Supplementary Notes for:

The divergence time of protein structures modelled by Markov matrices and its relation to the divergence of sequences

Sandun Rajapaksa,[1] Lloyd Allison,[1] Peter J. Stuckey,[1,2] Maria Garcia de la Banda,[1,2] and Arun S. Konagurthu [1,*]

[1] Department of Data Science and Artificial Intelligence, Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia [2] OPTIMA ARC Industrial Training and Transformation Centre

S1 Introduction to MML-based parameter estimation

Strict Minimum Message Length (SMML) inference is an information theoretic criterion introduced by Wallace and Boulton [1975]. It follows the MML principle strictly in which, the data space is partitioned to nominate a representative model for each partition to minimize the expected two-part message length. An approximation of the SMML scheme named MML87 was introduced by Wallace and Freeman [1987]. The following section describes how MML87 is used for estimating multiple continuous parameters, which forms the main methodology of parameter inference and the estimation of encoding lengths (given in Equation 2-4 in the main text) to achieve the tasks stated in the main text.

Define $\vec{\theta} = [\theta_1, \theta_2, ..., \theta_n]$ with its prior probability distribution as $h(\vec{\theta})$, likelihood as $f(D|\vec{\theta})$ and negative likelihood as $\mathcal{L}(\vec{\theta}) = -\log[f(D|\vec{\theta})]$. The area under the curve of $h(\theta)$ for a single parameter θ is the probability of the region of uncertainty. Similarly, for multiple parameters, the probability of the region of uncertainty is a volume V. Let accuracy of parameter value $A_o PV = V_\theta$. Then the total transmission message length is as follows.

$$I(\theta, D) = I(\theta) + I(D|\theta) = -\log[h(\theta).V_\theta] + I(D|\theta) \tag{1}$$

This can be further expanded as follows. (For a detailed step-by-step computation see [Wallace and Wallace, 2005].)

$$I(\vec{\theta}, D) = \frac{d}{2}\log[c_d] - \log[h(\vec{\theta})] + \frac{1}{2}\log[det[F(\theta)]] + \mathcal{L}(\vec{\theta}) + \frac{d}{2} \tag{2}$$

where d is the number of dimensions, c_d is the Conway constant [Conway and Sloane, 1984], and $det[F(\theta)]$ is the determinant of the *expected* Fisher– a matrix of second order partial derivatives of the negative log-likelihood function.

Below, we will first provide an introduction to Dirichlet priors and the next section will explain how Dirichlet priors can be incorporated for estimation of the encoding lengths in Equation 2 for a set of parameters.

S1.1 Dirichlet distribution

Dirichlet distributions are used in this work to losslessly encode each column vector of $\mathbf{M}$ and the transition probabilities of a 3-state machine (over $\{\texttt{match}, \texttt{insert}, \texttt{delete}\}$ states). It is the multivariate generalization of the Beta distribution and a commonly used continuous probability distribution family for prior modeling. The following briefly describes the associated statistics.

Let Dir($\vec{\alpha}$) be a dirichlet distribution with model parameters $\vec{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_d]$ (for $\alpha_i > 0$) that describes a data sample $\vec{\Theta} = [\theta_1, \theta_2, \ldots, \theta_d]$, representing a point in the $d-1$ standard unit simplex (i.e. $\sum_{i=1}^{d} \theta_i = 1$). The following intuitive reparameterization of $\vec{\alpha}$ extends the insight on how the distribution concentrates with a concentration parameter κ around its mean vector $\vec{\mu}$ on the $d-1$ simplex.

$$\vec{\alpha} = \underbrace{\left(\sum_{i=1}^{d} \alpha_i\right)}_{\kappa} \underbrace{\left[\frac{\alpha_1}{\sum_{i=1}^{d}\alpha_i}, \frac{\alpha_2}{\sum_{i=1}^{d}\alpha_i}, \ldots, \frac{\alpha_d}{\sum_{i=1}^{d}\alpha_i}\right]}_{\vec{\mu}}$$

The mode vector $[x_1, x_2, \ldots, x_d]$ of Dir($\vec{\alpha}$) is defined by $x_i = \frac{\alpha_i - 1}{\kappa - d}$. The probability density function (pdf) $f(\vec{\Theta}|\vec{\alpha})$ of Dir($\vec{\alpha}$) is defined as:

$$f(\vec{\Theta}|\vec{\alpha}) = \frac{1}{B(\vec{\alpha})} \prod_{i=1}^{d} \theta_i^{\alpha_i - 1}$$

where $B(\vec{\alpha})$ is the multivariate form of the Beta function. The likelihood over data $\mathcal{D}$ with N data samples $\mathbf{\Theta} = [\vec{\Theta}_1, \vec{\Theta}_2, \ldots, \vec{\Theta}_{\mathrm{N}}]$ is defined as

$$F(\mathcal{D}|\vec{\alpha}) = \prod_{n=1}^{\mathrm{N}} f(\vec{\Theta}^n|\vec{\alpha})$$

Thus, the negative log likelihood function is:

$$\mathcal{L}(\mathbf{\Theta}|\vec{\alpha}) = -\mathrm{N}\log\Gamma(\kappa) + \mathrm{N}\sum_{i=1}^{d}\log\Gamma(\alpha_i) - \sum_{n=1}^{\mathrm{N}}\sum_{i=1}^{d}(\alpha_i - 1)\log\theta_{n,i}$$

The determinant of the Fisher matrix which indicates how sensitive the expected negative log-likelihood function is, to the changes of $\vec{\alpha}$ [Allison, 2018] is:

$$det[F(\vec{\alpha})] = \mathrm{N}^d\{\prod_{i=1}^{d}\psi(\alpha_i)\}\{1 - \psi_i(\kappa)\left(\sum_{i=1}^{d}\frac{1}{\psi_1(\alpha_i)}\right)\} \tag{3}$$

where $\psi_1(.)$ is the poly gamma function of order 1 (trigamma function).

S1.1.1 Encoding a probability vector $\vec{\Theta}$ using Dir($\vec{\alpha}$)

The encoding length of stating $\vec{\alpha}$ and $\vec{\Theta}$ is given by $I(\vec{\alpha}, \vec{\Theta}) = I(\vec{\alpha}) + I(\vec{\Theta}|\vec{\alpha})$ (see equation 1 and 2) where,

$$I(\vec{\alpha}) = \frac{d}{2}\log[c_d] - \log[h(\vec{\alpha})] + \frac{1}{2}\log[det[F(\vec{\alpha})]] \tag{4}$$

Here $h(\vec{\alpha})$ is the prior on dirichlet parameters $\vec{\alpha}$, and c_d is the lattice constant [Conway and Sloane, 1984] associated with d degrees of freedom. $det[F(\vec{\alpha})]$ given by Equation 3 is the determinant of the expected Fisher of $\vec{\alpha}$.

$$I(\vec{\Theta}|\vec{\alpha}) = \frac{d-1}{2}\log[c_{d-1}] - \log[\mathrm{Dir}(\vec{\Theta};\vec{\alpha})] + \frac{1}{2}\log[det[F(\vec{\Theta})]] + \frac{d-1}{2} \tag{5}$$

where $\mathrm{Dir}(\vec{\Theta};\vec{\alpha})$ is the Dirichlet prior on $\vec{\Theta}$, c_{d-1} is the lattice constant and

$$det[F(\vec{\Theta})] = \frac{\left(\sum_{j=1}^{d}\mathrm{count}(x_j)\right)^{d-1}}{\prod_{j=1}^{d}\theta_j} \tag{6}$$

Given the state frequency vector observed in $\mathcal{D}$ as $\{x_1, x_2, \ldots x_d\}$, then $\sum_{j=1}^{d}\mathrm{count}(x_j)$ is the total number of observations combining all states.

S2 Computation of each term in Equation 2 and 3 in the main text

S2.1 Computation of $I(\mathbf{M})$

$\mathbf{M}$ is a time parameterized stochastic Markov matrix over K secondary structural states of size 3×3 accounting for the secondary structure categories: $\{\mathtt{H}, \mathtt{E}, \mathtt{C}\}$. Each column vector $\vec{v_j}$ in $\mathbf{M}$ is an $\mathbb{L}_1$-normalized (i.e, $\sum_{j=1}^{K}\mathbf{M}_{i,j} = 1$) K-state probability vector in a unit $(K-1)$-simplex. The encoding length of $\mathbf{M}$ at any time $t^{\mathrm{3D}\mapsto\mathrm{2D}}$ can be derived using Equation 5 by assuming a uniform prior $h(\vec{v_j}) = \frac{(K-1)!}{\sqrt{K}}$. Hence,

$$I(\mathbf{M}) = \sum_{j=1}^{K}\frac{K-1}{2}\log(c_{K-1}) - \log[h(\vec{v_j})] + \frac{1}{2}\log\left(\frac{X_j^{K-1}}{\prod_{\forall p\in\vec{v}_j}p}\right) \tag{7}$$

where X_j is the total count of secondary structure transitions represented by $\vec{v}_j$ in the benchmark dataset $\mathbf{D}^{\mathrm{3D}\mapsto\mathrm{2D}}$, and p is the conditional probability of a secondary structural state indexed by j changing to another secondary structure state in $\vec{v_j}$.

S2.2 Computation of $I(\boldsymbol{\alpha})$, $I(\vec{\Theta}_i|\boldsymbol{\alpha}(t_i^{\mathrm{3D}\mapsto\mathrm{2D}}))$ and $I(\mathcal{A}_i^{\mathrm{3D}\mapsto\mathrm{2D}}|\vec{\Theta}_i)$

Our framework involves the estimation of two Dirichlet priors: $\mathtt{Dir}_{\mathtt{match}}(\vec{\alpha}_{\mathtt{match}})$ and $\mathtt{Dir}_{\mathtt{insert}}(\vec{\alpha}_{\mathtt{insert}}) \equiv \mathtt{Dir}_{\mathtt{delete}}(\vec{\alpha}_{\mathtt{delete}})$ for `match`, `insert` and `delete` states respectively. $\mathtt{Dir}_{\mathtt{match}}(\vec{\alpha}_{\mathtt{match}})$ describes a data sample

$$\vec{\Theta}_{\mathtt{match}} = \{\Pr(\mathtt{match}|\mathtt{match}), (1 - \Pr(\mathtt{match}|\mathtt{match}))\}$$

representing a point in 1-simplex, and $\mathtt{Dir}_{\mathtt{insert}}(\vec{\alpha}_{\mathtt{insert}})$ describes a data sample

$$\vec{\Theta}_{\mathtt{insert}} = \{\Pr(\mathtt{insert}|\mathtt{insert}), \Pr(\mathtt{match}|\mathtt{insert}), (1 - \Pr(\mathtt{insert}|\mathtt{insert}) - \Pr(\mathtt{match}|\mathtt{insert}))\}$$

having $d = 2$ degrees of freedom. The remaining 6 transition probabilities $\{\Pr(\mathtt{insert}|\mathtt{match}), \Pr(\mathtt{delete}|\mathtt{match}), \Pr(\mathtt{delete}|\mathtt{insert}), \Pr(\mathtt{match}|\mathtt{delete}), \Pr(\mathtt{insert}|\mathtt{delete}), \Pr(\mathtt{delete}|\mathtt{delete})\}$ of the alignment 3-state machine can be derived from the symmetry between `insert`, `delete` states and the constraint that all transition probabilities out of any state add up to 1.

The goal is to infer time-dependant optimal estimates of $\boldsymbol{\alpha} = \{\vec{\alpha}_{\texttt{match}}, \vec{\alpha}_{\texttt{insert}} \equiv \vec{\alpha}_{\texttt{delete}}\}$ that minimizes the total message length. First, we grouped each alignment in the benchmark dataset $\mathbf{D}^{\text{3D}\mapsto\text{2D}}$ into discrete time bins in the range $t^{\text{3D}\mapsto\text{2D}} \in [1, 250]$. The subset of the alignments grouped into $t^{\text{3D}\mapsto\text{2D}}$ time bin is denoted as $\mathbf{A}(t^{\text{3D}\mapsto\text{2D}})$ in the subsequent equations. Since the 1-simplex Dirichlet ($\texttt{Dir}_{\texttt{match}}(\vec{\alpha}_{\texttt{match}})$) and the 2-simplex Dirichlet ($\texttt{Dir}_{\texttt{insert}}(\vec{\alpha}_{\texttt{insert}})$) models are independent of each other, this minimization can be carried out separately. Below presents the general total message length formulation which communicates $\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})$, $\vec{\Theta}_{(\texttt{x})}$ and $\mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})$ jointly for any state $\texttt{x} \in \{\texttt{match}, \texttt{insert}\}$ using any evolutionary time bin $t^{\text{3D}\mapsto\text{2D}}$.

$$I(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}), \vec{\Theta}_{(\texttt{x})}, \mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})) = I(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})) + I(\vec{\Theta}_{(\texttt{x})}|\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})) + I(\mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})|\vec{\Theta}_{(\texttt{x})}) \tag{8}$$

The right-hand side terms of Equation 8 can be further expanded using the methods of estimation detailed in section S1.1.1 as follows.

$$I(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})) = \frac{d}{2}\log[c_d] - \log[h(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}))] + \frac{1}{2}\log[det[F(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}))]] \tag{9}$$

Here $h(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}))$ is the prior on dirichlet parameters $\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})$. See [Sumanaweera et al., 2019] for details on the prior used in the inference process. c_d is the lattice constant [Conway and Sloane, 1984] associated with d degrees of freedom, where $c_d = \frac{5}{36\sqrt{3}}$ for $\vec{\alpha}_{\texttt{match}}$ with $d = 2$ degrees of freedom and $c_d = \frac{19}{192\sqrt[3]{2}}$ for $\vec{\alpha}_{\texttt{insert}}$ with $d = 3$ degrees of freedom. $det[F(\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}))]$ given by Equation 3 is the determinant of the expected Fisher of $\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})$.

Note: The total message length of encoding the time-dependant Dirichlet parameter $\boldsymbol{\alpha}$ can be computed using Equation 9 as follows.

$$I(\boldsymbol{\alpha}) = \sum_{\forall t^{\text{3D}\mapsto\text{2D}} \in [1,250]} I(\vec{\alpha}_{\texttt{match}}(t^{\text{3D}\mapsto\text{2D}})) + I(\vec{\alpha}_{\texttt{insert}}(t^{\text{3D}\mapsto\text{2D}}))$$

The second term $I(\vec{\Theta}_{(\texttt{x})}|\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}}))$ in Equation 8 can be expanded as:

$$I(\vec{\Theta}_{(\texttt{x})}|\vec{\alpha}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})) = \left\{ \sum_{i=1}^{|\mathbf{A}(t^{\text{3D}\mapsto\text{2D}})|} \frac{1}{2}\log\left(1 + \frac{det[F(\vec{\Theta}_{i(\texttt{x})})](c_{d-1})^{d-1}}{f(\vec{\Theta}_{i(\texttt{x})}|\vec{\alpha}_{\texttt{x}})^2}\right) \right\} + \frac{d}{2} \tag{10}$$

where the determinant of the expected Fisher information is given by Equation 6. Note for $\texttt{x} = \texttt{match}$, the term $\text{count}(x_j)$ in Equation 6 refers to any one of $\texttt{match} \rightarrow \texttt{match}, \texttt{match} \rightarrow \texttt{insert}$, and $\texttt{match} \rightarrow \texttt{delete}$ transitions while the denominator refers to the product of all their probabilities: $\{\Pr(\texttt{match}|\texttt{match}), (1 - \Pr(\texttt{match}|\texttt{match}))\}$. Similarly, for $\texttt{x} = \texttt{insert}$, the term $\text{count}(x_j)$ refers to any one of $\texttt{insert} \rightarrow \texttt{match}, \texttt{insert} \rightarrow \texttt{insert}, \texttt{insert} \rightarrow \texttt{delete}, \texttt{delete} \rightarrow \texttt{match}, \texttt{delete} \rightarrow \texttt{insert}, \texttt{delete} \rightarrow \texttt{delete}$ transitions and the denominator refers to the product of all their probabilities: $\{\Pr(\texttt{insert}|\texttt{insert}), \Pr(\texttt{match}|\texttt{insert}), (1 - \Pr(\texttt{insert}|\texttt{insert}) - \Pr(\texttt{match}|\texttt{insert}))\}$.

Each $\theta_j \in \vec{\Theta}_{i(\texttt{x})}$ can be computed using MML87 estimation [Wallace and Wallace, 2005] as folows.

$$\theta_j = \frac{\text{count}(x_j) + \alpha_{\texttt{x},j} - \frac{1}{2}}{\sum_{l=1}^{d}\text{count}(x_l) + \kappa_{\texttt{x}} - \frac{d}{2}}$$

The last term $I(\mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})|\vec{\Theta}_{(\texttt{x})})$ deals with transmitting the 3-state string $\mathcal{A}_i \in \mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})$ using the parameter $\vec{\Theta}_{(\texttt{x})}$. Accordingly,

$$I(\mathbf{A}_{\texttt{x}}(t^{\text{3D}\mapsto\text{2D}})|\vec{\Theta}_{(\texttt{x})}) = \sum_{i=1}^{d}(x_i \times -\log(\theta_i))$$

where d is the number of state parameters; $\theta_i \in \{\vec{\Theta}_{i(\texttt{x})}, 1 - \sum_{i=1}^{d-1}\theta_i\}$ and x_i is the number of state transitions observed in the alignment $\mathcal{A}_i$. Note: $\mathbf{A}_{\texttt{match}}(t^{\text{3D}\mapsto\text{2D}})$ contains all instances of $\texttt{match} \rightarrow \texttt{match}, \texttt{match} \rightarrow \texttt{insert}$, and $\texttt{match} \rightarrow \texttt{delete}$ transitions in the set of alignments $\mathbf{A}(t^{\text{3D}\mapsto\text{2D}})$ for the case of $\texttt{x} = \texttt{match}$ and $\mathbf{A}_{\texttt{insert}}(t^{\text{3D}\mapsto\text{2D}})$ contains all instances of $\texttt{insert} \rightarrow \texttt{match}, \texttt{insert} \rightarrow \texttt{insert}, \texttt{insert} \rightarrow \texttt{delete}, \texttt{delete} \rightarrow \texttt{match}, \texttt{delete} \rightarrow \texttt{insert}$, and $\texttt{delete} \rightarrow \texttt{delete}$ transitions for the case of $\texttt{x} = \texttt{insert}$. The term $I(\mathcal{A}_i^{\text{3D}\mapsto\text{2D}}|\vec{\Theta}_i)$ in Equation 3 of the main text refers to the statement of the 3-state string only for a single alignment $\mathbf{A}_i^{\text{3D}\mapsto\text{2D}}$ using $\vec{\Theta}_i$ parameter.

S2.3 Computation of $I(t_i^{\text{3D}\mapsto\text{2D}})$

This computes the statement length of the optimal evolutionary time parameter $t_i^{\text{3D}\mapsto\text{2D}} \in \{t_1^{\text{3D}\mapsto\text{2D}}, t_2^{\text{3D}\mapsto\text{2D}}, \ldots, t_{|\mathbf{D}|}^{\text{3D}\mapsto\text{2D}}\}$, inferred for each pair $\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle$ in $\mathbf{D}^{\text{3D}\mapsto\text{2D}}$. We search over the integral values of $t_i^{\text{3D}\mapsto\text{2D}} \in [1, 250]$, where in each iteration the objective in Equation 6 in main text is optimized to find the best $t_i^{\text{3D}\mapsto\text{2D}}$ given an alignment $\mathcal{A}_i^{\text{3D}\mapsto\text{2D}}$. The statement of $I(t_i^{\text{3D}\mapsto\text{2D}})$ is considered to be uniform. Hence $I(t_i^{\text{3D}\mapsto\text{2D}}) = \log(t_{\text{max}})$ where $t_{\text{max}} = 250$.

S2.4 Computation of $I(\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle | \mathcal{A}_i^{\text{3D}\mapsto\text{2D}}, \mathbf{M}(t_i^{\text{3D}\mapsto\text{2D}}))$

Consider a pair of proteins $\langle S_i, T_i\rangle$ and their secondary structure information $\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle$, where $S_i^{\text{2D}} = \{s_1, s_2, \ldots s_{|S_i^{\text{2D}}|}\}$ and $T_i^{\text{2D}} = \{r_1, r_2, \ldots r_{|T_i^{\text{2D}}|}\}$. Then, $I(\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle | \mathcal{A}_i^{\text{3D}\mapsto\text{2D}}, \mathbf{M}(t_i^{\text{3D}\mapsto\text{2D}}))$ involves stating each secondary structure state in $\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle$ using the inferred models as follows.

$$I(\langle S_i^{\text{2D}}, T_i^{\text{2D}}\rangle | \mathcal{A}_i^{\text{3D}\mapsto\text{2D}}, \mathbf{M}(t_i^{\text{3D}\mapsto\text{2D}})) = \sum_{\forall\langle s_k, r_l\rangle \in \mathcal{A}_{\texttt{match}}^{\text{3D}\mapsto\text{2D}}} I(\langle s_k, r_l\rangle, \mathbf{M}(t_i^{\text{3D}\mapsto\text{2D}})) + \sum_{\forall s_k \in \mathcal{A}_{\texttt{insert}}^{\text{3D}\mapsto\text{2D}}} I(\pi(s_k)) + \sum_{\forall r_l \in \mathcal{A}_{\texttt{delete}}^{\text{3D}\mapsto\text{2D}}} I(\pi(r_l))$$

The first term refers to the statement of the secondary structure pairs in the matched regions of the alignment $\mathcal{A}_i^{\text{3D}\mapsto\text{2D}}$ using the conditional probability of secondary structure substitution in $\mathbf{M}(t_i^{\text{3D}\mapsto\text{2D}})$. The second and third term refers to the statement of secondary structures in inserted and deleted regions respectively using the stationary distribution π. Note the diagonal of the matrix $\mathbf{M}$ at t=1 gives the stationary probabilities.

S3 Search for the best Markov matrix M^* and associated time-dependent Dirichlet parameters $\boldsymbol{\alpha}$

S3.1 Search for M^* with fixed $\{\boldsymbol{\alpha}, \vec{\Theta}\}$

Given the objective function in Equation 2-4 in the main text, and a dataset $\mathbf{D}^{\text{3D}\mapsto\text{2D}}$, we first search for the best matrix $\mathbf{M}^*$ using a simulated annealing approach while holding the $\boldsymbol{\alpha}$ fixed and consequently $\vec{\Theta}$ fixed. Initially, the search starts with a matrix where the probabilities on the diagonal cells are set to 0.90 and $1 - 0.9$ is distributed uniformly across the remaining cells in each column on SSTSUM. The same set of Dirichlet parameters $\boldsymbol{\alpha}$ and corresponding $\vec{\Theta}$ parameters of the MMLSUM matrix were used as a starting point in the inference process.

The cooling schedule of the simulated annealing process starts with a temperature of $T = 10,000$ and decreases by a factor of 0.88. At each iteration, the current matrix $\mathbf{M}$ is perturbed randomly by selecting one of its column vectors. This random selection is made based on the stationary distribution π. The perturbation step samples a new vector from a Dirichlet distribution where $\vec{\mu}$ is the selected column and κ is a specified high-concentration parameter. Initially, the concentration parameter is set to 10,000 and increased by a factor of 0.88 at each temperature step. A random sampling of a k-dimensional probability vector $\vec{\theta}$ from a k-dimensional Dir($\vec{\alpha}$) involves two steps. First, for each component $\alpha_i \in \vec{\alpha}$, generate a Gamma distributed random sample y_i from the Gamma distribution $\Gamma(\alpha_i, 1)$. Then, $\mathbb{L}_1$-normalise the sampled vector. During each perturbation step, the matrix is properly normalized, and the expected change of the matrix is always ensured to be 1%. Once the perturbed matrix $\tilde{\mathbf{M}}$ is identified, the next state $x(T+1)$ is determined using the Metropolis criterion as follows.

$$
\begin{aligned}
&\text{if } I(H,D):\tilde{\mathbf{M}} \leq I(H,D):\mathbf{M}, \text{ then } x(T+1) = \tilde{\mathbf{M}} \\
&\text{if } I(H,D):\tilde{\mathbf{M}} > I(H,D):\mathbf{M}, \text{ then} \\
&\quad x(T+1) = \tilde{\mathbf{M}}, \text{ with probability } 2^{\frac{-\left(I(H,D):\tilde{\mathbf{M}} - I(H,D):\mathbf{M}\right)}{T}} \\
&\quad x(T+1) = \mathbf{M}, \text{otherwise}
\end{aligned}
\tag{11}
$$

At each temperature step, the matrix is perturbed 5000 times until the temperature reaches 0.0001.

S3.2 Search for $\boldsymbol{\alpha}$ with fixed $\{\mathbf{M}, \{t_1^{\text{3D}\mapsto\text{2D}}, t_2^{\text{3D}\mapsto\text{2D}}, \ldots, t_{|\mathbf{D}|}^{\text{3D}\mapsto\text{2D}}\}$

Given a dataset $\mathbf{D}^{\text{3D}\mapsto\text{2D}}$, we can infer the evolutionary time for each alignment (see Section 2.4 in the main text). Then we group each alignment into their discrete bins of structure time $t_i^{\text{3D}\mapsto\text{2D}}$, which result in subsets of alignments for each time $t_i^{\text{3D}\mapsto\text{2D}} \in [1, t_{\text{max}} = 250]$. We hold the stochastic matrix $\mathbf{M}$ fixed and consequently $\{t_1^{\text{3D}\mapsto\text{2D}}, t_2^{\text{3D}\mapsto\text{2D}}, \ldots, t_{|\mathbf{D}|}^{\text{3D}\mapsto\text{2D}}\}$ fixed and infer 1-simplex and 2-simplex Dirichlet models for each discrete time bin by minimizing the objective function in Equation 2-4 in the main text. We use a simulated annealing approach with the same parameters used for the inference of $\mathbf{M}^*$ (see Section S3.1), except for the concentration parameter κ. For `match` state, κ is initialized to 10,000 and for `insert` state, κ is initialized to 1,000. At each temperature step we randomly perturbed either μ or the concentration parameter κ of the Dirichlets (see Fig. 1 for the pseudocode). The resultant $\vec{\alpha}$ is accepted/rejected based on the same metropolis criterion in Equation 11.

At each temperature step, $\vec{\alpha}$ is perturbed 5000 times until the temperature reaches 0.001. The same process is continued to compute the best $\boldsymbol{\alpha}$, $\forall t_i^{\text{3D}\mapsto\text{2D}} \in [1, 250]$. $\vec{\Theta}$ is estimated from these values of $\vec{\alpha}(t_i^{\text{3D}\mapsto\text{2D}})$ as described in Section S2.2.

Function `Perturb_Mean` $(\vec{\mu}, \kappa)$ **:**
- $\vec{y} \leftarrow \vec{0}$;
- sum $\leftarrow 0$;
- $\vec{\alpha} \leftarrow \kappa \cdot \vec{\mu}$;
- **for** $i \leftarrow 1$ **to** $|\vec{\alpha}|$ **do**
  - $y_i \leftarrow$ **Γ_random**$(\vec{\alpha}_i, 1)$;
  - sum $\leftarrow$ sum $+ y_i$;
- **end**
- **for** $i \leftarrow 1$ **to** $|\vec{\alpha}|$ **do**
  - $y_i \leftarrow \frac{y_i}{\text{sum}}$;
- **end**

return $\vec{y}$

Function `Perturb_Kappa` $(\vec{\mu}, \kappa)$ **:**
- $\delta \leftarrow$ `random_uniform`$(0.1, 1)$;
- $v \leftarrow$ `random_uniform`$(0, 1)$;
- $\vec{y} \leftarrow \vec{0}$;
- **if** $v \leq 0.5$ **then**
  - $\kappa \leftarrow \kappa + \delta$;
- **else**
  - $\kappa \leftarrow \kappa - \delta$;
- **end**
- **for** $i \leftarrow 1$ **to** $|\vec{\alpha}|$ **do**
  - $y_i \leftarrow \kappa \times \mu_i$;
  - sum $\leftarrow$ sum $+ y_i$;
- **end**
- **for** $i \leftarrow 1$ **to** $|\vec{\alpha}|$ **do**
  - $y_i \leftarrow \frac{y_i}{\text{sum}}$;
- **end**

return $\vec{y}$

Figure 1. Pseudocode for perturbing the parameters of a Dirichlet distribution

S3.3 Statistics on $\mathbf{M}^*$

A stochastic Markov matrix $\mathbf{M}^{t^{3D\mapsto 2D}}$ can be decomposed as $\mathbf{M}^{t^{3D\mapsto 2D}} = \mathbf{S}\Lambda^{t^{3D\mapsto 2D}}\mathbf{S}^{-1}$ based on the eigen decomposition theorem. Here $\mathbf{S}$ is the eigenvector and Λ is the diagonal eigenvalue matrix of $\mathbf{M}^1$. The set of all eigenvalues $\{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ (in their descending order) is real and positive. The largest eigenvalue λ_1 which is known as the *Perron-Frobenius* eigenvalue is 1 as shown in Fig. 2. The eigenvector associated with λ_{max} corresponds to the stationary distribution. All eigenvalues reach 0 at equilibrium, except for λ_{max} which remains a constant. This means all eigenvalues except for λ_{max} control the convergence of the matrix.

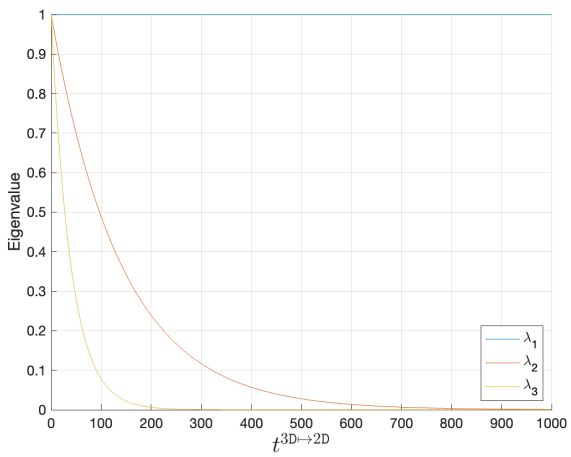

Figure 2. Variation of eigenvalues ($\lambda^{t^{3D\mapsto 2D}}$) of SSTSUM with $t^{3D\mapsto 2D}$.

Kullback-Leibler (KL) divergence estimates the measure of relative Shannon entropy between two probability distributions. KL divergence between each secondary structure column in the conditional probability matrix and its stationary distribution reflects the speed at which each column converges into its equilibrium state. This can be computed as:

$$\text{KL divergence} = \sum_{i=1}^{K}\sum_{j=1}^{K} M_{i|j}\log\left(\frac{M_{i|j}}{\pi_i}\right)$$

where $M_{i|j}$ is the conditional probability for the pair of secondary structure states indexed by i and j. π_i is the probability of the stationary distribution indexed by i. Figure 3 shows the KL divergence of SSTSUM which measures the convergence of each column vector to their respective stationary distributions. Once a secondary structure state reaches equilibrium (stationary probability), it is no longer able to differentiate any secondary structure substitution against random occurrences.

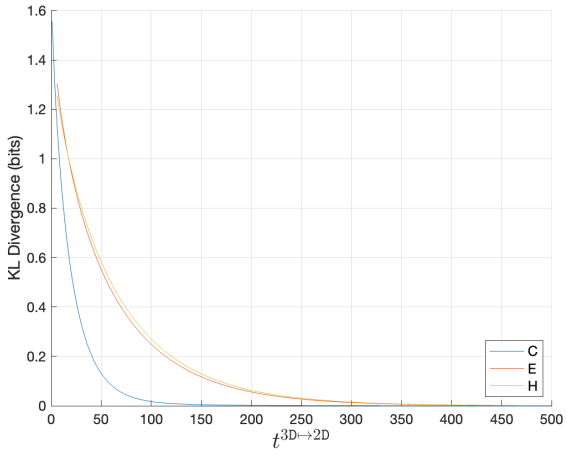

Figure 3. KL divergence measuring the convergence of each column vector in SSTSUM to their respective stationary distributions.

S4 Choice of the data source

S4.1 SCOP2 benchmark dataset

We used the same benchmark dataset SCOP2 [Sumanaweera et al., 2022] which was used to infer MMLSUM (a time-parameterized model for amino acid substitutions) for the inference of SSTSUM and associated models. This dataset contains 59,092 unique domain pairs sampled from family and superfamily levels of the Structural Classification of Proteins (SCOP v2.07) [Murzin et al., 1995]. The list of alignments is available from http://lcb.infotech.monash.edu/sstsum/smdata/scop2/SCOP2_PairwiseAlignmentList.txt and the tarball containing all the structure alignments produced by MMLigner [Collier et al., 2017] can be downloaded from http://lcb.infotech.monash.edu/sstsum/smdata/scop2/SCOP2_mmligner_benchmark.tar.gz

S4.2 Set of million domain pairs sampled from family and superfamily levels of SCOP

In this work, we randomly sampled a million domain pairs from the (SCOPe v2.08) database [Murzin et al., 1995] to compare the divergence time of structures ($t^{3D\mapsto 2D}$) and sequences (t^{1D}). Table 1 shows the key statistics of this dataset. The full list of scop domain pairs including the SCOP domain identifiers, SCOP paths, and the SCOP classification level can be downloaded from http://lcb.infotech.monash.edu/sstsum/smdata/rawdata/million.txt.

Table 1. Distribution of the sampled one million domain pairs based on different classes of SCOP.

SCOP Class	SCOP Level		
	Family	Superfamily	Total
all-α	133,939	103,159	237,098
all-β	86,403	172,402	258,805
α/β	70,694	183,851	254,545
α+β	168,013	81,539	249,552

S4.3 Five sets of domain pairs sampled at varying levels of SCOP hierarchy

We did another analysis to compare the divergence time of structures classified in each hierarchical level of SCOP. We further sampled distinct sets of domain pairs from each hierarchical level of SCOP such that each domain appears at most once in the dataset. This comprised 5 sets of domain pairs sampled at the same family, same superfamily, same fold, same class, and decoy (different class) levels respectively. See Table 2 for more information on the datasets. The list of domain pairs of these 5 datasets is available on the supplementary website: http://lcb.infotech.monash.edu/sstsum.

Table 2. Distribution of the five distinct sets of domain pairs sampled from varying levels of SCOP classification.

SCOP Level	No. of domain pairs
Family	55,201
Superfamily	31,600
Fold	40,582
Class	40,551
Decoy	40,466

S4.4 Raw data used in secondary structure prediction

We used a non-redundant dataset containing 45,887 protein sequences and their secondary structure assignments that were deposited before the 1st of January 2017 to search for hits in the secondary structure prediction method. The list of PDB IDs is available from http://lcb.infotech.monash.edu/sstsum/smdata/rawdata/pdb-90.txt.

S4.5 Data used in plots

All the raw data used to generate the plots in the main text is available on the supplementary website: http://lcb.infotech.monash.edu/sstsum. This includes the list of PDB IDs of the targets released in CASP 14 and 15 which was used to evaluate SSTPred with 3 other secondary structure predictors.

References

L. Allison. *Coding Ockham's Razor*. Springer, 2018. URL https://doi.org/10.1007/978-3-319-76433-7.

J. H. Collier, L. Allison, A. M. Lesk, P. J. Stuckey, M. Garcia de la Banda, and A. S. Konagurthu. Statistical inference of protein structural alignments using information and compression. *Bioinformatics*, 33(7):1005–1013, 2017.

J. H. Conway and N. J. Sloane. On the voronoi regions of certain lattices. *SIAM Journal on Algebraic Discrete Methods*, 5(3):294–305, 1984.

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*, 247(4):536–540, 1995.

D. Sumanaweera, L. Allison, and A. S. Konagurthu. Statistical compression of protein sequences and inference of marginal probability landscapes over competing alignments using finite state models and dirichlet priors. *Bioinformatics*, 35(14):i360–i369, 2019.

D. Sumanaweera, L. Allison, and A. S. Konagurthu. Bridging the gaps in statistical models of protein alignment. *Bioinformatics*, 38(Supplement_1): i229–i237, 2022.

C. S. Wallace and D. M. Boulton. An invariant Bayes method for point estimation. *Classification Society Bulletin*, 3(3):11–34, 1975.

C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society: Series B (Methodological)*, 49 (3):240–252, 1987.

C. S. Wallace and C. S. Wallace. *Statistical and inductive inference by minimum message length*. Springer, 2005.