Statistical compression of protein folding patterns for inference of recurrent substructural themes

Ramanan Subramanian[1], Lloyd Allison[1], Peter J. Stuckey[2], Maria Garcia de la Banda[1], David Abramson[3], Arthur M. Lesk[4], and Arun S. Konagurthu[1,*]

[1] Faculty of Information Technology, Monash University, Clayton VIC 3800, Australia.
[2] Department of Computing and Information Systems, University of Melbourne, Parkville VIC 3010, Australia.
[3] Research Computing Centre, University of Queensland, St Lucia QLD 4072, Australia.
[4] Department of Biochemistry and Molecular Biology, Pennsylvania State University, University Park PA 16802, USA.
*Corresponding author: Arun S. Konagurthu (arun.konagurthu@monash.edu)

Abstract

Computational analyses of the growing corpus of three-dimensional (3D) structures of proteins have revealed a limited set of recurrent substructural themes, termed super-secondary structures. Knowledge of super-secondary structures is important for the study of protein evolution and for the modeling of proteins with unknown structures. Characterizing a comprehensive dictionary of these super-secondary structures has been an unanswered computational challenge in protein structural studies. This paper presents an unsupervised method for learning such a comprehensive dictionary using the statistical framework of lossless compression on a database comprised of concise geometric representations of protein 3D folding patterns. The best dictionary is defined as the one that yields the most compression of the database. Here we describe the inference methodology and the statistical models used to estimate the encoding lengths. An interactive website for this dictionary is available at http://lcb.infotech.monash.edu.au/proteinConcepts/scop100/dictionary.html.

1 Introduction

Proteins are functional biomolecules synthesised in the cells of living organisms and involved in almost all known fundamental processes of life [1]. Each protein consists of one or more unbranched chain(s), each comprised of a *sequence* of amino acids that, upon synthesis, spatially folds into a native three-dimensional (3D) *structure*. It is this 3D structure that determines the protein's function [2] and, thus, understanding the principles underlying protein structure is critically important for protein studies.

Over the past seventy years, steady advances in experimental methods to resolve protein structures to atomic resolution have resulted in a rich data stream collected into a publicly accessible worldwide Protein Data Bank (wwPDB) [3]. This database stores the 3D atomic coordinates of over 115,000 protein structures, and has enabled researchers to study the complex spatial organization of proteins and provide novel insights into the principles underpinning their architecture, function and evolution [2].

Nearly all protein structures contain recurrent (sub)structural patterns. The most basic of these are *helices* and extended *strands* of pleated sheets, termed the standard *secondary structural elements* of protein 3D structure [4]. See Fig. 1 for an example.

2375-0359/17 $31.00 © 2017 IEEE
DOI 10.1109/DCC.2017.46

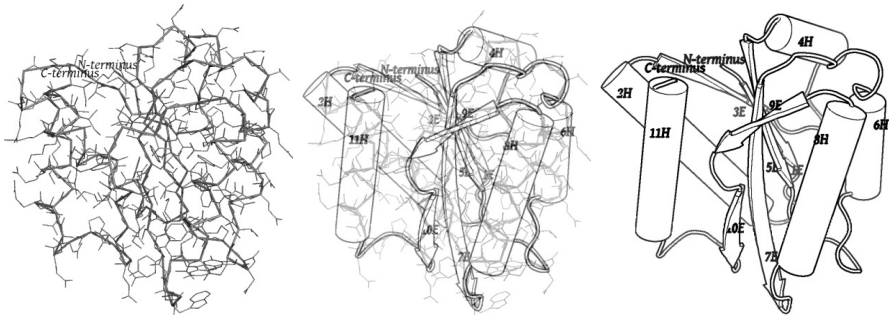

Figure 1: Left: Experimentally determined 3D structure of oxidised flavodoxin protein from the organism *Clostridium beijerinckii* (wwPDB ID:`5nll`). Middle: Assignment of helices (cylinders) and strands of sheet (arrows) to contiguous regions along the protein chain. Right: The folding pattern of `5nll` shown as the assembly of the assigned secondary structural elements.

.

The structureless sequence of a newly synthesised protein first locally folds into such secondary structures, which in turn assemble into more complex 3D shapes. Within the context of understanding proteins' architectural principles, the 3D shape or *fold* of a given protein can be compactly described by a *tableau* containing [5, 6]: (1) The sequence (or order) of standard secondary structural elements along the protein chain, (2) the interaction (or contact) between these elements in 3D space, and (3) the relative orientation (or geometry) of each pair of elements. See Fig. 2.

Studies of protein folding patterns have identified *super-secondary structures*, compact folding themes that recur, even across in unrelated proteins [7]. These comprise two or more interacting secondary structural elements that appear in succession in the protein chain, with a specific geometric arrangement [8]. Characterizing these super-secondary structural themes and their combinations is central to the classification of known protein structures and to the recognition of their evolutionary relationships. For instance, the most-widely used manually-curated database on structural classification (SCOP [9]) uses the geometry of super-secondary structures as a basis for its classification [10]. Furthermore, knowledge of super-secondary structures has been shown to be an effective starting point for modeling proteins with known sequence but unknown structural information [1, 11, 12].

Despite their importance, only a restricted set of super-secondary structures has been identified thus far. Further, the main identification method relies predominantly on visual inspection by human experts [11, 13]. Although some theoretical models have been proposed to systematically enumerate all physically-realizable super-secondary structural themes, a systematic enumeration is computationally prohibitive [14]. As a result, the problem of identifying a comprehensive dictionary of super-secondary structures has been an open computational problem [15].

This paper presents an unsupervised method, based on statistical inductive inference [16, 17] and lossless data compression, to learn a comprehensive dictionary of super-secondary structures. The inference is made on a source collection of tableaux representing the protein folding patterns of a non-redundant corpus of known protein 3D structures from SCOP [9]. The method automatically learns a static dictionary

of contiguous sub-tableaux that best compresses the source collection, exploiting the statistical redundancy in those source data. We shall call these sub-tableaux *concepts*. Thus, each concept inferred in the dictionary provides a geometric definition of the corresponding super-secondary structural theme.

The following methodological elements support this work: Section 2.2 describes the main statistical framework for inference. The details of the compression scheme and the probabilistic models employed to estimate the lengths of encoding of various terms supporting this work appear in Sections 2.3-2.5. Section 2.6 describes the heuristic search used to identify a static dictionary over all tableaux in the source collection. Section 3 explores some quantitative aspects of our experimental results, which include a rich super-secondary structural dictionary containing 4,487 concepts.

2 The Algorithm

2.1 Terminology and notations

Tableau representation and the source collection of tableaux: As mentioned before, the essence of any protein folding pattern can be captured by the order, relative orientations and interactions among its secondary structural elements [5, 6]. This has led to the concise two-dimensional *tableau* [6] representation of protein folding patterns (Fig. 2) that encapsulates: (1) the order in which helices and strands-of-sheet appear in the protein chain, represented by a string, $\mathbf{S}$, of length $|\mathbf{S}|$ over the $\{\text{H(for helix)}, \text{E(for strand)}\}$ alphabet; (2) the geometry of each pair of secondary structural elements, represented by a square-symmetric matrix of angles, $\mathbf{\Omega}$, of order $|\mathbf{S}| \times |\mathbf{S}|$, where angles are in the range $(-180°, 180°]$; (3) the corresponding interactions between pairs of secondary structural elements, represented by a contact matrix, $\mathbf{\Xi}$, of 0/1 values and order $|\mathbf{S}| \times |\mathbf{S}|$, where 1 represents contact and 0 otherwise. Formally, any tableau τ is a three-tuple of the form $(\mathbf{S}, \mathbf{\Omega}, \mathbf{\Xi})$. A *source collection* is a collection of (source) tableaux, denoted by the set $\mathcal{T} = \left\{\tau_1, \tau_2, \ldots, \tau_{|\mathcal{T}|}\right\}$.

Sub-tableaux and the dictionary of concepts: In this work, a super-secondary structure takes the form of a contiguous sub-tableau, referred to as *concept*. A concept, denoted by c, can be instantiated by selecting a source tableau $\tau_\ell \in \mathcal{T}$ and specifying a continuous range of indices $[i, i+1, \ldots, j-1, j]$, such that $1 \leq i < j \leq |\tau_\ell|$, and each secondary structural element in this range has at least one contact with other elements in it (see shaded cells in Fig. 2). Formally, any concept $c \in \tau_\ell^{[i \ldots j]}$ defines a three-tuple $\left(\mathbf{S}_{\tau_\ell}^{[i \ldots j]}, \mathbf{\Omega}_{\tau_\ell}^{[i \ldots j]}, \mathbf{\Xi}_{\tau_\ell}^{[i \ldots j]}\right) \subseteq \left(\mathbf{S}_{\tau_\ell}^{[1 \ldots |\tau_\ell|]}, \mathbf{\Omega}_{\tau_\ell}^{[1 \ldots |\tau_\ell|]}, \mathbf{\Xi}_{\tau_\ell}^{[1 \ldots |\tau_\ell|]}\right)$. We define a *dictionary* as a set of concepts, denoted by the set $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$. A dictionary can contain an arbitrary number of concepts ($|\mathcal{C}|$), with each concept $c_y \in \mathcal{C}$ containing an arbitrary number of secondary structural elements ($|c_y| \geq 2$). Any dictionary is a potential candidate to compress the source collection of tableaux, $\mathcal{T}$.

Associated with each concept $c_y \in \mathcal{C}$ is a concentration parameter, κ_y, corresponding to a von Mises circular (angular) probability distribution [18]. This parameter controls the assignment of probabilities used to estimate the encoding length of entries in $\mathbf{\Omega}$ when compressing regions of the source tableaux. That is, κ_y controls the

flexibility of an inferred concept. A smaller/larger κ_y yields greater/lesser flexibility of the concept's usages for compressing source tableaux regions. In this work, all $\{\kappa_y\}_{\forall 1 \leq y \leq |\mathcal{C}|}$ lie in the range $[\kappa_{\min} = 10, \kappa_{\max} = 100]$, with their precise value inferred (to a precision of $\epsilon_\kappa = 0.5$) as a part of the dictionary search (see Section 2.6).

1E	**174.0°**	**−18.9°**	**−136.9°**	**−21.9°**	**−139.3°**	**−31.6°**	127.9°	−94.7°	−44.2°	**135.2°**
174.0°	*2H*	**−166.5°**	39.6°	**152.3°**	45.6°	**142.6°**	−57.9°	79.5°	**130.9°**	**−50.4°**
−18.9°	**−166.5°**	*3E*	−151.9°	**−38.5°**	**127.4°**	−48.0°	112.0°	−109.8°	−56.0°	**120.3°**
−136.9°	39.6°	−151.9°	*4H*	**−125.8°**	**−57.0°**	118.7°	−76.5°	70.6°	120.8°	67.8°
−21.9°	**152.3°**	**−38.5°**	**−125.8°**	*5E*	**−158.7°**	**−9.7°**	**149.8°**	**−72.8°**	−24.6°	**157.0°**
−139.3°	45.6°	**127.4°**	**−57.0°**	**−158.7°**	*6H*	−164.1°	**−19.7°**	122.5°	176.5°	11.5°
−31.6°	**142.6°**	−48.0°	118.7°	**−9.7°**	−164.1°	*7E*	**159.5°**	**−63.1°**	**−18.4°**	**166.2°**
127.9°	−57.9°	112.0°	−76.5°	**149.8°**	**−19.7°**	**159.5°**	*8H*	**−137.4°**	**161.8°**	−8.8°
−94.7°	79.5°	−109.8°	70.6°	**−72.8°**	122.5°	**−63.1°**	**−137.4°**	*9E*	**54.3°**	**129.7°**
−44.2°	**130.9°**	−56.0°	120.8°	−24.6°	176.5°	**−18.4°**	**161.8°**	**54.3°**	*10E*	**−169.0°**
135.2°	**−50.4°**	**120.3°**	67.8°	**157.0°**	11.5°	**166.2°**	−8.8°	**129.7°**	**−169.0°**	*11H*

Figure 2: A tableau representation capturing the geometry of the folding pattern of the flavodoxin protein, `5nll`, shown in Fig. 1. Its three-tuple data $(\mathbf{S}, \mathbf{\Omega}, \mathbf{\Xi})$ is coalasced and presented as follows: the secondary structural string $\mathbf{S}$ is shown in the main diagonal as enumerated helices ('*H*'s) and strands ('*E*'s); the off-diagonal cells display the $\mathbf{\Omega}$ angles between secondary structural pairs. The contact information in $\mathbf{\Xi}$ is shown via bold (representing value 1) and non-bold (0) angles. The shaded region shows a candidate sub-tableau, $\tau^{[3\ldots5]}$, within the whole tableau, $\tau^{[1\ldots11]}$.

2.2 The framework of inference of the best static dictionary on source tableaux

Our goal is to learn the static dictionary $\mathcal{C}$ (i.e., hypothesis) that offers the best compression of the source collection $\mathcal{T}$ (i.e, observed data). The general statistical framework used to achieve this relies on the criterion of minimum message length (MML) inference [16, 17]. MML inference is best understood as a lossless two-part communication between an *imaginary transmitter-receiver pair*. In the first part the transmitter encodes and communicates the hypothesis to the receiver, while in the second part it communicates the observed data *given* the stated hypothesis. The best hypothesis in this framework is the one that yields the shortest two-part lossless message to communicate the observed data. Formally, for any static dictionary $\mathcal{C}$ and source collection $\mathcal{T}$, the two-part message length (in bits) is denoted by the terms:

$$\mathcal{I}(\mathcal{C}\&\mathcal{T}) = \underbrace{\mathcal{I}(\mathcal{C})}_{\text{first part}} + \underbrace{\mathcal{I}(\mathcal{T}|\mathcal{C})}_{\text{second part}} \qquad \text{bits}, \tag{1}$$

where $\mathcal{I}(\cdot) = -\log_2(\Pr(\cdot))$ is the Shannon's measure of information content [19].

The two-part message shown in Eqn. 1 is contrasted with the (single-part) *null model* message, that is, the encoding of the observed data *as is*, without the support of any hypothesis. The null model message length is denoted as $\mathcal{I}_{\text{null}}(\cdot)$. Thus, the quality of an inferred dictionary $\mathcal{C}$ is measured as the compression obtained by encoding the source collection $\mathcal{T}$ using $\mathcal{C}$, i.e., as $\mathcal{I}_{\text{null}}(\mathcal{T}) - \mathcal{I}(\mathcal{C}\&\mathcal{T})$. This yields an inference problem with the following objective: $\underset{\mathcal{C}}{\textbf{arg max}}\ \mathcal{I}_{\text{null}}(\mathcal{T}) - \mathcal{I}(\mathcal{C}\&\mathcal{T})$.

Addressing this inference problem requires the following: (1) A method to estimate the null model encoding length, $\mathcal{I}_{\text{null}}(\mathcal{T})$, for any given collection $\mathcal{T}$; (2) A method to estimate the dictionary model encoding length, $\mathcal{I}(\mathcal{C}\&\mathcal{T})$ for any given dictionary $\mathcal{C}$ and collection $\mathcal{T}$; (3) A search method for an *optimal* dictionary (one that maximizes compression, as per the stated objective). These methods are presented below.

2.3 Estimation of $\mathcal{I}_{null}(\mathcal{T})$.

The null encoding of the source collection $\mathcal{T}$ involves the encoding of the number of tableaux over an integer code, followed by the null encoding of each tableau $\tau_\ell \in \mathcal{T}$:

$$\mathcal{I}_{\text{null}}(\mathcal{T}) = I_{\text{integer}}\big(|\mathcal{T}|\big) + \sum_{\ell=1}^{|\mathcal{T}|} I_{\text{null}}\big(\tau_\ell\big) \qquad \text{bits}, \tag{2}$$

where $I_{\text{integer}}(.)$ is the message length of encoding any positive integer over a $\log^*$ distribution [20]. Further, the estimation of each $I_{\text{null}}\big(\tau_\ell\big)$ term is carried out by encoding the number of secondary structural elements using the same integer code, followed by encoding the three-tuples $(\mathbf{S}_{\tau_\ell}, \mathbf{\Omega}_{\tau_\ell}, \mathbf{\Xi}_{\tau_\ell})$ using uniform probability distributions on their respective supports. This implies that each character in the $\mathbf{S}_\ell$ string takes one bit to encode, each contact state in $\mathbf{\Xi}_\ell$ also takes one bit, and each angle in $\mathbf{\Omega}_\ell$, specified to a precision of 0.1° in the range $(-180^\circ, +180^\circ]$, takes $\log_2(360/0.1) = \log_2 3600$ bits. Thus, the null message length for communicating each tableau τ_ℓ is given by:

$$I_{\text{null}}\big(\tau_\ell\big) = \underbrace{I_{\text{integer}}\big(|\mathbf{S}_\ell|\big) + |\mathbf{S}_\ell|}_{\mathcal{I}_{\text{null}}(\mathbf{S}_\ell)} + \underbrace{\tbinom{|\mathbf{S}_\ell|}{2} \log_2(3600)}_{\mathcal{I}_{\text{null}}(\mathbf{\Omega}_\ell|\mathbf{S}_\ell)} + \underbrace{\tbinom{|\mathbf{S}_\ell|}{2}}_{\mathcal{I}_{\text{null}}(\mathbf{\Xi}_\ell|\mathbf{S}_\ell)} \qquad \text{bits}. \tag{3}$$

2.4 Estimation of the first part, $\mathcal{I}(\mathcal{C})$, term in Eqn. 1

Each concept c_y in any given dictionary is a (sub-)tableau. Therefore, c_y can be encoded using the null model as shown in Section 2.3, using $\mathcal{I}_{\text{null}}(c_y)$ bits. In addition, its associated κ_y parameter also needs to be encoded. As seen in Section 2.1, each κ_y lies in the range $[\kappa_{\min}, \kappa_{\max}]$ specified to a precision of ϵ_κ, and can be encoded using a uniform probability distribution over this support. Using these component terms, the resulting encoding length for the full dictionary takes:

$$\mathcal{I}(\mathcal{C}) = I_{\text{integer}}\big(|\mathcal{C}|\big) + \sum_{j=1}^{|\mathcal{C}|} I_{\text{null}}\big(c_j\big) + |\mathcal{C}| \log_2\big(\frac{\kappa_{\max} - \kappa_{\min}}{\epsilon_\kappa} + 1\big) \qquad \text{bits}. \tag{4}$$

2.5 Estimation of the second part, $\mathcal{I}(\mathcal{T}|\mathcal{C})$, term in Eqn. 1

The encoding length of the source collection $\mathcal{T}$ given the dictionary $\mathcal{C}$ is computed as the sum of the code lengths required to encode each tableau $\tau_\ell \in \mathcal{T}$ using $\mathcal{C}$. To compute this code length, $\mathcal{I}(\tau_\ell|\mathcal{C})$, each tableau τ_ℓ is *partitioned* into *non-overlapping* regions of variable sizes (see Fig. 3). Any *partition* of τ_ℓ, $p(\tau_\ell)$, is specified by an increasing sequence of integer indices $1 \equiv z_0 < z_1 < z_2 < \ldots < z_{|p(\tau_\ell)|} \equiv |\tau_\ell| + 1$. The set of possible 2-grams from this partition sequence, $\{\langle z_{k-1}, z_k \rangle\}_{\forall 1 \le k \le |p(\tau_\ell)|}$, define $|p(\tau_\ell)|$ consecutive, non-overlapping regions of τ_ℓ of the form $\tau_\ell^{[z_{k-1} \ldots z_k - 1]} \subseteq \tau_\ell$. The total number of possible partitions for a given τ_ℓ is $2^{|\tau_\ell|-1}$.

Assigned to each non-overlapping region $\tau_\ell^{[z_{k-1} \ldots z_k - 1]}$ in a specified partition $p(\tau_\ell)$ is one of the concepts $c_{y_k} \in \mathcal{C}$ (provided $\mathbf{S}_{\tau_\ell}^{[z_{k-1} \ldots z_k - 1]} \equiv \mathbf{S}_{c_{y_k}}$), where $1 \le y_k \le |\mathcal{C}|$, or a *null*

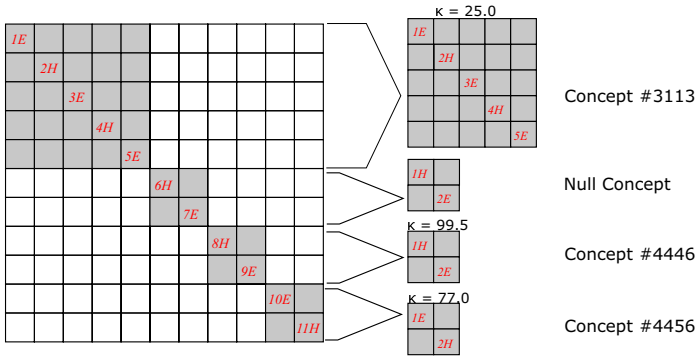

Figure 3: An illustration of a partition of a tableau.

concept c_0. The key idea here is that the burden of explanation of the data in each non-overlapping region $\tau_\ell^{[z_{k-1}\ldots z_k-1]} \subseteq \tau_\ell$ defined by $p(\tau_\ell)$ is borne by the corresponding data in its assigned concept. That is, the data in $\left(\mathbf{S}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}, \mathbf{\Omega}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}, \mathbf{\Xi}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}\right)$ is communicated using the corresponding data $\left(\mathbf{S}_{c_{y_k}}, \mathbf{\Omega}_{c_{y_k}}, \mathbf{\Xi}_{c_{y_k}}\right)$. When the region is assigned to the null concept c_0, the encoding of the data in that region is identical to null model encoding described in Section 2.3. In the remaining text, $p(\tau_\ell)$ specifies not only the non-overlapping regions, but also their corresponding concept assignments.

Thus, given $p(\tau_\ell)$, the data within each $\tau_\ell \in \mathcal{T}$ can be communicated as follows. First, the tableau size is communicated with an integer code using $I_{\text{integer}}\left(|\tau_\ell|\right)$ bits. Second, the details specified by $p(\tau_\ell)$ are communicated using $\mathcal{I}(p(\tau_\ell))$ bits. Third, each non-overlapping region $\tau_\ell^{[z_{k-1}\ldots z_k-1]}$ is explained by its assigned concept c_{y_k} (shown as the grey coloured regions in Fig. 3), using $\mathcal{I}(\tau_\ell^{[z_{k-1}\ldots z_k-1]}|c_{y_k})$ bits. Finally, the remaining data of τ_ℓ (the white coloured cells in Fig. 3) are communicated using the null model (Section 2.3). We denote this data (using set notations) as $(\tau_\ell|p(\tau_\ell))^{\complement} = \tau_\ell \setminus \cup_{k=1}^{|p(\tau_\ell)|}\{\tau_\ell^{[z_{k-1}\ldots z_k-1]}\}$, with code length denoted by $\mathcal{I}_{\text{null}}((\tau_\ell|p(\tau_\ell))^{\complement})$.

Based on the above details, the *shortest* lossless encoding length of any tableau $\tau_\ell \in \mathcal{T}$ given a static dictionary $\mathcal{C}$, is one that that minimizes the following objective:

$$\mathcal{I}(\tau_\ell|\mathcal{C}) = I_{\text{integer}}\left(|\tau_\ell|\right) + \min_{\forall p(\tau_\ell)}\left(\mathcal{I}(p(\tau_\ell)) + \sum_{k=1}^{|p(\tau_\ell)|}\mathcal{I}(\tau_\ell{}^{[z_{k-1}\ldots z_k-1]}|c_{y_k})\right) + \mathcal{I}_{\text{null}}((\tau_\ell|p(\tau_\ell))^{\complement}).$$

Combining the above term with Eqn. 4, gives us $\mathcal{I}(\mathcal{T}|\mathcal{C})$ as:

$$\mathcal{I}(\mathcal{T}|\mathcal{C}) = I_{\text{integer}}\left(|\mathcal{T}|\right) + \sum_{\ell=1}^{|\mathcal{T}|}\mathcal{I}(\tau_\ell|\mathcal{C}). \tag{5}$$

Below we describe the details of computing the code length terms involved in Eqn. 5.

Computation of $\mathcal{I}(p(\tau_\ell))$: A partition $p(\tau_\ell)$ is encoded as follows. Since the tableau size $|\tau_\ell|$ has already been communicated, the size of the partition $|p(\tau_\ell)|$ is encoded in $\log_2|\tau_\ell|$ bits. Each index in the corresponding set of concept assignments $\{y_k\}_{\forall 1\leq k\leq|p(\tau_\ell)|}$ can take the values $0 \leq y_k \leq |\mathcal{C}|$, and, thus, can be encoded in $\log_2(|\mathcal{C}|+1)$ bits. Given this information, the values of $z_0 = 1$, $z_{|p(\tau_\ell)|} = |\tau_\ell|+1$, and the subset of $\{z_k\}$'s $(2 \leq k < |p(\tau_\ell)|)$ associated with regions *not* assigned to the

null concept c_0 are already decipherable based on the assigned concept sizes. The remaining ones, associated with c_0, each take $\log_2(|\tau_\ell|-z_{k-1}+1)$ bits to state.

Computation of $\mathcal{I}(\tau_\ell^{[z_{k-1}\ldots z_k-1]}|c_{y_k})$: A region $\tau_\ell^{[z_{k-1}\ldots z_k-1]}$ can be encoded using a null concept c_0, or using any concept $c_{y_k} \in \mathcal{C}$. The computation of the null concept encoding of a region follows the same scheme as the null-model encoding of a tableau described in Section 2.3. On the other hand, the encoding of $\tau_\ell^{[z_{k-1}\ldots z_k-1]}$ using a concept $c_{y_k} \in \mathcal{C}$ is permitted only when the corresponding secondary structural strings are identical, and when the corresponding contact information between pairs of secondary structural elements differ in no more than 10% $\left(= \lfloor \frac{1}{10} \cdot \binom{|c_{y_k}|}{2} \rfloor\right)$ places.

The details of encoding $\tau_\ell^{[z_{k-1}\ldots z_k-1]} \equiv \left(\mathbf{S}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}, \mathbf{\Omega}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}, \mathbf{\Xi}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}\right)$ with $c_{y_k} \equiv \left(\mathbf{S}_{c_{y_k}}, \mathbf{\Omega}_{c_{y_k}}, \mathbf{\Xi}_{c_{y_k}}\right)$ are now considered. Since $\mathbf{S}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}$ is identical to $\mathbf{S}_{c_{y_k}}$, it is only necessary to encode $\mathbf{\Xi}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}$ and $\mathbf{\Omega}_{\tau_\ell}^{[z_{k-1}\ldots z_k-1]}$ using the corresponding concept's matrices $\mathbf{\Xi}_{c_{y_k}}$ and $\mathbf{\Omega}_{c_{y_k}}$, respectively. Thus, the encoding N_m requires $\log_2\left(1+\lfloor \frac{1}{10}\binom{|c_{y_k}|}{2} \rfloor\right)$ bits, where $N_\mathrm{m} \in [0, \lfloor \frac{1}{10}\binom{|c_{y_k}|}{2} \rfloor]$ is the total number of mismatches in this assigned region where $\mathbf{\Xi}_{c_{y_k}}$ and $\mathbf{\Xi}_{\tau_\ell}{}^{[z_{k-1}\ldots z_k-1]}$ differ. The locations of the mismatches are then encoded using a uniform distribution over the number of ways of identifying N_m locations out of $\binom{|c_{y_k}|}{2}$ cells. The resulting code length to identify each mismatched entry takes the logarithm of the corresponding binomial coefficient. Once this information is communicated, the corresponding entries of $\mathbf{\Omega}_{\tau_\ell}{}^{[z_{k-1}\ldots z_k-1]}$ are encoded using the null model, each using $\log_2 3600$ bits.

The matched angles are now transmitted. Each signed angle $\theta \in \mathbf{\Omega}_{\tau_\ell}{}^{[z_{k-1}\ldots z_k-1]}$ is encoded using the corresponding angle $\theta_\mu \in \mathbf{\Omega}_{c_{y_k}}$ and a $(90\%, 10\%)$ mixture model of von-Mises circular distribution [17] (parameterized on θ_μ and concept's κ) and a uniform distribution over the support $(-180°, +180°]$.

Computation of $\mathcal{I}_{\mathbf{null}}((\tau_\ell|p(\tau_\ell))^{\complement})$: This refers to the encoding of the off-diagonal $\mathbf{\Omega}_{\tau_\ell}$ and $\mathbf{\Xi}_{\tau_\ell}$ entries, shown as the white coloured cells in Fig. 3, and denoted here as $\mathbf{\Omega}_{(\tau_\ell|p(\tau_\ell))^{\complement}}$ and $\mathbf{\Xi}_{(\tau_\ell|p(\tau_\ell))^{\complement}}$. The total number of entries in this off-diagonal area in each of these matrices is $\binom{|\tau_\ell|}{2} - \sum_{k=1}^{|p(\tau_\ell)|}\binom{z_k-z_{k-1}}{2}$. Each angle in $\mathbf{\Omega}_{(\tau_\ell|p(\tau_\ell))^{\complement}}$ is encoded using the null model in $\log_2(3600)$ bits. Each contact in $\mathbf{\Xi}_{(\tau_\ell|p(\tau_\ell))^{\complement}}$ requires 1 bit.

Optimal partition of τ_ℓ given $\mathcal{C}$ via dynamic programming: The computation of $\mathcal{I}(\tau_\ell|\mathcal{C})$ is carried out on the optimal partition of τ_ℓ given the concepts in $\mathcal{C}$. The identification of the optimal $p(\tau_\ell)$, the one that minimizes $\mathcal{I}(\tau_\ell|\mathcal{C})$, is achieved using a one-dimensional dynamic programming algorithm, similar to the one devised in our earlier work for a *completely different* problem from the same domain [21]. The specific details of the recurrences are omitted here due to lack of space.

2.6 Searching for an optimal dictionary

Our goal is to address the problem of inferring an optimal static dictionary, i.e., one that minimizes the two-part message length given by Eqn. 1. Finding a provably

optimal dictionary is computationally intractable due to the enormous search space. Hence, a simulated annealing (SA) heuristic is devised to address this problem. Algorithms based on SA require an aperiodic irreducible Markov chain defined on a certain state space, and a *cooling schedule* to iteratively push the solution towards the optimum. In our case, the state space is the set of all possible dictionaries. The desired Markov chain is generated by defining a neighbourhood and the corresponding transition probabilities for every state $\mathcal{C}$. A local neighbourhood for every state is explored through the following perturbation primitives: (1) `Add concept:` Creates a concept randomly from the source collection and adds it to $\mathcal{C}$. (2) `Remove element:` Chooses a concept randomly from the dictionary and deletes it. (3) `Perturb concept length:` Chooses a concept randomly from the dictionary, and extends/shortens it, in reference to its original source. (4) `Perturb concept kappa:` Increments/decrements the current value of κ associated with a randomly chosen concept. (5) `Swap concept with usage:` Chooses a concept randomly from the dictionary, and swaps it with a region in the collection that is currently encoded by it. (Note, the usage swapped-in as the perturbed concept could weakly voilate the connectivity constraint in terms of its contact map, unless strict connectivity is also imposed on that chosen usage.)

At each iteration, one of the above five perturbations is chosen uniformly at random. The transition probability to the neighbour is computed as follows. If the two-part message length given by Eqn. 1 decreases, the transition probability to the perturbed state is 1. If the two-part message length increases by ΔI bits, the probability is $2^{-\Delta I/T}$, where T is the *temperature* parameter of the system controlled by the following cooling schedule: Start with a temperature of 5,000 and decrease it by a factor of 0.88. At each temperature step, perform 50,000 random perturbations unless the temperature is below 10, where the number of perturbations is increased to 500,000 per temperature step. When the temperature reaches below 0.1, the search stops and the current state of the dictionary is reported. We implemented this search in the `C++` programming language and parallelized it using `OpenMPI`. (For pseudocode, see `http://lcb.infotech.monash.edu.au/proteinConcepts/scop100/pseudocode.pdf`).

3 Results

This work compresses a *source collection* containing 51,368 tableaux constructed by applying our program, `SST` [21], to a non-redundant set of the protein structural coordinate files derived from the SCOP (v.2.05) database [9]. This set is non-redundant in the sense that no two structures have the same amino acid sequence.

The inference algorithm described in Section 2 resulted in a rich super-secondary structural dictionary with 4,487 inferred concepts (sub-tableaux). The largest concept contains 43 secondary structural elements, while the smallest contain only 2 elements. The complete inferred dictionary is available via an interactive website at `http://lcb.infotech.monash.edu.au/proteinConcepts/scop100/dictionary.html`. (See Fig. 4 for some examples.) A discussion of the biological implications of this expressive dictionary and its relationship with the limited set of concepts reported in the literature, which is beyond the scope and context of this manuscript, is under preparation.

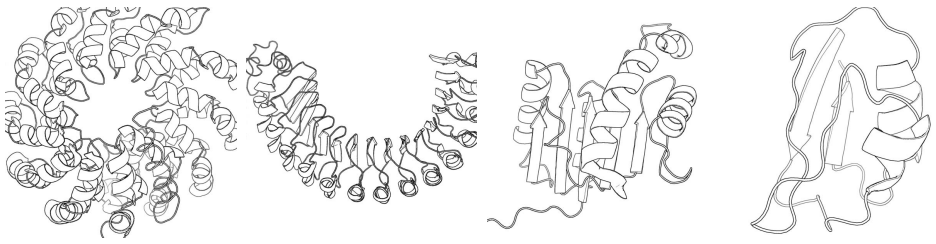

Figure 4: Four of the 4,487 concepts inferred in our dictionary, shown in the context of their source proteins structures, with Helices as ribbons, and strands as arrows.These correspond to concepts IDs 1, 9, 1378, 3623 respectively. For details visit dictionary web link in the previous paragraph.

The null model encoding length of our source collection (computed as per Eqn. 2) is 76,688,552 bits. The two-part message length for describing the same collection using the inferred dictionary (computed as per Eqn. 1) is 70,512,544 bits, where the first part takes 2,622,960 bits, and second part takes 67,889,584 bits. The resulting compression is 6,176,008 bits (or 8.05%) over the null model.

To show the dynamics of the search heuristic described in Section 2.6 and the evolution of the final dictionary: Fig. 5(a) shows the variation of the dictionary size $|\mathcal{C}|$ as a function of number of iterations in the simulated annealing search; Fig. 5(b) shows the complexity ($\mathcal{I}(\mathcal{C})$) of the first part of the two-part message; Fig. 5(c) combines the evolution of the $\mathcal{I}_{\text{null}}(\mathcal{T})$, $\mathcal{I}(\mathcal{T}|\mathcal{C})$ and $\mathcal{I}(\mathcal{C}\&\mathcal{T})$ terms over the search; and Fig. 5(d) shows the percentage compression gained over the null model, as the inferred dictionary evolves. These plots confirm the trade-off achieved by the MML inference, between the complexity of hypothesis (here, the dictionary of concepts) and the fit of this hypothesis to the data (here, the source collection of tableaux).

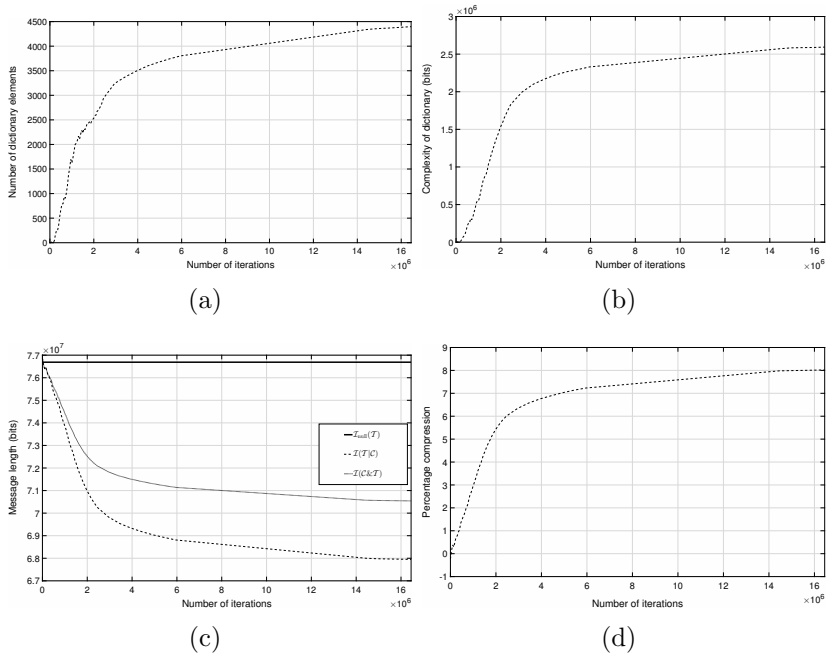

Figure 5: Dynamics of various terms during the simulated annealing search.

Acknowledgements: Authors acknowledge funding from Australian Research Council (DP150100894) and University of Queensland (RCC) computing facilities.

4 References

[1] P. Huang, S. E Boyken, and D. Baker, "The coming of age of de novo protein design," *Nature*, vol. 537, no. 7620, pp. 320–327, 2016.

[2] A. M. Lesk, *Introduction to Protein Science: Architecture, Function, and Genomics*, Oxford university press, 2010.

[3] H. Berman, K. Henrick, and H. Nakamura, "Announcing the worldwide Protein Data Bank," *Nature Structural & Molecular Biology*, vol. 10, no. 12, pp. 980–980, 2003.

[4] K. U. Linderstrøm-Lang, "Proteins and enzymes," Lane Medical Lectures, Stanford University, 1952.

[5] C. Chothia and A. V. Finkelstein, "The classification and origins of protein folding patterns," *Annual Review of Biochemistry*, vol. 59, no. 1, pp. 1007–1035, 1990.

[6] A. M. Lesk, "Systematic representation of protein folding patterns," *Journal of Molecular Graphics*, vol. 13, no. 3, pp. 159–164, 1995.

[7] S. T. Rao and M. G. Rossmann, "Comparison of super-secondary structures in proteins," *Journal of Molecular Biology*, vol. 76, no. 2, pp. 241–256, 1973.

[8] C. Chothia and M. Levitt, "Structural patterns in globular proteins," *Nature*, vol. 261, pp. 552–558, 1976.

[9] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures," *Journal of Molecular Biology*, vol. 247, no. 4, pp. 536–540, 1995.

[10] A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey, and A. M. Lesk, "MUSTANG: a multiple structural alignment algorithm," *Proteins: Structure, Function, and Bioinformatics*, vol. 64, no. 3, pp. 559–574, 2006.

[11] A. V. Efimov, "Super-secondary structures and modeling of protein folds," *Protein Supersecondary Structures*, pp. 177–189, 2013.

[12] D. Baker, "A surprising simplicity to protein folding," *Nature*, vol. 405, no. 6782, pp. 39–42, 2000.

[13] A. E. Kister, *Protein Supersecondary Structures*, Humana Press, 2013.

[14] B. Chitturi, S. Shi, L. N. Kinch, and N. V. Grishin, "Compact structure patterns in proteins," *Journal of Molecular Biology*, vol. (in press), 2016.

[15] A. M. Lesk and G. D. Rose, "Folding units in globular proteins," *Proceedings of the National Academy of Sciences*, vol. 78, no. 7, pp. 4304–4308, 1981.

[16] C. S. Wallace and D. M. Boulton, "An information measure for classification," *Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.

[17] C. S. Wallace, *Statistical and Inductive Inference using Minimum Message Length*, Information Science and Statistics. SpringerVerlag, 2005.

[18] K. V. Mardia and P. E. Jupp, *Directional statistics*, vol. 494, John Wiley & Sons, 2009.

[19] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 1948.

[20] C. S. Wallace and J. D. Patrick, "Coding decision trees," *Machine Learning*, vol. 11, no. 1, pp. 7–22, 1993.

[21] A. S. Konagurthu, A. M. Lesk, and L. Allison, "Minimum message length inference of secondary structure from protein coordinate data," *Bioinformatics*, vol. 28, no. 12, pp. i97–i105, 2012.