] - [\![\ -\vec{\mathbf{x}}_i\cdot\vec{\mathbf{q}}\ ,\ \ q_1\vec{\mathbf{x}}_i+\vec{\mathbf{x}}_i\times\vec{\mathbf{q}}\ ]\!]\mid^2 \\
&= \sum \mid [\![\ -\vec{\mathbf{q}}\cdot\vec{\mathbf{x}}_i'+\vec{\mathbf{x}}_i\cdot\vec{\mathbf{q}}\ ,\ \ q_1\vec{\mathbf{x}}_i'+\vec{\mathbf{q}}\times\vec{\mathbf{x}}_i'+\vec{\mathbf{q}}\cdot\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i\cdot\vec{\mathbf{q}}\ ]\!]\mid^2 \\
&= \sum \mid [\![\ -\vec{\mathbf{q}}\cdot(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i)\ \ ,\ \ q_1(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i)+\vec{\mathbf{q}}\times(\vec{\mathbf{x}}_i'+\vec{\mathbf{x}}_i)\ \ ]\!]\mid^2, \quad (\because \vec{\mathbf{a}}\times\vec{\mathbf{b}}=-\vec{\mathbf{b}}\times\vec{\mathbf{a}}).
\end{aligned}$$

Let $A=-\vec{\mathbf{q}}\cdot(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i)$, and $B=q_1(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i)+\vec{\mathbf{q}}\times(\vec{\mathbf{x}}_i'+\vec{\mathbf{x}}_i)$.
Then expanding $A$ and $B$ using the respective components of $\vec{\mathbf{q}}$, $\vec{\mathbf{x}}_i$ and $\vec{\mathbf{x}}_i'$ we get,

$$\begin{aligned}
A &= -\vec{\mathbf{q}}\cdot(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i), \\
&= \hat{n}_x q_2(x_i'-x_i)+\hat{n}_y q_2(y_i'-y_i)+\hat{n}_z q_2(z_i'-z_i)
\end{aligned}$$

Similarly,

$$\begin{aligned}
B &= q_1(\vec{\mathbf{x}}_i'-\vec{\mathbf{x}}_i)+\vec{\mathbf{q}}\times(\vec{\mathbf{x}}_i'+\vec{\mathbf{x}}_i) \\
&= q_1(x_i'-x_i)\mathbf{i}+q_1(y_i'-y_i)\mathbf{j}+q_1(z_i'-z_i)\mathbf{k} \\
&\quad +(\hat{n}_y q_2(z_i'+z_i)-\hat{n}_z q_2(y_i'+y_i))\mathbf{i}+(\hat{n}_z q_2(x_i'+x_i)-\hat{n}_x q_2(z_i'+z_i))\mathbf{j} \\
&\qquad +(\hat{n}_x q_2(y_i'+y_i)-\hat{n}_y q_2(x_i'+x_i))\mathbf{k} \\
&= (q_1(x_i'-x_i)+\hat{n}_y q_2(z_i'+z_i)-\hat{n}_z q_2(y_i'+y_i))\mathbf{i} \\
&\qquad +(q_1(y_i'-y_i)+\hat{n}_z q_2(x_i'+x_i)-\hat{n}_x q_2(z_i'+z_i))\mathbf{j} \\
&\qquad\quad +(q_1(z_i'-z_i)+\hat{n}_x q_2(y_i'+y_i)-\hat{n}_y q_2(x_i'+x_i))\mathbf{k}
\end{aligned}$$

Since $\mid Q \mid^2 = q_1^2 + \vec{\mathbf{q}} \cdot \vec{\mathbf{q}}$, we get

$$\begin{aligned}\xi = \sum \quad & \Big( \big(\hat{n}_x q_2(x_i' - x_i) + \hat{n}_y q_2(y_i' - y_i) + \hat{n}_z q_2(z_i' - z_i)\big)^2 \\ & + \big(q_1(x_i' - x_i) + \hat{n}_y q_2(z_i' + z_i) - \hat{n}_z q_2(y_i' + y_i)\big)^2 \\ & + \big(q_1(y_i' - y_i) + \hat{n}_z q_2(x_i' + x_i) - \hat{n}_x q_2(z_i' + z_i)\big)^2 \\ & + \big(q_1(z_i' - z_i) + \hat{n}_x q_2(y_i' + y_i) - \hat{n}_y q_2(x_i' + x_i)\big)^2 \Big).\end{aligned}$$

To ensure pure rotations (without dilations) the $\mid Q \mid = \sqrt{(q_1^2 + \hat{n}_x q_2^2 + \hat{n}_y q_2^2 + \hat{n}_z q_2^2}$ is now constrained to unity and combined with $\xi$ by writing the *Lagrangian*:

$$\begin{aligned}\Lambda(Q,\lambda) = \sum \quad & \Big( \big(\hat{n}_x q_2(x_i' - x_i) + \hat{n}_y q_2(y_i' - y_i) + \hat{n}_z q_2(z_i' - z_i)\big)^2 \\ & + \big(q_1(x_i' - x_i) + \hat{n}_y q_2(z_i' + z_i) - \hat{n}_z q_2(y_i' + y_i)\big)^2 \\ & + \big(q_1(y_i' - y_i) + \hat{n}_z q_2(x_i' + x_i) - \hat{n}_x q_2(z_i' + z_i)\big)^2 \\ & + \big(q_1(z_i' - z_i) + \hat{n}_x q_2(y_i' + y_i) - \hat{n}_y q_2(x_i' + x_i)\big)^2 \Big) \\ & + \lambda(1 - q_1^2 - q_2^2(\hat{n}_x^2 + \hat{n}_y^2 + \hat{n}_z^2)) \end{aligned} \tag{5}$$

Let $M_x$ represent $(x_i' - x_i)$ (similarly $M_y$ and $M_z$), and $P_x$ represent $(x_i' + x_i)$ (similarly $P_y$ and $P_z$). Then the equation

$$\begin{aligned}\Lambda(Q,\lambda) = \sum \quad & \Big( (q_2(\hat{n}_x M_x + \hat{n}_y M_y + \hat{n}_z M_z))^2 \\ & + (q_1 M_x + q_2(\hat{n}_y P_z - \hat{n}_z P_y))^2 \\ & + (q_1 M_y + q_2(\hat{n}_z P_x - \hat{n}_x P_z))^2 \\ & + (q_1 M_z + q_2(\hat{n}_x P_y - \hat{n}_y P_x)^2 \\ & + \lambda(1 - q_1^2 - q_2^2) \qquad\qquad (\because n_x^2 + n_y^2 + n_z^2 = 1) \end{aligned} \tag{6}$$

Recovering the parallel normal equations by using partial derivatives with respect to the quaternion components we get,

$$\begin{aligned}\frac{\partial}{\partial q_1}\Lambda(Q,\lambda) &= \sum q_1(M_x^2 + M_y^2 + M_z^2) + \\ & \sum q_2\left(\hat{n}_x(P_y M_z - M_y P_z) - \hat{n}_y(P_x M_z - M_x P_z) + \hat{n}_z(P_x M_y - M_x P_y)\right) \\ & -\lambda q_1 = 0. \\ \frac{\partial}{\partial q_2}\Lambda(Q,\lambda) &= \sum q_1\left(\hat{n}_x(P_y M_z - M_y P_z) - \hat{n}_y(P_x M_z - M_x P_z) + \hat{n}_z(P_x M_y - M_x P_y)\right) + \\ & \sum q_2\left((\hat{n}_x M_x + \hat{n}_y M_y + \hat{n}_z M_z)^2 + (\hat{n}_x P_y - \hat{n}_y P_x)^2 + (\hat{n}_x P_z - \hat{n}_z P_x)^2 + (\hat{n}_y P_z - \hat{n}_z P_y)^2\right) \\ & -\lambda q_2 = 0.\end{aligned}$$

The above linear equations in quaternion variables $q_1$ and $q_2$ can be organised as an eigenvalue problem of the form:

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = \lambda \begin{pmatrix} q_1 \\ q_2 \end{pmatrix}.$$

where:
$A = \left(M_x^2 + M_y^2 + M_z^2\right)$,
$B = (\hat{n}_x(P_y M_z - M_y P_z) - \hat{n}_y(P_x M_z - M_x P_z) + \hat{n}_z(P_x M_y - M_x P_y))$ and
$C = \left((\hat{n}_x M_x + \hat{n}_y M_y + \hat{n}_z M_z)^2 + (\hat{n}_x P_y - \hat{n}_y P_x)^2 + (\hat{n}_x P_z - \hat{n}_z P_x)^2 + (\hat{n}_y P_z - \hat{n}_z P_y)^2\right)$

From above, the characteristic polynomial is:
$(A - \lambda)(C - \lambda) - B^2 = 0$
$\lambda^2 - (A + C)\lambda + (AC - B^2) = 0$

The roots of the above quadratic are real because:
$(A + C)^2 - 4(AC - B^2) = (A - C)^2 + B^2 \geq 0$

The smallest $\lambda$ corresponds to the rotation that minimizes the least-squares, from which the transformation follows.

## 2 Supporting figure for the Results section in the main text (See Section 6, last paragraph)

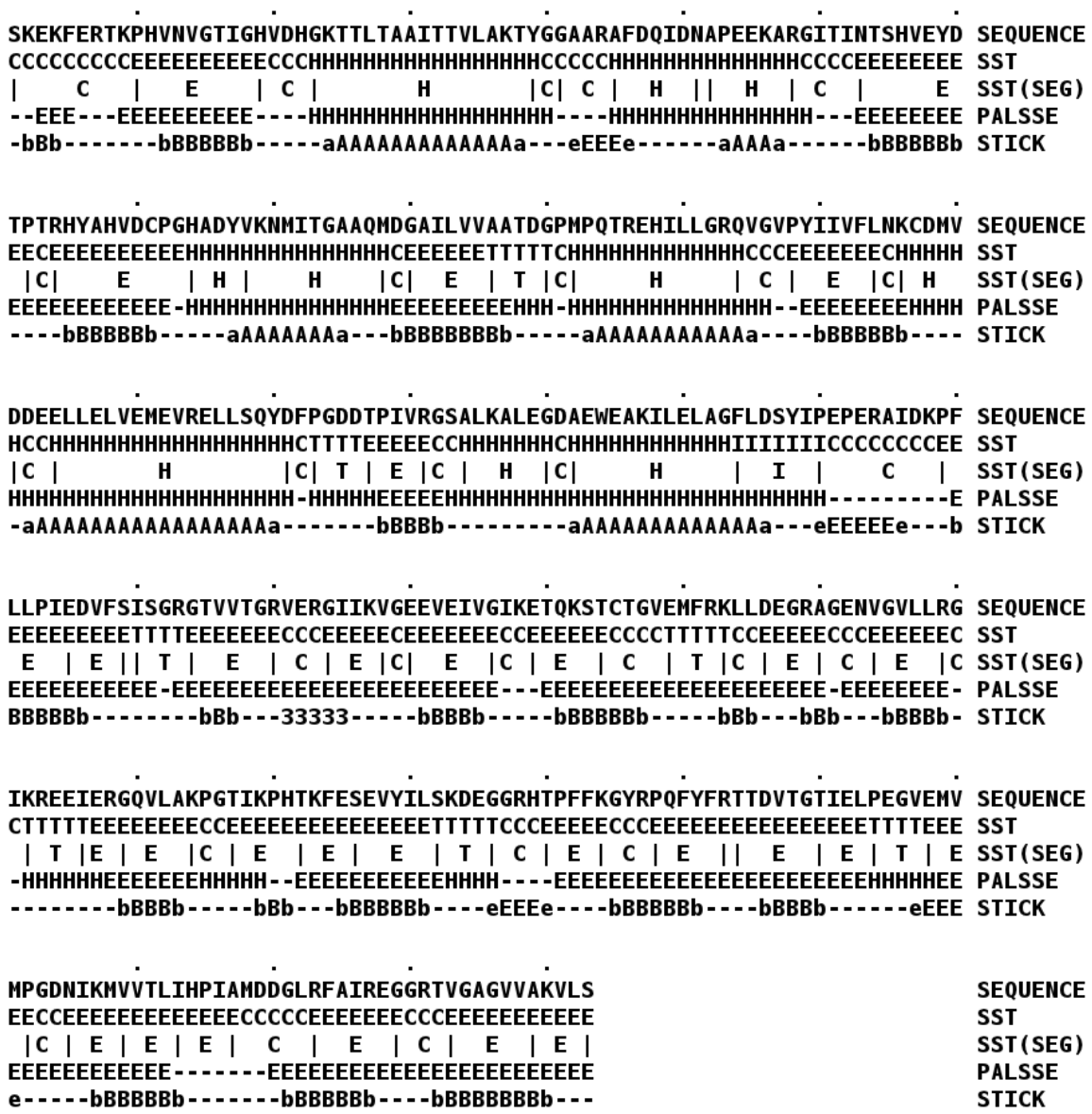


Figure 1: Residue-level secondary structure assignment of a 10 Å cryo-EM structure (wwPDB: 1QZD) using `SST`,`PALSSE`, and `STICK`. `DSSP`, `STRIDE` and `P-SEA` fail to process the coordinates. `KAKSI` and `XTLSSTR` assign the entire chain to Coil.