Bioinformatics, 35, 2019, i360–i369
doi: 10.1093/bioinformatics/btz368
ISMB/ECCB 2019

Statistical compression of protein sequences and inference of marginal probability landscapes over competing alignments using finite state models and Dirichlet priors

Dinithi Sumanaweera, Lloyd Allison* and Arun S. Konagurthu*

Faculty of Information Technology, Monash University, Clayton, VIC 3800, Australia

*To whom correspondence should be addressed.

Abstract

The information criterion of minimum message length (MML) provides a powerful statistical framework for inductive reasoning from observed data. We apply MML to the problem of protein sequence comparison using finite state models with Dirichlet distributions. The resulting framework allows us to supersede the *ad hoc* cost functions commonly used in the field, by systematically addressing the problem of arbitrariness in alignment parameters, and the disconnect between substitution scores and gap costs. Furthermore, our framework enables the generation of marginal probability landscapes over all possible alignment hypotheses, with potential to facilitate the users to simultaneously rationalize and assess competing alignment relationships between protein sequences, beyond simply reporting a single (best) alignment. We demonstrate the performance of our program on benchmarks containing distantly related protein sequences.

Availability and implementation: The open-source program supporting this work is available from: http://lcb.infotech.monash.edu.au/seqmmligner.

Contact: arun.konagurthu@monash.edu or lloyd.allison@monash.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Identifying the evolutionarily retained similarities between macromolecular sequences remains an indispensable first step of many biological studies (Lesk, 2017). Comparison of sequences are carried out using the computational technique of *alignment*. An alignment is used as a surrogate for how two (or more) sequences are evolutionarily related and provides a quantitative measure of their similarity (Allison *et al.*, 1992, 1999; Barton and Sternberg, 1987; Lesk, 2017).

Alignment techniques commonly depend on choosing a model for relating two sequences with stated parameters to score matched and penalize unmatched (gap) regions in any alignment. However, in practice, it is left for the users to fine-tune these parameters in their quest for meaningful sequence relationships. This parameter tuning remains a 'black art based on trial and error' (Do *et al.*, 2005), and several studies have underscored major problems with parameter tuning and demonstrated the conflicting effects this has on resulting alignments (Do *et al.*, 2006; Löytynoja and Goldman, 2008; Rivas and Eddy, 2015; Vingron and Waterman, 1994). These limitations were best summarized by Löytynoja and Goldman (2008) in their observation that 'alignment is still a highly error-prone step in comparative sequence analysis'.

The problem becomes more pronounced when comparing amino acid sequences of proteins, which additionally rely on substitution matrices. A protein sequence alignment gives a one-to-one correspondence between amino acids symbols, and substitution scores are used to quantify these correspondences. A substitution matrix over the standard amino acid alphabet is parameterized on a distance (or alternatively a similarity) parameter between sequences, and conveys the mutability of one amino acid changing into another. PAM (Dayhoff *et al.*, 1978) and BLOSUM (Henikoff and Henikoff, 1992) are widely used series of substitution matrices.

Yet, in the use of these substitution matrices, there remains a severe disconnect between the substitution scores (used to score the matched amino acid correspondences) and gap parameters (used to

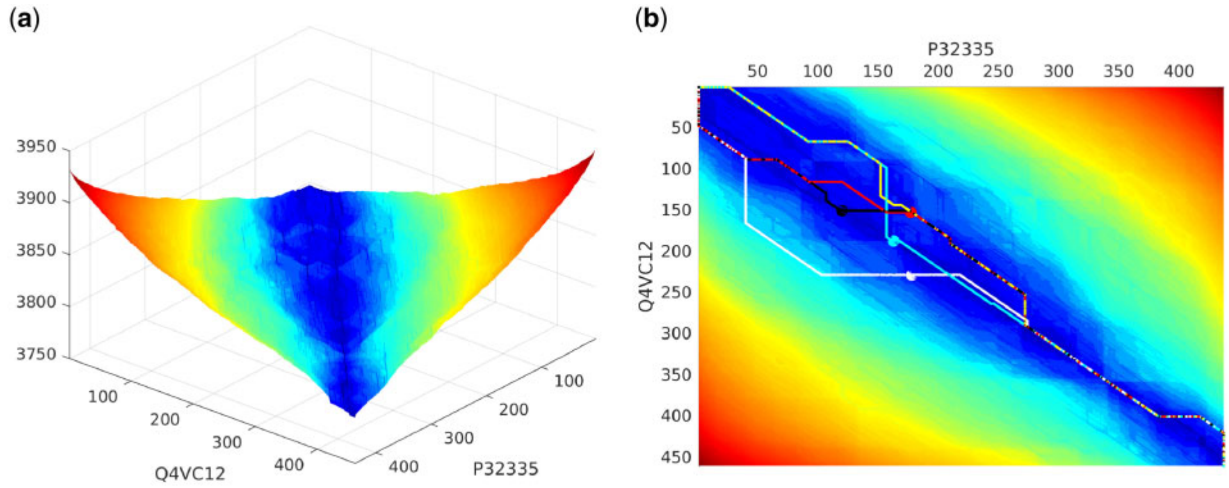

Fig. 1. (**a**) Marginal probability landscape to visualize the relatedness of two protein sequences over all competing alignments. (**b**) Most probable alignments passing through a set of specific cells (*i*, *j*) in the sequence landscape, including the most probable over all cells (shown in black)

penalize the unmatched regions of alignments). Previous studies have shown that, different protein sequence alignment programs with varying choices of substitution and gap parameters convey radically different alignments (Barton and Sternberg, 1987; Blake and Cohen, 2001; Fitch and Smith, 1983; Löytynoja and Goldman, 2008; Vingron and Waterman, 1994). Further, the confidence of reported relationship in an alignment plummets when comparing protein sequences from the *twilight zone* [by common consensus defined as sequences sharing less than 35% sequence identity (Doolittle, 1986)]. Do *et al.* (2006) observed that the correctness of a reported sequence alignment is highly questionable when it is below a 25% sequence identity.

Beyond the problem with tuning alignment parameters, another major shortcoming is that most current programs report a single alignment. Many studies have demonstrated that optimizing an objective function under a specified model of sequence relatedness does not necessarily imply an optimization in homology (Fitch and Smith, 1983; Durbin *et al.*, 1998; Levy Karin *et al.*, 2019; Redelings and Suchard, 2005; Rosenberg, 2009) (see Supplementary notes for a more detailed discussion).

Aforementioned deficiencies provide a motivation for the work presented here. Specifically for comparing protein sequences, we attempt to rectify: (i) the problem of arbitrariness of alignment parameters, (ii) the disconnect between substitution scores and gap parameters, and (iii) the lack of a rigorous statistical framework where all competing alignments can be simultaneously rationalized and assessed, beyond reporting just a single (best) alignment.

Our work uses the statistical inductive inference method of minimum message length (MML) encoding (Allison, 2018; Wallace, 2005; Wallace and Boulton, 1968) to compare the relatedness of protein sequences in information-theoretic terms, measurable in *bits*. In this framework, any alignment between protein sequences is a string generated by a three-state alignment machine with `match`, `insert` and `delete` states. This allows us to compute robust probability estimates of sequence alignments.

Independently, we infer from a set of 118 384 structural alignments between protein domains, the parameters of Dirichlet probability distributions that allow us to link the distance parameter of existing substitution matrices (e.g. parameter '*n*' of PAM-*n*) with the state machine (transition probability) parameters that produce the three-state alignment strings. These parameters provide a substantial statistical rationalization of the otherwise *ad hoc* use of gap penalties. In statistical learning theory, the Dirichlet distribution is often used as a prior distribution over the parameters of finite-state models. It is a conjugate prior for multistate models, which implies that the resultant posterior is also a Dirichlet distribution (Allison, 2018).

Furthermore, building on these inferred Dirichlet priors, we design a statistical compression based alignment methodology with automatically estimated parameters, to compute the marginal probability landscape of any given protein sequence pair (e.g. see Fig. 1(a)). Marginal probability estimation (Trumpler and Weaver, 1953) applied to sequence alignment provides a powerful technique to highlight the relationship between sequences, by *marginalizing* over all the alignments between the sequences. Specifically, for a pair of sequences $\mathbf{S}_{1\ldots|\mathbf{S}|} : \mathbf{T}_{1\ldots|\mathbf{T}|}$, any cell (i, j) in this landscape gives the product of marginal probabilities that the prefixes $\mathbf{S}_{1\ldots i} : \mathbf{T}_{1\ldots j}$ and suffixes $\mathbf{S}_{i+1\ldots|\mathbf{S}|} : \mathbf{T}_{j+1\ldots|\mathbf{T}|}$ are related. This involves integrating over all possible alignments passing through the cell (i, j) in any of the three alignment states. Since these are rigorous estimates of probabilities of relationship between sequences, the resultant landscape allows the user to visualize not just the most probable alignment, but also interactively query closely competing alignments passing through any cell (e.g. see Fig. 1(b)).

Most importantly, the MML framework provides a natural statistical significance test when assessing any alignment or comparing one with another. This is measurable in bits of compression with respect to the *null model* message length that gives the Shannon's information content (Shannon, 1948) [related to the Kolmogorov complexity (Kolmogorov, 1963)] of each of the two sequences independently summed up.

Finally, asymptotic computational complexity to compare sequences under this framework and generate these marginal alignment landscapes is $O(|\mathbf{S}||\mathbf{T}|)$. Any specific competing alignment can be probed and reported in $O(|\mathbf{S}| + |\mathbf{T}|)$ time after the initial $O(|\mathbf{S}||\mathbf{T}|)$ effort.

We note that our work develops and significantly extends the basic ideas of finite state models for alignment introduced by Allison

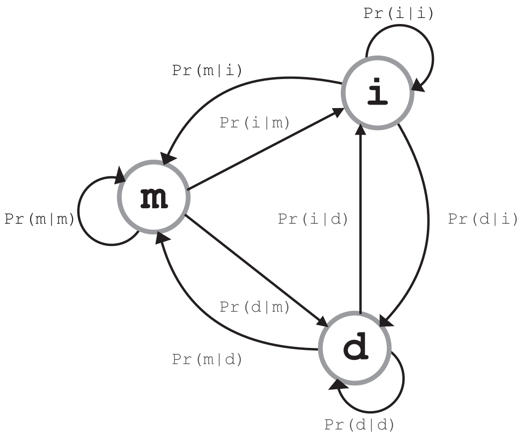

Fig. 2. Three-state machine for alignment string modeling

et al. (1992, 1999), Powell et al. (2004) and Sumanaweera et al. (2018). More generally, other noteworthy attempts at probabilistic modeling of alignments, although with different aims and motivations, include: Durbin et al. (1998), Zhu et al. (1998), Do et al. (2005, 2006) and Rivas and Eddy (2015). See supplementary Section S4 for an overview of the literature on this topic.

2 Materials and methods

2.1 Primer on MML criterion

MML encoding is a Bayesian method for hypothesis (model) selection that is grounded in information and coding theory (Allison, 2018; Wallace, 2005; Wallace and Boulton, 1968).

For any hypothesis H on observed data D, their joint probability is given by: $\Pr(H,D) = \Pr(H)\Pr(D|H) = \Pr(D)\Pr(H|D)$. Independently, Shannon (1948) in his *Mathematical Theory of Communication* quantified the information content (measurable in *bits*) in any event E that occurs with a probability of $\Pr(E)$ as $\mathrm{I}(E) = -\log_2(\Pr(E))$. In other words, $\mathrm{I}(E)$ is the shortest message length required to communicate *losslessly* the information conveyed by E. Applying Shannon's insight, the joint probability between any hypothesis and data can be expressed in terms of Shannon's information content as: $\mathrm{I}(H,D) = \mathrm{I}(H) + \mathrm{I}(D|H) = \mathrm{I}(D) + \mathrm{I}(H|D)$. This can be rationalized as the length of the message communicated between an imaginary transmitter-receiver pair; the transmitter's goal is to encode and send the data D over a chosen hypothesis H as efficiently as possible, so that the receiver can decode the message and losslessly recover D. The transmission message is in two parts: the first deals with encoding and sending the hypothesis, taking $\mathrm{I}(H)$ bits; the second deals with encoding the data given the hypothesis, taking $\mathrm{I}(D|H)$ bits.

The difference in the message lengths, $\mathrm{I}(H_1,D) - \mathrm{I}(H_2,D)$, between any two hypotheses H_1 and H_2 explaining the same data D gives the log-odds posterior ratio test:

$$\mathrm{I}(H_1,D) - \mathrm{I}(H_2,D) = -\log\left(\frac{\Pr(D)\Pr(H_1|D)}{\Pr(D)\Pr(H_2|D)}\right) = \log\left(\frac{\Pr(H_2|D)}{\Pr(H_1|D)}\right).$$

More importantly, implicit in this framework is a null hypothesis (or model). The null model message involves stating D without venturing a hypothesis—i.e. stating D raw, but as efficiently as possible—taking $\mathrm{I}_{\mathrm{NULL}}(D)$ bits. If the best hypothesis H^*, the one that yields the shortest message length $\mathrm{I}(H^*,D)$ over the space of all possible hypotheses, does not beat the null model (i.e. $\mathrm{I}(H^*,D) > \mathrm{I}_{\mathrm{NULL}}(D)$), then that hypothesis has to be *rejected*.

The MML paradigm for hypothesis selection provides a direct tradeoff between hypothesis complexity $\mathrm{I}(H)$ and its fit to the data $\mathrm{I}(D|H)$: a more complex hypothesis results in a higher value of $\mathrm{I}(H)$ but may describe the data better with a lower value of $\mathrm{I}(D|H)$, and vice versa. Furthermore, MML ensures complete transparency in communication. Any information that is not common knowledge or based on preconceived notions implicit in the data has to be included as part of the message sent by the transmitter. Otherwise the transmission is not lossless, and the message sent will be indecipherable by the receiver. No parameters can be hidden from the communication, and the precision of statement of real-valued parameters has to be directly dealt with as part of the MML framework (Wallace, 2005; Wallace and Freeman, 1987). Note, the practical realization of MML is built on strong statistical foundations developed over 40 years in the general statistical learning literature outside molecular biology (Allison, 2018; Wallace, 2005; Wallace and Freeman, 1987).

2.2 Protein sequence comparison in MML paradigm

Given two protein sequences $\mathbf{S}$ and $\mathbf{T}$, we want to know if they are related, and if so, how? Any alignment $\mathcal{A}$ between $\langle\mathbf{S},\mathbf{T}\rangle$ proposes a specific hypothesis of their relationship. Consequently, using the MML framework, the fitness of any alignment relationship between sequences can be quantified over a two-part message. The first part encodes the explanation of the relationship specified by $\mathcal{A}$, while the second part encodes the details of the amino acid symbols of $\mathbf{S}$ and $\mathbf{T}$ under the relationship specified by $\mathcal{A}$. We call this form of transmission of sequences using an alignment, the *alignment-model* message. This model gives the following message length terms:

$$\mathrm{I}(\mathcal{A},\langle\mathbf{S},\mathbf{T}\rangle) = \underbrace{\mathrm{I}(\mathcal{A})}_{\text{First part}} + \underbrace{\mathrm{I}(\langle\mathbf{S},\mathbf{T}\rangle|\mathcal{A})}_{\text{Second part}} \text{ bits} \qquad (1)$$

Any alignment hypothesis $\mathcal{A}$ over the sequences $\langle\mathbf{S},\mathbf{T}\rangle$ specifies a string over three states, `match` (`m`), `insert` (`i`) and `delete` (`d`), generated from a three-state machine with unknown parameters (Fig. 2). For a given pair of sequences, these state-machine parameters are to be inferred in concert with the distance parameter (also unknown) between the two sequences. (These details together with computation of the message length terms denoted in Equation 1 are specified in Sections 2.3–2.4.)

Thus in this framework, the best alignment hypothesis $\mathcal{A}^*$ is the one that yields the shortest two-part message, $\mathrm{I}(\mathcal{A}^*,\langle\mathbf{S},\mathbf{T}\rangle)$. More importantly, in reasoning about the relationship between the sequences, MML allows the contribution of *all* alignments to be considered according to their respective probabilities. In probabilistic terms, this is equivalent to computing the marginal probability $\Pr(\langle\mathbf{S},\mathbf{T}\rangle)$ that the sequences are related. More formally, since the set of all possible alignments (say $\mathbf{A}$) contains alignment hypotheses that are pairwise disjoint from each other, by the law of total probability (Bayes, 1763), we have:

$$\Pr(\langle\mathbf{S},\mathbf{T}\rangle) = \sum_{\forall\mathcal{A}\in\mathbf{A}} \Pr(\mathcal{A})\Pr(\langle\mathbf{S},\mathbf{T}\rangle|\mathcal{A}). \qquad (2)$$

The summation over all possible alignments as per Equation 2 gives the marginal probability of the relationship between two sequences $\langle\mathbf{S},\mathbf{T}\rangle$. The corresponding message length takes $\mathrm{I}_{\mathrm{marginal}}(\langle\mathbf{S},\mathbf{T}\rangle) = -\log_2(\Pr(\langle\mathbf{S},\mathbf{T}\rangle))$ bits. Marginal probability provides a general and unbiased probability estimate of the relationship compared with that of any specified alignment (see Equation 1). It follows that $\mathrm{I}_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle) < \mathrm{I}(\mathcal{A}^*, \langle \mathbf{S}, \mathbf{T} \rangle)$.

Finally, MML framework provides a natural null hypothesis test to verify the significance of any hypothesis (see Section 2.1). This involves explaining **S** and **T** independently, assuming they are unrelated. We refer to this as the *null-model* message whose length is given by:

$$\mathrm{I}_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle) = \mathrm{I}_{\text{NULL}}(\mathbf{S}) + \mathrm{I}_{\text{NULL}}(\mathbf{T}) \quad \text{bits} \qquad (3)$$

In general, if $\mathrm{I}_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle) - \mathrm{I}_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle) > 0$, the hypothesis that $\langle \mathbf{S}, \mathbf{T} \rangle$ are related is accepted. Specifically, if $\mathrm{I}_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle) - \mathrm{I}(\mathcal{A}^*, \langle \mathbf{S}, \mathbf{T} \rangle) > 0$, then $\mathcal{A}^*$ provides the best hypothesis of their relationship. Otherwise, both the marginal and optimal hypotheses are rejected.

2.3 MML inference of Dirichlet priors on state machine parameters as a function of sequence distance

As introduced in Section 2.2, any alignment $\mathcal{A}$ of $\langle \mathbf{S}, \mathbf{T} \rangle$ is a string produced by a three-state machine. Figure 2 illustrates the three-state machine over `match` (`m`), `insert` (`i`) and `delete` (`d`) states, with nine one-step transition probabilities. This machine has an implicit constraint that the sum of one-step transitions out of each state should add up to the probability of 1: $\sum \Pr(*|\mathtt{m}) = \sum \Pr(*|\mathtt{i}) = \sum \Pr(*|\mathtt{m}) = 1, \forall * \in \{\mathtt{m}, \mathtt{i}, \mathtt{d}\}$. Consistent to the accepted practice of aligning macromolecular sequences, the following transition symmetries are additionally enforced in this work: $\Pr(\mathtt{i}|\mathtt{m}) = \Pr(\mathtt{d}|\mathtt{m})$; $\Pr(\mathtt{m}|\mathtt{i}) = \Pr(\mathtt{m}|\mathtt{d})$; $\Pr(\mathtt{d}|\mathtt{i}) = \Pr(\mathtt{i}|\mathtt{d})$; $\Pr(\mathtt{i}|\mathtt{i}) = \Pr(\mathtt{d}|\mathtt{d})$. With these symmetries, the three-state alignment machine now has three free (unknown) parameters. Notionally, these are: (i) $\Pr(\mathtt{m}|\mathtt{m})$, (ii) $\Pr(\mathtt{i}|\mathtt{i})$ and (iii) $\Pr(\mathtt{m}|\mathtt{i})$. Once these free parameters are estimated, the enforced symmetries allow the remaining transition probabilities to be assigned as follows: $\Pr(\mathtt{i}|\mathtt{m}) = \Pr(\mathtt{d}|\mathtt{m}) = \frac{1-\Pr(\mathtt{m}|\mathtt{m})}{2}$; $\Pr(\mathtt{d}|\mathtt{i}) = \Pr(\mathtt{i}|\mathtt{d}) = 1 - \Pr(\mathtt{i}|\mathtt{i}) - \Pr(\mathtt{m}|\mathtt{i})$; $\Pr(\mathtt{d}|\mathtt{d}) = \Pr(\mathtt{i}|\mathtt{i})$; $\Pr(\mathtt{m}|\mathtt{d}) = \Pr(\mathtt{m}|\mathtt{i})$.

Therefore, our use of the symmetric three-state alignment machine entails estimation of the three free parameters distributed over a 1-simplex for $\Pr(\mathtt{m}|\mathtt{m})$, and a 2-simplex for $\Pr(\mathtt{i}|\mathtt{i})$ and $\Pr(\mathtt{m}|\mathtt{i})$. We note that the unit $(k-1)$-simplex involves L1 normalized, k dimensional vectors (Allison, 2018). Thus, a vector of k probabilities corresponding to mutually exclusive events can be represented as a point in a unit $(k-1)$-simplex.

To estimate the state machine parameters, especially as a function of the distance between two sequences, we undertake the following one-time probabilistic modeling exercise using Dirichlet distributions over simplexes. These derived Dirichlet priors support the parameter estimation in our MML based protein sequence comparison framework.

2.3.1 Preparation of datasets for inference

We randomly sampled from SCOP (ver. 2.07) (Murzin *et al.*, 1995) a set of 118 384 protein structural domain-pairs, containing 47 687 domain-pairs that are related at a family level and 70 697 domain pairs related more distantly at a superfamily level. See Supplementary Section S3 for details of the domain-pairs and the method used for random selection. All 118 384 domain pairs are *structurally aligned* using their 3D coordinate information (Collier *et al.*, 2017). These structural alignments are used in our modeling exercise below.

Furthermore, these structural alignments are partitioned into subsets based on the observed sequence-distance between their corresponding SCOP domain-pairs. A surrogate measure of sequence-distance was provided by Dayhoff *et al.* (1978) in terms of Point Accepted Mutation (PAM) units. They derived PAM-1 transition probability matrix over the standard 20-letter amino acid alphabet that gives the probability of one amino acid mutating into another in a unit PAM distance step. The PAM-1 matrix can be generalized to any PAM-n via exponentiation, PAM-n = (PAM-1)^n, which gives the probability of one amino acid mutating into another in n PAM distance steps. (Indeed other notions of sequence-distances could be used (e.g. BLOSUM). We use PAM in this work only for its convenience that allows us to generalize it systematically to arbitrary distances between sequences.)

Therefore, for each structural alignment in our set of alignments, we identify the best integer n in the range [1, 1000] that maximizes the probability of matched amino acids in that alignment using PAM-n, yielding 1000 subsets of alignments. These alignment subsets are used to infer 1000 Dirichlet priors, one for each subset of observed alignments as a function of their corresponding sequence-distance.

Dirichlet distributions are conjugate priors for multistate models over finite-state strings with any fixed k discrete states. Multistate models define k parameters $[\theta_1, \theta_2, \ldots \theta_k]$ (of which $k-1$ are free) that lie in a $(k-1)$-simplex. In our work, for each subset corresponding to the sequence-distance parameter n in the range [1, 1000], we have a set of observed three-state alignment strings. Our goal is to model these and infer Dirichlet prior parameters over the corresponding multistate transition probability parameters. As discussed above, the three free parameters of the symmetric three-state alignment machine (see Fig. 2) can be decomposed and modeled using Dirichlet distributions over a unit 1-simplex (accounting for the free parameter $\Pr(\mathtt{m}|\mathtt{m})$), and a unit 2-simplex (accounting for the remaining free parameters $\Pr(\mathtt{i}|\mathtt{i})$ and $\Pr(\mathtt{m}|\mathtt{i})$).

Below we summarize the MML method of inferring free parameters from an observed dataset containing finite-state strings, along with their optimal Dirichlet prior parameters.

2.3.2 Estimation of finite state transition probabilities over any Dirichlet prior

Let $\mathrm{Dir}(\vec{\alpha})$ be a Dirichlet distribution with model parameters $\vec{\alpha} = [\alpha_1, \alpha_2, \ldots \alpha_k]$ (for $\alpha_i > 0$) that describes a random variable (data sample) $\vec{\Theta} = [\theta_1, \theta_2, \ldots \theta_k]$ representing a point in the unit $(k-1)$-simplex (i.e. $\sum\limits_{i=1}^{k} \theta_i = 1$). The $\vec{\alpha}$ can be reparameterized as $(\kappa, \hat{\mu})$ to intuitively show how the distribution concentrates with a concentration parameter κ around its L1-normalized mean vector $\hat{\mu}$ in the $k-1$ simplex:

$$\vec{\alpha} = \underbrace{\left(\sum_{i=1}^{k} \alpha_i\right)}_{\kappa=\text{concentration}} \times \underbrace{\left[\frac{\alpha_1}{\sum_{i=1}^{k} \alpha_i}, \frac{\alpha_2}{\sum_{i=1}^{k} \alpha_i}, \ldots \frac{\alpha_k}{\sum_{i=1}^{k} \alpha_i}\right]}_{\hat{\mu}=\text{mean} \quad \text{vector}}.$$

Dirichlet probability density function is given by:

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

where $\mathrm{B}(\vec{\alpha})$ is the multivariate Beta function written in terms of Gamma functions: $\mathrm{B}(\vec{\alpha}) = \Pi_1^k \Gamma(\alpha_i)/\Gamma(\kappa)$. The likelihood over data $\boldsymbol{\Theta}$ with N data samples: $[\vec{\Theta}^1, \vec{\Theta}^2, \ldots \vec{\Theta}^{\mathrm{N}}]$ is defined as: $f(\Theta|\vec{\alpha}) = \prod_{n=1}^{\mathrm{N}} f(\vec{\Theta}^n|\vec{\alpha})$. Thus, the negative log likelihood function is:

$$\mathcal{L}(\vec{\alpha}) = -\mathrm{N}\log\Gamma(\kappa) + \mathrm{N}\sum_{i=1}^{k}\log\Gamma(\alpha_i) - \sum_{n=1}^{\mathrm{N}}\sum_{i=1}^{k}(\alpha_i - 1)\log\theta_i^n$$

The determinant of the Fisher matrix, which indicates the sensitivity of the expected negative log likelihood function to the changes of $\vec{\alpha}$ is given by Allison (2018):

$$\det(\mathrm{Fisher}(\vec{\alpha})) = \mathrm{N}^k\left(\prod_{i=1}^{k}\psi_1(\alpha_i)\right)\left(1 - \psi_1(\kappa)\left(\sum_{i=1}^{k}\frac{1}{\psi_1(\alpha_i)}\right)\right),$$

where $\psi_1(z) = \frac{\partial^2}{\partial z^2}\log(\Gamma(\mathrm{z}))$ is the Trigamma function.

The general estimation method of Wallace and Freeman (1987) is used to derive the MML estimates of the free parameters of a multistate model with a specified Dirichlet prior $\mathrm{Dir}(\vec{\alpha})$:

$$\hat{\theta}_i = \left(\frac{x_i + \alpha_i - 0.5}{X + \sum_{i=1}^{k}\alpha_i - \frac{k}{2}}\right) \tag{4}$$

where x_i is the number of observations of the i-th state-transitions (corresponding to the i-th parameter) in the multistate model, X is the total number of state-transitions over all states and $\alpha_i \in \vec{\alpha}$ is the corresponding Dirichlet prior parameter (see Supplementary Sections S1). The inference of the optimal Dirichlet parameters is discussed below.

2.3.3 Estimation of Dirichlet parameters over finite-state strings

Let **A** be a set of N finite-state strings, each with k discrete states, and $\boldsymbol{\Theta} = [\vec{\Theta}^1, \vec{\Theta}^2, \ldots \vec{\Theta}^{\mathrm{N}}]$ be their corresponding set of N state parameter vectors each lying in a $(k - 1)$-simplex. The MML estimate $\vec{\alpha}^{\mathrm{MML}}$ of the Dirichlet parameters over the set **A** and **Θ** is the one that minimizes the two-part message length, as follows:

$$\underset{\vec{\alpha}}{\arg\min}\ (\mathrm{I}(\vec{\alpha}, \boldsymbol{\Theta}, \mathbf{A}) = \underbrace{\mathrm{I}(\vec{\alpha}) + \mathrm{I}(\boldsymbol{\Theta}|\vec{\alpha})}_{\text{First part}} + \underbrace{\mathrm{I}(\mathbf{A}|\boldsymbol{\Theta}, \vec{\alpha})}_{\text{Second part}})\ \text{bits} \tag{5}$$

$\mathrm{I}(\vec{\alpha})$ and $I(\boldsymbol{\Theta}|\vec{\alpha})$ deal with the message length terms required to transmit the Dirichlet and state parameters, respectively. Finally, $I(\mathbf{A}|\boldsymbol{\Theta}, \vec{\alpha})$ deals with the transmission of all finite-state strings in **A**, using the corresponding state parameters **Θ** and Dirichlet prior $\vec{\alpha}$ (see Supplementary Section S2).

The Dirichlet parameter estimator defined by Equation 5 is used on the subsets of structural alignments defined over PAM-based sequence distances in the range $n \in [1, 1000]$, to derive 1000 Dirichlet priors as a function of sequence-distance parameter n (see Section 3). These precomputed priors are employed to assign alignment state machine parameters when comparing any two sequences, as discussed below.

2.4 Practical considerations for protein sequence comparison and alignment using the MML framework

2.4.1 Estimation of null-model message length

Equation 3 in Section 2.2 involves the computation of null model message lengths of individual amino acid sequences, S and T. This in turn involves the statement of each amino acid symbol in these sequences using their respective null probabilities.

The MML estimate of the null probability for any amino acid symbol is computed over the large corpus of protein sequences derived from the Universal Protein Resource (UniProt-Consortium *et al.*, 2017). Specifically, $[\theta_1, \theta_2, \ldots \theta_{20}]$ corresponding to the 20-state amino acid strings are computed using Equation 4 by setting the 20-dimensional Dirichlet parameter vector to $\vec{\alpha} = (1, 1, \ldots, 1)$. This is same as using a uniform prior for the estimation of the null probabilities. We note that the estimation of null probabilities for amino acid sequences is one-off and independent of any particular sequence comparison.

The individual encoding of any sequence $\mathbf{Y} = (y_1 y_2 \ldots y_{|\mathbf{Y}|})$ first encodes the length $|\mathbf{Y}|$ over a universal integer code—we use Wallace Tree Codes (Wallace, 2005; Wallace and Patrick, 1993)—followed by successive statements of individual amino acids in the sequence using the null probability estimates:

$$\mathbf{I}_{\mathrm{NULL}}(\mathbf{Y}) = \mathbf{I}_{\mathrm{integer}}(|\mathbf{Y}|) + \sum_{i=1}^{|\mathbf{Y}|}\mathbf{I}_{\mathrm{NULL}}(y_i)\ \text{bits}$$

Applying the above equation to **S** and **T** yields the required null-model message length, $\mathrm{I}_{\mathrm{NULL}}(\langle\mathbf{S}, \mathbf{T}\rangle)$.

2.4.2 Estimation of alignment-model message length

Equation 1 in Section 2.2 gives the length of the two-part message to communicate any sequence pair $\langle\mathrm{S}, \mathrm{T}\rangle$ over an alignment hypothesis $\mathcal{A}$. The first part encoding deals with the statement of an alignment hypothesis $\mathcal{A}$ as a three-state string over the finite state machine with `m`, `i` and `d` states, as depicted in Figure 2. From the Dirichlet modeling exercise carried out in Section 2.3, for any stated PAM-n distance between $\langle\mathrm{S}, \mathrm{T}\rangle$ with $n \in [1, 1000]$, we have an associated Dirichlet prior. In this work, we chose the mode (i.e. point of maximum probability density) of the nominated Dirichlet distribution to define the nine state transition parameters. Since Dirichlet priors can be treated as common knowledge between the transmitter and receiver, the transmitter needs only to state the parameter n. (Under a uniform assumption of $n \in [1, 1000]$, the code length to state n is estimated as $\mathbf{I}(n) = \log_2(1000) = 9.965$ bits.) Once n is decoded, the receiver gains knowledge of the precise state transition probabilities used in the transmission of $\mathcal{A}$.

Consequently, the first part message length can be decomposed into its constituents as follow.

$$\mathrm{I}(\mathcal{A}) = \underbrace{\mathrm{I}(n)}_{\text{distance}} + \underbrace{\mathrm{I}_{\mathrm{integer}}(|\mathcal{A}|)}_{\text{length}} + \underbrace{\mathrm{I}(\mathcal{A}|n)}_{3-\text{state string}}\ \text{bits}$$

The second part encoding involves explaining the amino acid symbols in $\langle\mathrm{S}, \mathrm{T}\rangle$ using the alignment hypothesis $\mathcal{A}$ and the distance parameter n (both communicated in the first part). Each position in the alignment indicates one of the three possibilities: (i) an amino acid symbol $s_i \in \mathbf{S}$ is unmatched (`delete`); (ii) an amino acid symbol $t_j \in \mathbf{T}$ is unmatched (`insert`); and (iii) a pair of amino acid symbols $s_i \in \mathbf{S}$ and $t_j \in \mathbf{T}$ are matched (`match`).

The statement of unmatched amino acid symbols in **S** and **T** are carried out using their null probabilities as defined in Section 2.4.1. For the matched amino acid symbols s_i: t_j, given the knowledge of their sequence distance n, their joint probability is computed using PAM-n as:

$$\Pr(\langle s_i, t_j\rangle|n) = (\Pr(s_i)\Pr(t_j|s_i) + \Pr(t_j)\Pr(s_i|t_j))/2.$$

This joint probability estimate is a symmetric measure independent of the assumed order of the two sequences being considered. Negative logarithm of this joint probability gives the Shannon's information content (i.e. statement length) of communicating the matched amino acid pair. The total length of stating $\langle\mathbf{S}, \mathbf{T}\rangle$ over any alignment $\mathcal{A}$, $\mathrm{I}(\langle\mathrm{S}, \mathrm{T}\rangle|\mathcal{A})$, sums up the statement lengths over all matched and unmatched positions in $\mathcal{A}$, as described above.

2.4.3 Dynamic programming algorithm to compute the marginal probability of sequences

Here, we propose a $O(|\mathbf{S}||\mathbf{T}|)$-time and space algorithm to compute the marginal probability of $\langle \mathbf{S}, \mathbf{T}\rangle$, as defined in Section 2.2. The approach requires storing three dynamic programming algorithm (DPA) history matrices: $\mathrm{Tot}_{\mathtt{m}}$, $\mathrm{Tot}_{\mathtt{i}}$, and $\mathrm{Tot}_{\mathtt{d}}$. Each cell (*i*, *j*) in these matrices stores the negative logarithm of the marginal probability that the prefixes $\mathbf{S}_{1\dots i} : \mathbf{T}_{1\dots j}$ are related, by summing over all alignments ending in a `match`, `insert` and `delete` state, respectively. This can be efficiently computed using the negative LogSumExp (LSE) function under the dynamic programming recurrences given below, where LSE computes the logarithm of the sum of exponentials of its arguments:

$$\mathrm{Tot}_{\mathtt{m}}(i,j) = -\mathrm{LSE}\begin{cases} \mathrm{Tot}_{\mathtt{m}}(i-1,j-1) - \log(\Pr(\mathtt{m}|\mathtt{m})\Pr(\langle s_i, t_j\rangle|n)) \\ \mathrm{Tot}_{\mathtt{d}}(i-1,j-1) - \log(\Pr(\mathtt{m}|\mathtt{d})\Pr(\langle s_i, t_j\rangle|n)) \\ \mathrm{Tot}_{\mathtt{i}}(i-1,j-1) - \log(\Pr(\mathtt{m}|\mathtt{i})\Pr(\langle s_i, t_j\rangle|n)) \end{cases}$$

$$\mathrm{Tot}_{\mathtt{d}}(i,j) = -\mathrm{LSE}\begin{cases} \mathrm{Tot}_{\mathtt{m}}(i-1,j) - \log(\Pr(\mathtt{d}|\mathtt{m})\Pr(s_i)) \\ \mathrm{Tot}_{\mathtt{d}}(i-1,j) - \log(\Pr(\mathtt{d}|\mathtt{d})\Pr(s_i)) \\ \mathrm{Tot}_{\mathtt{i}}(i-1,j) - \log(\Pr(\mathtt{d}|\mathtt{i})\Pr(s_i)) \end{cases}$$

$$\mathrm{Tot}_{\mathtt{i}}(i,j) = -\mathrm{LSE}\begin{cases} \mathrm{Tot}_{\mathtt{m}}(i,j-1) - \log(\Pr(\mathtt{i}|\mathtt{m})\Pr(t_j)) \\ \mathrm{Tot}_{\mathtt{d}}(i,j-1) - \log(\Pr(\mathtt{i}|\mathtt{d})\Pr(t_j)) \\ \mathrm{Tot}_{\mathtt{i}}(i,j-1) - \log(\Pr(\mathtt{i}|\mathtt{i})\Pr(t_j)) \end{cases}$$

Thus, the negative logarithm of the marginal probability that $\mathbf{S} : \mathbf{T}$ are related is given by $-\mathrm{LSE}\{\mathrm{Tot}_{\mathtt{m}}(|\mathbf{S}|, |\mathbf{T}|),\ \mathrm{Tot}_{\mathtt{d}}(|\mathbf{S}|, |\mathbf{T}|),\ \mathrm{Tot}_{\mathtt{i}}(|\mathbf{S}|, |\mathbf{T}|)\}$, plus the constants associated with stating *n*, taking log(1000) bits, and the sum of lengths of the two sequences, $\mathrm{I}_{\mathrm{integer}}(|\mathbf{S}| + |\mathbf{T}|)$. Furthermore, in the above set of DPA recurrences, replacing –LSE function by min function yields the method to compute the best alignment hypothesis under our MML framework.

Finally, an approach similar to the bisection method is used to identify the corresponding optimal distance parameter *n* that yields the best estimate for $\mathrm{I}_{\mathrm{marginal}}(\langle \mathbf{S}, \mathbf{T}\rangle)$, and separately for $\mathrm{I}(\mathcal{A}^*, \langle \mathbf{S}, \mathbf{T}\rangle)$, searching over the domain $lo = 1 \le n \le 1000 = hi$. In each iteration, this approach truncates the search domain from [*lo*, *hi*] to either $\left[lo + \lfloor\frac{(hi-lo)}{4}\rfloor, hi\right]$ or $\left[lo, hi - \lfloor\frac{(hi-lo)}{4}\rfloor\right]$, by removing either the first or the last quarter of the domain after evaluating the marginal probability (or similarly, the best alignment hypothesis) at those two quarter points. The value of *lo* upon termination is taken to be the optimal estimate of *n*.

2.4.4 Marginal probability landscapes

In order to generate the marginal probability landscapes that allow users to visualize all competing alignments simultaneously, the following approach is used. For a given $\langle \mathbf{S}, \mathbf{T}\rangle$, the DPA is run in the forward direction (i.e. natural direction of the sequences). This yields the inferred distance parameter *n* and the dynamic programming history matrices $\overrightarrow{\mathrm{Tot}}_{\mathtt{m}}, \overrightarrow{\mathrm{Tot}}_{\mathtt{i}}$ and $\overrightarrow{\mathrm{Tot}}_{\mathtt{d}}$. Using the inferred distance parameter from the forward run, the DPA is run in the backward direction (i.e. reverse direction of the sequences), yielding the history matrices $\overleftarrow{\mathrm{Tot}}_{\mathtt{m}}, \overleftarrow{\mathrm{Tot}}_{\mathtt{i}}$ and $\overleftarrow{\mathrm{Tot}}_{\mathtt{d}}$.

A combined landscape matrix is derived by computing, for each cell (*i*, *j*), the negative LSE function over the following nine arguments: (i) $\overrightarrow{\mathrm{Tot}}_{\mathtt{m}}(i,j) - \log(\Pr(\mathtt{m}|\mathtt{m})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{m}}(i,j)$, (ii) $\overrightarrow{\mathrm{Tot}}_{\mathtt{m}}(i,j) - \log(\Pr(\mathtt{i}|\mathtt{m})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j)$, (iii) $\overrightarrow{\mathrm{Tot}}_{\mathtt{m}}(i,j) - \log(\Pr(\mathtt{d}|\mathtt{m})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j)$, (iv) $\overrightarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j) - \log(\Pr(\mathtt{m}|\mathtt{i})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j)$, (v) $\overrightarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j) - \log(\Pr(\mathtt{i}|\mathtt{i})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j)$, (vi) $\overrightarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j) - \log(\Pr(\mathtt{d}|\mathtt{i})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j)$, (vii) $\overrightarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j) - \log(\Pr(\mathtt{m}|\mathtt{d})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{m}}(i,j)$, (viii) $\overrightarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j) - \log(\Pr(\mathtt{i}|\mathtt{d})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{i}}(i,j)$ and (ix) $\overrightarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j) - \log(\Pr(\mathtt{d}|\mathtt{d})) + \overleftarrow{\mathrm{Tot}}_{\mathtt{d}}(i,j)$.

Any cell (*i*, *j*) in this landscape gives the product of marginal probabilities that the prefixes $\mathbf{S}_{1\dots i} : \mathbf{T}_{1\dots j}$ and suffixes $\mathbf{S}_{i+1\dots|\mathbf{S}|} : \mathbf{T}_{j+1\dots|\mathbf{T}|}$ are related. The marginal probability landscape provides insightful visualization of alignment relationships between two sequences. It also allows users to interactively generate competing alignments passing through any specified cell (*i*, *j*) in the landscape (see Section 3).

3 Results and discussion

3.1 Inferred Dirichlet priors

Using the inference method described in Section 2.3, we derived 1000 Dirichlet priors to model the observed distributions of state machine parameters, as a function of sequence distance $n \in [1, 1000]$ measured in terms of Point Accepted Mutations (Dayhoff *et al.*, 1978) (see Supplementary Section S3).

Figure 3(a) shows all the 1000 distributions of the free parameter $\Pr(\mathtt{m}|\mathtt{m})$ associated with the `match` state. This parameter influences the observed lengths of the matched blocks produced by the three-state machine. These lengths are geometrically distributed, and the probability of seeing a matched block of length *L* is $(1 - \Pr(\mathtt{m}|\mathtt{m})) \times \Pr(\mathtt{m}|\mathtt{m})^{L-1}$. The expected length of the matched block is given by $1/(1 - \Pr(\mathtt{m}|\mathtt{m}))$. Furthermore, due to the enforced symmetry of state transitions from `match` state to `insert` or `delete` states (see Section 2.3), we have: $1 - \Pr(\mathtt{m}|\mathtt{m}) = \Pr(\mathtt{i}|\mathtt{m})) + \Pr(\mathtt{d}|\mathtt{m})$. This value informs the probability of observing a gap in any alignment produced by the state machine. Figure 3(c) plots the expected probability of observing a gap as a function of distance *n*, when $\Pr(\mathtt{m}|\mathtt{m})$ is set to the mean (red) and mode (blue) values under the Dirichlet distributions shown in Figure 3(a). We notice the trend that the probability of a gap increases linearly with *n* in the range [1, 350], and stays relatively flat when $n > 350$. This linear trend agrees with the observations by Gonnet *et al.* (1992) and Benner *et al.* (1993).

On the other hand, Figure 3(b) shows a small selection of nine Dirichlet distributions, for specific values of $n \in \{1, 40, 80, 120, 160, 200, 400, 600, 800\}$, associated with the `insert` state parameters. (By symmetry, `insert` and `delete` state parameters are equivalent.) The heat map shows the inferred concentration (refer Section 2.3.2) of the probability density about the mode shown as a yellow dot (at the center of the heat map). The black dot shows the mean under the same distribution. The three corners (bottom-left, bottom-right, top) of each triangle (denoting the 2-simplex support for L1-normalized transition probabilities of `insert` state) show the points where $\Pr(\mathtt{i}|\mathtt{i}), \Pr(\mathtt{m}|\mathtt{i})$, and $\Pr(\mathtt{d}|\mathtt{i}))$ become exactly 1, while remaining two parameters become 0.

As *n* increases, it can be seen from Figure 3(b) that $\Pr(\mathtt{i}|\mathtt{i})$ also increases (see mean and mode—yellow and black dots in the plots—approach the bottom-left corner). This parameter influences the observed length of an `insert` (and, by symmetry, `delete`) block in an alignment. Again, these block lengths are geometrically distributed, with the probability of seeing a gap of length *L* defined by $(1 - \Pr(\mathtt{i}|\mathtt{i})) \times \Pr(\mathtt{i}|\mathtt{i})^{L-1}$, and whose expected gap length is given by $1/(1 - \Pr(\mathtt{i}|\mathtt{i}))$.

Figure 3(d) plots the expected gap length as a function of *n* with the mode estimate assigned to the parameter $\Pr(\mathtt{i}|\mathtt{i})$. On average, the expected gap length is about five amino acid residues for *n* in the range [1, 120], and barring a few outliers at *n* = [43, 44, 45, 91,

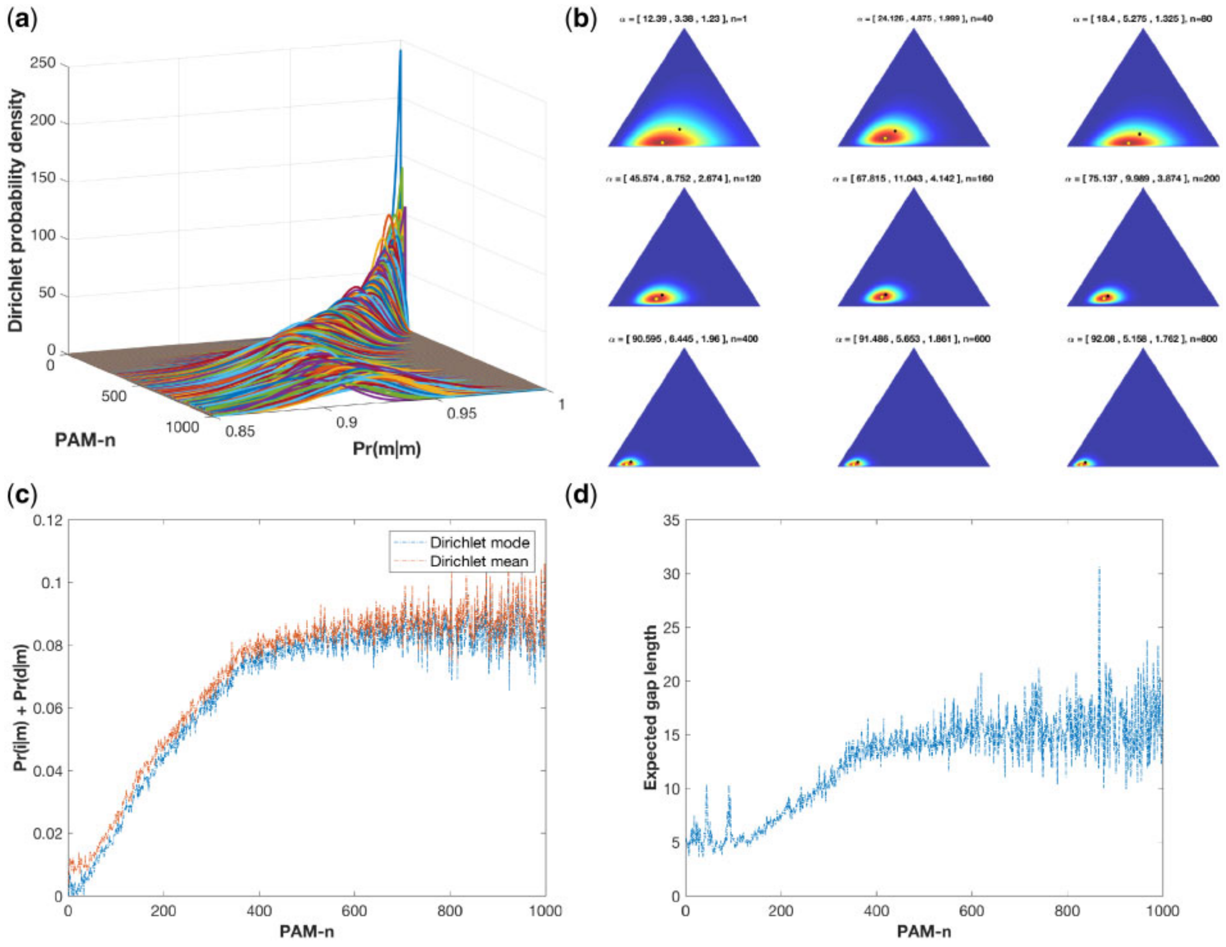

Fig. 3. Visualization of the inferred Dirichlet distributions modeling the three free parameters of the finite-state machine, and their associated statistics. (**a**) One-simplex distributions of $\Pr(\mathbf{m}|\mathbf{m})$ associated with the `match`, as a function of sequence-distance $n \in [1, 1000]$. (**b**) 2-simplex distributions of $\Pr(\mathbf{i}|\mathbf{i})$ and $\Pr(\mathbf{m}|\mathbf{i})$ associated with `insert` (and by symmetry, `delete`), as a function of sequence-distance $n \in \{1, 40, 80, 120, 160, 200, 400, 600, 800\}$. Since the values of $\Pr(\mathbf{i}|\mathbf{i})$ (estimated from the mode) are in the range of [0.81, 0.93] for $n \in [1, 1000]$, its simplex support shown above has been truncated for clarity to the range of [0.5, 1.0]. (**c**) The distribution of mean and mode values of $\Pr(\mathbf{i}|\mathbf{m}) + \Pr(\mathbf{d}|\mathbf{m})$ under the Dirichlet priors, as a function of n. (**d**) The distribution of expected gap lengths derived from the mode-estimate of $\Pr(\mathbf{i}|\mathbf{i})$ under the inferred Dirichlet priors

94], the trend is flat. Examining the source structural alignment data of these outliers (i.e. n-based alignment subsets on which the Dirichlet priors have been inferred), we find protein domain pairs with circularly permuted amino acid sequences and other pairs with plastic deformations in their structures. We note that a circular permutation between proteins results in a *non-sequential* relationship. Enforcing a *sequence alignment* on such a relationship yields regions that cannot be sequentially-aligned, which are then misinterpreted as long gaps. Similarly, domain pairs with plastic deformations have the same effect on gap lengths.

Further, in the range of $n \in [120, 350]$, we see in Figure 3(d) that the expected gap length increases linearly as a function of n. This contradicts the observation by Benner *et al.* (1993) who have stated that 'the distribution of gap size is essentially independent of the evolutionary distance between two sequences, with only a modest decrease in average gap length at increasing PAM distance'.

Finally, in the range of $n > 350$, the expected gap length in Figure 3(d) shows a noisy-yet-flat trend line, averaging about 13 to 15 amino acid residues. This trend more likely reveals the limits of applicability of PAM (Markov) matrix, and its convergence to its stationary distribution for large n (Dayhoff *et al.*, 1978). To test this, we measure the Kullback–Leibler (KL) divergence of each amino acid with varying PAM-n against its stationary distribution. We note that the KL-divergence between any two discrete probability distributions f and g over N mutually exclusive events, measures the additional number of bits required to encode samples drawn from f using g (i.e. it measures their relative Shannon entropy): $\mathrm{KL}(f, g) = \sum_{x=1}^{N} f(x) \log\left(\frac{f(x)}{g(x)}\right)$. By varying the value of $n \in [1, 1000]$, we computed the KL-divergence between each amino-acid's substitution probability distribution (i.e. each column of PAM-n) and the stationary distribution (i.e. the eigenvector associated with the dominant eigenvalue of 1 of PAM matrices). Figure 4 plots the KL-divergence for each amino acid with varying PAM-n. We observe that, for $n > 350$, most columns of PAM-n (i.e. amino acid substitution probabilities) converge to the stationary distribution. This indicates the limits of PAM's ability to differentiate amino acid substitutions at larger evolutionary distances.

3.2 Comparison with other programs on distantly related pairs of protein sequences

We used two types of benchmark datasets to test the performance of our sequence comparison framework. The first is a set of experimentally verified remote orthologs reported by Szklarczyk *et al.* (2012), containing a total of 877 protein sequence pairs spread over two groups. The second is the entire *twilight* zone dataset from SABmark (Van Walle *et al.*, 2005) containing 10 250 protein sequence pairs, spread over 209 groups.

Specifically, the dataset of Szklarczyk *et al.* (2012) include, in the first group, 405 pairs of human orthologous fungal mitochondrial proteins found in *Saccharomyces cerevisiae*, and in the second group, 472 pairs of orthologs found in *Schizosaccharomyces pombe*. Their average percentage sequence identity is reported as 27.7%, with about 40% of this dataset having pairs with <25% sequence identity and 36% between 25% and 35% sequence identity.

We ran our MML based alignment program in two separate modes. The first mode finds the best alignment hypothesis that minimizes the message length term given in Equation 1. This yields the $I(\mathcal{A}, \langle \mathbf{S}, \mathbf{T} \rangle)$ statistic, together with the information measures of alignment complexity $I(\mathcal{A})$ and alignment fidelity $I(\langle \mathbf{S}, \mathbf{T} \rangle | \mathcal{A})$. The second mode, which is the main focus of this work, is the one that estimates the negative logarithm of the marginal probability as per Equation 2 that yields $I_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ statistic. Both $I(\mathcal{A}, \langle \mathbf{S}, \mathbf{T} \rangle)$ and $I_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ are compared with their corresponding null model message length, $I_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ as per Equation 3, yielding the bits of 'Compression' statistic.

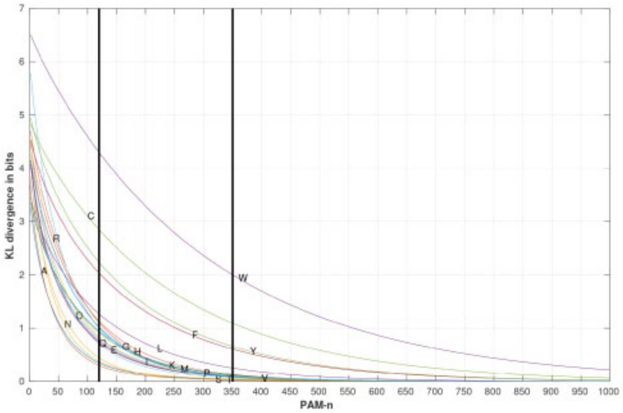

Fig. 4. Kullback–Leibler divergence between each amino acid related column in the PAM-*n* matrix and its stationary distribution, measured in bits. The black partitions define the [1, 120], [120, 350] and [350, 1000] regions corresponding to the PAM-*n* ranges where we see different trends for the expected gap length

Further, we compare the results of this dataset against seven widely-used protein sequence alignment programs: ClustalW (Larkin, 2007), CONTRAlign (Do *et al.*, 2006), KAlign (Lassmann and Sonnhammer, 2005), MAFFT (Katoh and Standley, 2013), MUSCLE (Edgar, 2004), ProbCons (Do *et al.*, 2005) and T-COFFEE (Notredame *et al.*, 2000). We evaluate the performance across all these program using the 'Compression' statistic. For each alignment produced by the programs, we infer the best parameters that minimize their $I(\mathcal{A}, \langle \mathbf{S}, \mathbf{T} \rangle)$ measure. This allows us to compare various message length terms against $I_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ to find its 'Compression'. We consider as *hits* (i.e. pairs that are correctly identified as related), only those alignments whose $I(\mathcal{A}, \langle \mathbf{S}, \mathbf{T} \rangle)$ is shorter than $I_{\text{NULL}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ (refer statistical significance test in Section 2.2). This allows us to compute the percentage of the total number of sequence pairs that pass the null hypothesis test for significance (%-Hits). Table 1 presents these results across the two benchmark groups of human remote orthologs (Szklarczyk *et al.*, 2012). In the table, we report the corresponding median values (across the whole group) against $I(\mathcal{A})$, $I(\langle \mathbf{S}, \mathbf{T} \rangle | \mathcal{A})$, and 'Compression' entries.

Our MML based marginal probability method has the highest percentage of hits (94.57% and 94.49% across the two groups, respectively). This is followed by our MML based method to identify the best alignment hypothesis (79.26% and 80.51%). MUSCLE and KAlign (both with 73.83% hits in the first group; 76.06% and 74.79% hits respectively in the second) are the next best performers.

We also compared the performance of all programs on the SABmark *twilight* zone sequences covering 10 250 sequence pairs. This is a substantially difficult dataset for most programs, especially those that rely on reporting a single alignment. Table 1 gives the results. As it can be seen, methods that rely on finding the best alignment under their respective criteria fare far worse than our MML based method that estimates the marginal probability of relationship between two sequences, and ascertains its statistical significance with the null model. The MML based marginal probability is able to identify significantly more number of hits (34.9%). A far second is our MML based best alignment approach (4.1%), followed by MUSCLE (2.5%).

Table 1. Comparison of various programs over two benchmark datasets: Human versus fungal ortholog groups reported by Szklarczyk *et al.* (2012), and SABmark (Van Walle *et al.*, 2005) twilight zone dataset

	Human remote orthologs of fungal mitochondrial proteins								SABmark proteins
	Human versus *S.cerevisiae* (405 pairs)				Human versus *S.pombe* (472 pairs)				Twilight (10 250 pairs)
Program	%-Hits	$I(\mathcal{A})$	$I(\mathbf{S},\mathbf{T}\|\mathcal{A})$	Compression	%-Hits	$I(\mathcal{A})$	$I(\mathbf{S},\mathbf{T}\|\mathcal{A})$	Compression	%-Hits
ClustalW	71.85	117.4	2678.1	82.6	74.58	110.3	2575.0	77.1	1.7951
CONTRAlign	71.11	117.8	2679.9	75.7	72.03	108.8	2573.6	70.7	2.3512
KAlign	73.83	134.3	2639.3	84.9	74.79	24.8	2533.1	81.8	2.4878
MAFFT	70.79	163.9	2622.1	81.8	71.91	150.9	2535.7	76.7	1.8187
MUSCLE	73.83	136.5	2639.3	86.1	76.06	129.8	2539.1	84.9	2.4976
ProbCons	70.12	143.5	2639.3	78.9	71.61	130.4	2543.9	75.9	1.6683
TCoffee	69.14	141.8	2640.9	76.5	71.19	130.8	2544.4	71.1	1.6390
MML ($I(\mathcal{A}^*, \langle \mathbf{S}, \mathbf{T} \rangle)$)	79.26	119.1	2666.5	93.7	80.51	109.5	2548.85	94.2	4.1399
MML ($I_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle)$)	94.57	N/A	N/A	125.0	94.49	N/A	N/A	117.9	34.9951

Note: The reported $I(\mathcal{A}), I(\langle \mathbf{S}, \mathbf{T} \rangle | \mathcal{A})$, and 'Compression' values are median statistics across the respective groups. These are information measures, reported in *bits*. (Only '%-Hits' statistic is shown for SABmark dataset. For full details of other statistics, see Supplementary Section S5.) N/A = Not Applicable.

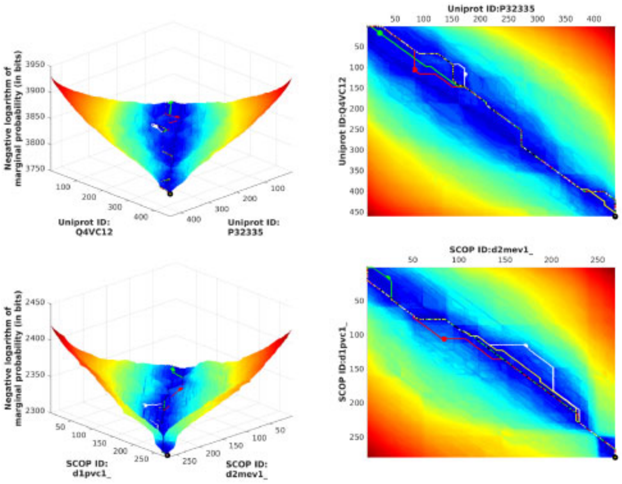

Fig. 5. Landscapes generated for two randomly selected pairs, one pair taken from the Human ortholog benchmark, and the other from SABmark benchmark. For more, see Supplementary Section S5

3.2.1 Marginal probability landscapes

As suggested in the introduction (Fig. 1), our work is able to produce the entire landscape of competing alignments based on marginal probability, for users to interactively study regions (and alignments) of interest, rather than just relying on the single best. Figure 5 gives a few more examples for randomly chosen pairs from our benchmark. This landscape can be queried to generate competing alignments, shown as paths in Figure 5. The Supplementary Section S5 provides a wider selection of landscapes comparing sequences across varying distances.

3.2.2 Computational complexity and run times

The asymptotic time complexity to align any two sequences in our MML framework grows as $O(|\mathbf{S}||\mathbf{T}|)$. Any specific competing alignment can be probed and reported in $O(|\mathbf{S}| + |\mathbf{T}|)$ time after the initial $O(|\mathbf{S}||\mathbf{T}|)$-effort to compute the marginal landscape. Using our distributed alignment program, the average run time required to compute $I(\mathcal{A}^*, \langle \mathbf{S}, \mathbf{T} \rangle)$ and $I_{\text{marginal}}(\langle \mathbf{S}, \mathbf{T} \rangle)$ for a sequence pair from SABmark benchmark (on a standard Linux-based computer) is approximately 1.5 s and 1.7 s, respectively.

Acknowledgements

The authors thank Françios Petitjean for helpful suggestions on parameterizing Dirichlet distributions.

Funding

D.S.'s PhD scholarship is funded by Monash University. A.K.'s research is funded by Australian Research Council's Discovery Project (DP150100894).

Conflict of Interest: none declared.

References

Allison,L. (2018) *Coding Ockham's Razor*. Springer, Cham, Switzerland.

Allison,L. *et al.* (1992) Finite-state models in the alignment of macromolecules. *J. Mol. Evol.*, **35**, 77–89.

Allison,L. *et al.* (1999) Compression and approximate matching. *Comput. J.*, **42**, 1–10.

Barton,G.J., and Sternberg,M.J. (1987) Evaluation and improvements in the automatic alignment of protein sequences. *Protein Eng.*, **1**, 89–94.

Bayes,T. (1763) A letter from the late Reverend Mr. Thomas Bayes, FRS to John Canton, MA and FRS. *Philos. Trans.*, **53**, 269–271.

Benner,S.A. *et al.* (1993) Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *J. Mol. Biol.*, **229**, 1065–1082.

Blake,J.D., and Cohen,F.E. (2001) Pairwise sequence alignment below the twilight zone1. *J. Mol. Biol.*, **307**, 721–735.

Collier,J.H. *et al.* (2017) Statistical inference of protein structural alignments using information and compression. *Bioinformatics*, **33**, 1005–1013.

Dayhoff,O.M. *et al.* (ed.) (1978) 22 a model of evolutionary change in proteins. In: *Atlas of Protein Sequence and Structure*. Vol. **5**. National Biomedical Research Foundation, Silver Spring, MD, pp. 345–352.

Do,C.B. *et al.* (2005) ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Res.*, **15**, 330–340.

Do,C.B. *et al.* (2006) CONTRAlign: discriminative training for protein sequence alignment. In: *Annual International Conference on Research in Computational Molecular Biology, Venice, Italy*. Springer, pp. 160–174.

Doolittle,R.F. (1986) *Of URFs and ORFs: A Primer on How to Analyze Derived Amino Acid Sequences*. University Science Books, CA, USA.

Durbin,R. *et al.* (1998) *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, New York, USA.

Edgar,R. (2004) MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.*, **32**, 1792–1797.

Fitch,W.M., and Smith,T.F. (1983) Optimal sequence alignments. *Proc. Natl. Acad. Sci. USA*, **80**, 1382–1386.

Gonnet,G.H. *et al.* (1992) Exhaustive matching of the entire protein sequence database. *Science*, **256**, 1443–1445.

Henikoff,S., and Henikoff,J.G. (1992) Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA*, **89**, 10915–10919.

Katoh,K., and Standley,D.M. (2013) Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, **30**, 772–780.

Kolmogorov,A.N. (1963) On tables of random numbers. *Sankhyā*, **25**, 369–376.

Larkin,M.A. *et al.* (2007) Clustal W and Clustal X version 2.0. *Bioinformatics*, **23**, 2947–2948.

Lassmann,T. and Sonnhammer,E.L. (2005) KAlign—an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, **6**, 298.

Lesk,A.M. (2017) *Introduction to Genomics*. Oxford University Press, New York, USA.

Levy Karin,E. *et al.* (2019) A simulation-based approach to statistical alignment. *Syst. Biol.* **68**, 252–266.

Löytynoja,A. and Goldman,N. (2008) Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, **320**, 1632–1635.

Murzin,A.G. *et al.* (1995) SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol.*, **247**, 536–540.

Notredame,C. *et al.* (2000) T-coffee: a novel method for fast and accurate multiple sequence alignment1. *J. Mol. Biol.*, **302**, 205–217.

Powell,D.R. *et al.* (2004) Modelling-alignment for non-random sequences. In: *Australian Conference on Artificial Intelligence*. Springer, Cairns, Australia, pp. 203–214.

Redelings,B.D. and Suchard,M.A. (2005) Joint bayesian estimation of alignment and phylogeny. *Syst. Biol.*, **54**, 401–418.

Rivas,E. and Eddy,S.R. (2015) Parameterizing sequence alignment with an explicit evolutionary model. *BMC Bioinformatics*, **16**, 406.

Rosenberg,M.S. (2009) *Sequence Alignment: Methods, Models, Concepts, and Strategies*. University of California Press, Los Angeles, USA.

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

Sumanaweera,D. *et al.* (2018) The bits between proteins. In: *2018 Data Compression Conference, Snowbird, USA*. IEEE, pp. 177–186.

Szklarczyk,R. *et al.* (2012) Iterative orthology prediction uncovers new mitochondrial proteins and identifies c12orf62 as the human ortholog of cox14, a protein involved in the assembly of cytochrome c oxidase. *Genome Biol.*, **13**, R12.

Trumpler,R.J. and Weaver,H.F. (1953) *Statistical Astronomy*. University of California Press, Berkeley and Los Angeles, USA.

UniProt-Consortium *et al*. (2017) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **45**, D158–D169.

Van Walle,I. *et al*. (2005) Sabmark-a benchmark for sequence alignment that covers the entire known fold space. *Bioinformatics*, **21**, 1267–1268.

Vingron,M. and Waterman,M.S. (1994) Sequence alignment and penalty choice: review of concepts, case studies and implications. *J. Mol. Biol.*, **235**, 1–12.

Wallace,C.S. (2005) *Statistical and Inductive Inference by Minimum Message Length*. Springer, New York, USA.

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

Wallace,C.S. and Freeman,P.R. (1987) Estimation and inference by compact coding. *J. R. Stat. Soc. Series B Methodol.*, **49**, 240–265. pages

Wallace,C.S. and Patrick,J.D. (1993) Coding decision trees. *Machine Learning*, **11**, 7–22.

Zhu,J. *et al*. (1998) Bayesian adaptive sequence alignment algorithms. *Bioinformatics*, **14**, 25–39.

Doc-Start

Supplementary Notes for:

“*Statistical Compression of Protein Sequences and Inference of Marginal Probability Landscapes over Competing Alignments using Finite State Models and Dirichlet Priors*”

Dinithi Sumanaweera, Lloyd Allison,* and Arun S. Konagurthu*

Faculty of Information Technology, Monash University, Clayton, VIC 3800 Australia.

* Correspondence: Arun Konagurthu (arun.konagurthu@monash.edu) or Lloyd Allison (lloyd.allison@monash.edu)

S1 MML ESTIMATION OF MULTISTATE MODEL PARAMETERS USING A DIRICHLET PRIOR

The Dirichlet probability density function, its negative log-likelihood function, and its Fisher matrix determinant were discussed in the main text. Below we discuss the mathematical derivation of the MML estimator for multistate model parameters using any specified Dirichlet prior.

A multistate model is defined on a data space containing k discrete states. Its parameter vector is denoted as $\vec{\Theta} = \{\theta_1, \theta_2, \dots, \theta_k\}$, where each θ_i gives the probability of the i-th state. Of these, only $k-1$ parameters are free, while the remaining parameter is dependent. Assume θ_k is dependent: $\theta_k = 1 - \sum_{i=1}^{k-1} \theta_i$. Thus, any $\vec{\Theta}$ is a point inside a unit $(k-1)$-Simplex [Allison, 2018].

Let D be multistate data we observed, containing x_i occurrences of any i-th discrete state. The MML (Wallace-Freeman) estimator [Wallace and Freeman, 1987, Wallace, 2005] of $\vec{\Theta}$ over these occurrences using the Dirichlet prior $\mathrm{Dir}(\vec{\alpha})$ requires the minimization of the message length term:

$$\mathrm{I}(\vec{\Theta}|\vec{\alpha}, D) = \underbrace{\mathrm{I}(\vec{\Theta}|\vec{\alpha})}_{\text{First part}} + \underbrace{\mathrm{I}(D|\Theta)}_{\text{Second part}}\ . \qquad (1)$$

Using the MML method of Wallace and Freeman [1987], each of the two parts can be expanded as:

$$\mathrm{I}(\vec{\Theta}|\vec{\alpha}) = -\log\left(f(\vec{\Theta}|\vec{\alpha})\right) + \frac{k-1}{2}\log\left(c_{k-1}\right) + \frac{1}{2}\log\left(\mathrm{det(Fisher}(\vec{\Theta}))\right)$$

and $\mathrm{I}(D|\vec{\Theta}) = \mathcal{L}(\vec{\Theta}) + \frac{k-1}{2}$, where, $\mathcal{L}(\vec{\Theta}) = -\sum_{i=1}^{k} x_i \log\left(\theta_i\right)$ is the negative log likelihood function of the multistate model, $\det(\mathrm{Fisher}(\vec{\Theta})) = (\sum_{i=1}^{k} x_i)^{k-1} / \prod_{i=1}^{k} \theta_i$ is the determinant of its Fisher information matrix, $f(\vec{\Theta}|\vec{\alpha})$ is the Dirichlet probability density function specified in the main text, and c_{k-1} is the Conway and Sloane [1984] lattice constant in $k-1$ dimensions.

As specified in the main text, the Dirichlet probability density function is given by

$$\begin{aligned} f(\vec{\Theta}|\vec{\alpha}) &= \frac{1}{B(\vec{\alpha})} \Pi_{i=1}^{k} (\theta_i)^{\alpha_i - 1} \\ \implies \log\left(f(\vec{\Theta}|\vec{\alpha})\right) &= -\log(B(\vec{\alpha})) + \sum_{i=1}^{k} (\alpha_i - 1)\log\theta_i \\ \implies \frac{\partial}{\partial\theta_i}\log\left(f(\vec{\Theta})|\vec{\alpha}\right) &= \frac{(\alpha_i - 1)}{\theta_i} - \frac{(\alpha_k - 1)}{\theta_k}, \end{aligned}$$

where $\theta_k = 1 - \sum_{i=1}^{k-1} \theta_i$.

Therefore, to find the optimal $\vec{\Theta}$ that minimises the two-part message length in Eqn. 1, we evaluate its extremum with respect to each $\vec{\theta_i}$, as:

$$\begin{aligned} \frac{\partial \mathrm{I}(\vec{\Theta}|\vec{\alpha}, D)}{\partial\vec{\theta_i}} &= -\frac{(\alpha_i - 1)}{\theta_i} + \frac{(\alpha_k - 1)}{\theta_k} - \frac{x_i + \frac{1}{2}}{\theta_i} + \frac{x_k + \frac{1}{2}}{\theta_k} = 0 \\ \implies \theta_i &= \frac{(\alpha_i + x_i - \frac{1}{2}) \times \theta_k}{\alpha_k + x_k - \frac{1}{2}} \qquad (2) \end{aligned}$$

However, since $\sum_{i=1}^{k} \theta_i = 1$, we have:

$$\begin{aligned} \sum_{i=1}^{k}\theta_i &= \sum_{i=1}^{k} \frac{(\alpha_i + x_i - \frac{1}{2}) \times \theta_k}{\alpha_k + x_k - \frac{1}{2}} = 1 \\ \implies \theta_k &= \frac{\alpha_k - x_k - \frac{1}{2}}{\sum_{j=1}^{k}\alpha_j + \sum_{j=1}^{k} x_j - \frac{k}{2}} \qquad (3) \end{aligned}$$

Substituting Eqn. 3 into Eqn. 2 yields the MML estimate of the probability of any i-th discrete state:

$$\theta_i{}^{\mathrm{MML}} = \left(\frac{x_i + \alpha_i - \frac{1}{2}}{\sum_{j=1}^{k} x_j + \sum_{j=1}^{k}\alpha_j - \frac{k}{2}}\right). \qquad (4)$$

Thus, using Eqn. 4, the statement cost of the data D, with $\{x_1, \dots, x_k\}$ occurrences of its k discrete states is given by:

$$\mathrm{I}(D|\Theta) = \sum_{i=1}^{k} x_i \times (-\log(\theta_i^{\mathrm{MML}})) \qquad (5)$$

S2 INFERRING DIRICHLET PRIORS OVER ALIGNMENT DATA

Let $\mathbf{A}_n$ be a set of alignment three-state strings (over **`match`**, **`insert`**, and **`delete`** states). The set of alignments $\mathbf{A}_n$ is curated such that each alignment is between amino acid sequences that are at the same estimated distance n (see section S3 and main text). Using $\mathbf{A}_n$, the goal is to infer the Dirichlet prior parameters $\vec{\alpha}$ that capture the observed distribution of the state machine parameters $\vec{\Theta}$ as a function of the sequence-distance parameter n.

As discussed in the main text, after constraining the three-state machine to have symmetric transition probabilities over **`insert`** and **delete** states, this inference problem reduces to finding the optimal Dirichlet distributions over a 1-simplex (to infer $\Pr(\mathtt{m}|\mathtt{m})$) and a 2-simplex (to infer $\Pr(\mathtt{i}|\mathtt{i})$ and $\Pr(\mathtt{m}|\mathtt{i})$).

For any k-1 simplex model, the $\vec{\alpha}$ that minimises the total message length: $\mathrm{I}(\vec{\alpha},\vec{\Theta},\mathbf{A}_n)$ yields the Dirichlet prior on the state parameters as a function of distance n. Note, $\vec{\alpha}$ contains k free parameters to be inferred. The MML two-part message length based objective function is:

$$\mathrm{I}(\vec{\alpha},\vec{\Theta},\mathbf{A}_n) = \underbrace{\mathrm{I}(\vec{\alpha}) + \mathrm{I}(\vec{\Theta}|\vec{\alpha})}_{\text{first part}} + \underbrace{\mathrm{I}(\mathbf{A}_n|\vec{\Theta},\vec{\alpha})}_{\text{second part}} \text{ bits} \tag{6}$$

The first part term $\mathrm{I}(\vec{\alpha})$ denotes the statement cost of the Dirichlet parameters $\vec{\alpha}$. Using Wallace and Freeman [1987] method of estimation, this term expands to:

$$\mathrm{I}(\vec{\alpha}) = -\log\left(h(\vec{\alpha})\right) + \frac{k}{2}\log\left(c_k\right) + \frac{1}{2}\log\left(\det(\mathrm{Fisher}(\vec{\alpha}))\right)$$

where $\mathrm{h}(\vec{\alpha})$ is the prior on the Dirichlet parameters $\vec{\alpha}$, and c_k is the Conway and Sloane [1984] lattice constant associated with k degrees of freedom (c_2=$\frac{5}{36\sqrt{3}}$ and c_3=$\frac{19}{192\times 2^{\frac{1}{3}}}$).

The Dirichlet parameters $\vec{\alpha}$ can be reparameterised into $(\kappa,\hat{\mu})$ denoting (concentration, mean) of the distribution, respectively – see main text. Thus, the prior on these parameters can be decomposed as:

$$\mathrm{h}(\vec{\alpha}) = \mathrm{h}(\kappa)\,\mathrm{h}(\hat{\mu}),$$

where $\mathrm{h}(\kappa)$ is the prior for the concentration parameter and $\mathrm{h}(\hat{\mu})$ is the prior for the L1-normalised mean vector.

In this work, we assume that $\hat{\mu}$ is uniform over the entire support. Thus, $\mathrm{h}(\hat{\mu})$ is simply the reciprocal of the volume of the (k$-$1) simplex. Specifically, for k=2 and k=3, the uniform prior yields $\frac{1}{\sqrt{2}}$ and $\frac{2}{\sqrt{3}}$, respectively.

On the other hand, κ controls the concentration of the Dirichlet probability density function (about the distribution's mode). For any y diffusing with d degrees of freedom, Wallace [2005] defined a well-behaved prior for y of the form:

$$g(y) = \frac{y^{d-1}}{(1+y^2)^{\frac{d+1}{2}}}$$

When $d = 2$, the normalization constant of $g(y)$ is computed as $\int_0^\infty g(y) = \int_0^\infty \frac{y}{(1+y^2)^{\frac{3}{2}}} = 1$. Thus, $h(\kappa) = g(\kappa)$ gives the prior probability distribution of κ for a 1-simplex Dirichlet. Similarly, when $d = 3$, computing the normalization constant gives $\int_0^\infty g_(y) = \int_0^\infty \frac{y^2}{(1+y^2)^2} = \frac{\pi}{4}$, which yields $\mathrm{h}(\kappa) = \frac{4}{\pi}g(\kappa)$.

For $\mathrm{I}(\vec{\Theta}|\vec{\alpha})$ term in the first part of Eqn. 6, we used the following alternative expansion (compared with the one specified in section S1) suggested by Wallace [2005]:

$$\mathrm{I}(\vec{\Theta}|\vec{\alpha}) = \left(\sum_{n=1}^{|\mathbf{A}_n|}\frac{1}{2}\log\left(1 + \frac{\det[\mathrm{Fisher}(\vec{\Theta})]\;{c_{k-1}}^{k-1}}{f(\vec{\Theta}\;|\vec{\alpha})^2}\right)\right) + \frac{k}{2}$$

This alternative formulation is especially useful to avoid underestimation of $\mathrm{I}(\vec{\Theta}|\vec{\alpha})$ when $\vec{\Theta}$ parameters are stated to high-precision [Wallace, 2005].

Finally, the second part of Eqn. 6 corresponds to transmitting all the alignment three-state strings in $\mathbf{A}_n$ using the parameters $\vec{\Theta}$ and $\vec{\alpha}$ stated in the first part. This requires:

$$\mathrm{I}(\mathbf{A}_n|\vec{\Theta},\vec{\alpha}) = \sum_{n=1}^{|\mathbf{A}_n|}\left(\mathrm{I}(\mathcal{A}_n\;|\vec{\Theta}) + \frac{k-1}{2}\right) \tag{7}$$

where, $\mathrm{I}(\mathcal{A}_n\;|\vec{\Theta})$ is computed as per Eqn. 5.

S3 DISCUSSION ON DIRICHLET INFERENCE

Choice of the data source

The main goal of this exercise is to identify a data set containing protein sequence-pairs, where each pair has a detectable amino acid sequence relationship between them, and as a collection, there is sufficient representation of these sequence-pairs at varying sequence distances, so that the Dirichlet models can be inferred as a function of these distances.

In this work, we randomly sampled a set of 118,384 protein domain-pairs from the Structural Classification of Proteins (SCOP v2.07) database [Murzin et al., 1995] and used this data set as the source collection for Dirichlet prior inference. Each domain in the collection is unique, in the sense that no protein domain repeated in the collection of domain-pairs.

The hierarchical organization of SCOP provides the convenience of identifying protein domains that have descended from a common ancestor with varying evolutionary distances. Each domain has a 4-level classification specifying the `Class`, `Fold`, `Superfamily` and `Family` that it belongs to. Since we are interested in domain-pairs with related amino acid sequences, we restrict our random selection of domain-pairs to the bottom two SCOP levels: '`superfamily`' and '`family`'. The domains within the same `family` are often closely related in their sequence, while those from the same `superfamily` but different `families` contain sequences which have diverged but with a detectable sequence signal.[1]

Using the random sampling method described below, we identified a source collection of protein domain-pairs, where 47,687 pairs are related at the `family` level and 70,697 pairs are related at the `superfamily` level. The full list of SCOP domain-pairs can be downloaded from: [here].

Random sampling method

Any domain-pair is randomly selected from SCOP using the procedure decribed below. The procedure uses the SCOP organization of domains within its hierarchical-classification tree. The internal nodes of this tree are associated with the 4-level classification of protein domains (specified above). Each domain in SCOP is organised as per its 4-level classification as a separate leaf-level node. Any traversal from the root to the leaf yields a domain.

[1] We note that other protein classification databases such as CATH or ECOD could have served us equally well for this exercise, but our choice of SCOP [Murzin et al., 1995] is mainly because it is built on a manually-curated database of protein domains, and is widely-used by protein scientists among the alternatives.

The sampling procedure involves traversing from the root to the leaf level by selecting, at each successive step of the traversal, a random child-node (a node from the next level). The choice of the child-node is a weighted random selection based on the number of leaves (i.e. domains) in the respective subtrees of the candidate child-nodes of any given node.

Thus, to identify domain-pairs within the same `superfamily` but from different families, the traversal proceeds from the root till the level of `superfamily` is reached. Then the weighted random sampling method selects two random domains (leaf-nodes) from different families (child-nodes), while considering only the SCOP superfamilies with ≥ 2 families. Similarly, to identify domain-pairs from the same `family`, when the traversal reaches the `family` level nodes, a pair of its children (leaf-nodes) are randomly selected, while only considering families with ≥ 2 domains.

Practical considerations

The quality of Dirichlet prior inference depends on: (1) the collection of protein sequence-pairs at varying evolutionary distances, and (2) the quality of their specified sequence relationships/alignments. Here, we discuss how these criteria are achieved in our work.

Since all domain-pairs chosen from SCOP have their associated three-dimensional (3D) atomic coordinate information, the pairs can be *structurally* aligned to decipher more reliable amino acid residue-residue correspondences from which their amino acid sequence alignments can be derived. The reliability of these correspondences are more trustworthy because functional constraints on evolving protein domains ensure that their 3D structures are far more conserved than their amino acid sequences [Lesk, 2010].

Thus, each sampled SCOP domain pair was structurally aligned using the MMLigner structural alignment program [Collier et al., 2017]. This generated a source collection of 118,384 structural alignments, from which their corresponding alignment three-state strings were generated and used in our Dirichlet inference exercise. From this point onwards, all obtained structural alignments are processed in terms of their amino acid sequence data.

Using PAM distance as a proxy for evolutionary distance between two sequences, for each alignment in our domain-pair collection, we infer the optimal PAM distance n that maximises the probability of the matched amino acids specified by the alignment (see main text). This allows us to group the alignments in the collection based on their integral n values, in the range of $[1, 1000]$. These n-based alignment groups are then used to infer a Dirichlet model over the unit 1-simplex, and separately a Dirichlet model over the unit 2-simplex, for $n \in [1, 1000]$.

The inferred n-based Dirichlet parameters can be downloaded from: [here]. The distribution of the number of alignments as a function of n can be downloaded from [here].

We emphasise that the modelling exercise discussed above entails a one-time preprocessing task. This task is independent of any specific protein sequences whose alignment is sought in our proposed MML-based alignment framework. In information-theoretic parlance, these parameters are a part of the 'codebook' of communication between an imaginary transmitter-receiver pair; a codebook should only contain common-knowledge or preconceived notions about the data being transmitted, and not the actual data itself. Consistent to this, in our alignment methodology, these inferred parameters are merely used and never recomputed.

Discussion on the computation of Dirichlet estimates We implemented and tested two methods to infer the Dirichlet parameters $\vec{\alpha}$ given any alignment data set. The first strategy is a gradient descent based approach, while the other is an exhaustive parameter sweep approach with fixed parameter-precision.

Gradient descent based approach converges rapidly, but its accuracy compared to the true parameter values is limited. On a standard Linux-based computer, it takes about 4 seconds to infer the Dirichlet parameters for an n-based group containing 100 alignments of domain-pairs. The exhaustive method does a parameter sweep with fixed precision, and hence is significantly more accurate than gradient descent. When the precision of $\vec{\alpha}$ is set 0.01 (two places after the decimal), the exhaustive method takes 24 seconds to find the optimal parameters (to the stated precision). Given this is a one-time exercise, we favoured the exhaustive approach for its accuracy.

S4 OVERVIEW TO THE RELATED LITERATURE

The common criteria for generating a sequence alignment encompasses a scoring scheme to quantify the relationship between protein sequences. Broadly, the matched regions are scored using an amino acid substitution matrix, while the unmatched (gap) regions are penalised. That is, a fixed scoring matrix and an associated gap penalty function form the inputs for any sequence alignment. In practice, besides the default settings, the choice of the scoring matrix and gap penalty function parameters are left for the users to fine-tune, and thus it remains a "trial and error based exercise" [Do et al., 2005, Vingron and Waterman, 1994]. Moreover, the manual fine-tuning is arduous due to the "non-convexity of ad hoc scoring functions" [Do et al., 2006].

Often, the users either tweak the default parameter setting of a program, or apply a commonly-used parameter settings prescribed by others. For instance, PAM250 and BLOSUM62 are commonly and widely used scoring matrices. As for the gap parameters, the default settings of many alignment programs include penalty combinations that adhere to the conventional choice of imposing a larger penalty for opening a gap and a smaller penalty for extending the same [Altschul et al., 1990]. However, different parameter settings yield radically different alignments [Vingron and Waterman, 1994]. Therefore several previous studies have sought to explore the parameter space in the quest for the optimal sequence alignment [Vingron and Waterman, 1994, Barton and Sternberg, 1987, Fitch and Smith, 1983, Blake and Cohen, 2001].

Specifically, Vingron and Waterman [1994] examined the tessellation of the parameter space with respect to gap penalties and a matrix bias over the PAM-250 scoring matrix. A flat surface in the tessellation encloses a set of different parameter settings that resulted in the same optimal alignment. While such an exhaustive search can provide an understanding of the alignment with respect to the parameter space, it is unarguably a tedious and inefficient exercise. ProbCons [Do et al., 2005] performed a Baum-Welch based Expectation-Maximization (EM) on benchmark datasets to train a pairwise Hidden Markov Model (pair-HMM). However, the use of a fixed scoring matrix is still an impediment to this process, and prone to over-fitting [Do et al., 2006]. A noteworthy solution was proposed in CONTRAlign[Do et al., 2006], where a pairwise alignment framework for parameter learning used conditional random fields. This estimates an optimal substitution matrix along with gap penalties, and demonstrate its effectiveness on a small data set of alignments.

However, a critically overlooked aspect in defining the parameter space is the relationship between evolutionary distance and insertion-deletion (indel) events. The evolutionary distance is often measured in terms of the sequence identity percentage between the sequences of interest. Alternatively, a Markov model of evolution such as the one used by PAM [Dayhoff et al., 1978] provides another proxy for evolutionary distance between two sequences. When two sequences become more diverged, structural stability constraints bear a greater precedence, involving more insertions and deletions to account for [Blake and Cohen, 2001]. This has been empirically observed where diverged sequence relationships can be modelled using smaller penalties for indel events [Blake and Cohen, 2001].

In typical use, the substitution parameters are independent of indel parameters [Redelings and Suchard, 2005], and only a few noticeable efforts have been made to make them work in concert. For instance, Blake and Cohen [2001] tested an alignment improvement by constructing a set of structural-superposition based amino acid substitution matrices for different evolutionary distance contexts and by obtaining the optimal gap penalties exhaustively for each, with the specific goal of improving remote pairwise homolog detection. Another study by Vogt et al. [1995] attempted to optimise gap penalties for a set of scoring matrices including a series of PAM, BLOSUM and Gonnet matrices. Several others [Gonnet et al., 1992, Chang and Benner, 2004, Benner et al., 1993] also explored the relationship between PAM-n and the length of gaps. Some of this work challenged the common notion of geometric distribution based gap length modelling, by empirically estimating a generalised Zipfian distribution, where the probability of a gap length is inversely-proportional to the gap length. Gonnet et al. [1992] and Benner et al. [1993] noted that the probability of observing a gap grows linearly with PAM-n. Moreover, they presented a relationship between PAM-n and the probability of a gap of certain length, claiming the Zipfian parameters to be independent of the evolutionary distance. Chang and Benner [2004] further observed that the Zipfian approximation does not change across a few bins spanned over PAM-10 and PAM-100, and also over the f2 measure (i.e., "the fraction of conserved nucleotides at the third position where the residue is conserved"), accounting for varying levels of selective pressure. To the contrary, experiments by Pascarella and Argos [1992] show an "exponential behaviour" for the expected "intervening sequence length".

S5 OTHER SUPPORTING INFORMATION

1. Detailed statistics comparing 8 programs (MML, ClustalW, CONTRAlign, KAlign, MAFFT, MUSCLE, ProbCons, T-Coffee) on:
   - Human fungal mitrochondrial proteins (remote ortholog) data set: [click here]
   - SABMark "Twilight" zone (twi) data set: [click here]
2. A selection of marginal probability landscapes: [click]
3. Download software and C++ code (GNU General Public License): [click here]

REFERENCES

L. Allison. *Coding Ockham's Razor*. Springer, 2018.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

G. J. Barton and M. J. Sternberg. Evaluation and improvements in the automatic alignment of protein sequences. *Protein Engineering, Design and Selection*, 1(2): 89–94, 1987.

S. A. Benner, M. A. Cohen, and G. H. Gonnet. Empirical and structural models for insertions and deletions in the divergent evolution of proteins. *Journal of Molecular Biology*, 229(4):1065–1082, 1993.

J. D. Blake and F. E. Cohen. Pairwise sequence alignment below the twilight zone1. *Journal of Molecular Biology*, 307(2):721–735, 2001.

M. S. Chang and S. A. Benner. Empirical analysis of protein insertions and deletions determining parameters for the correct placement of gaps in protein sequence alignments. *Journal of Molecular Biology*, 341(2):617–631, 2004.

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.

M. Dayhoff, R. Schwartz, and B. Orcutt. 22 a model of evolutionary change in proteins. In *Atlas of protein sequence and structure*, volume 5, pages 345–352. National Biomedical Research Foundation Silver Spring, MD, 1978.

C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou. ProbCons: probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15(2):330–340, 2005.

C. B. Do, S. S. Gross, and S. Batzoglou. CONTRAlign: discriminative training for protein sequence alignment. In *Annual International Conference on Research in Computational Molecular Biology*, pages 160–174. Springer, 2006.

W. M. Fitch and T. F. Smith. Optimal sequence alignments. *Proceedings of the National Academy of Sciences*, 80(5):1382–1386, 1983.

G. H. Gonnet, A. Cohen, and S. A. Benner. Exhaustive matching of the entire protein sequence database. *issues*, 3:10, 1992.

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

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.

S. Pascarella and P. Argos. Analysis of insertions/deletions in protein structures. *Journal of Molecular Biology*, 224(2):461–471, 1992.

B. D. Redelings and M. A. Suchard. Joint bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418, 2005.

M. Vingron and M. S. Waterman. Sequence alignment and penalty choice: Review of concepts, case studies and implications. *Journal of Molecular Biology*, 235(1):1–12, 1994.

G. Vogt, T. Etzold, and P. Argos. An assessment of amino acid exchange matrices in aligning protein sequences: the twilight zone revisited. *Journal of Molecular Biology*, 249(4):816–831, 1995.

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

C. S. Wallace and P. R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 240–265, 1987.