Discriminative Learning of Substitution Matrices and Gap Penalties for Pairwise Alignment of Biological Sequences

Michał Aleksander Ciach[1]
Department of Applied Biomedical Science, Faculty of Health Sciences, University of Malta, Msida, Malta

Elissavet Zacharopoulou
Department of Applied Biomedical Science, Faculty of Health Sciences, University of Malta, Msida, Malta

Michał Piotr Startek
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland

Błażej Miasojedow
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland

Panagiotis Alexiou
Department of Applied Biomedical Science, Faculty of Health Sciences, University of Malta, Msida, Malta

Abstract

Pairwise alignment scores are used to classify pairs of sequences in many areas of bioinformatics, including homology search, predicting interactions, or read mapping. The relative scores of different pairs strongly depend on the choice of a substitution matrix and gap penalties. However, current approaches for the estimation of these parameters typically describe patterns observed in a collection of ground-truth alignments instead of optimizing specifically for the task of classification. In this work, we present DiscrimAlign, a statistical model for discriminative learning of substitution matrices and gap penalties from a dataset of positive and negative pairs of unaligned DNA or amino acid sequences. The model links the alignment score of a sequence pair with the associated binary label through a logistic function and learns the parameters by likelihood maximization. We analyze theoretical properties of the model, derive and implement a learning procedure, study its performance in simulated experiments, and apply it to predict microRNA-target interactions. We show that sequence alignment with discriminative substitution matrices and gap penalties predicts the interactions comparably to black-box neural network classifiers while being more interpretable. An implementation of the model and reproducibility workflows are available at `https://github.com/BioGeMT/DiscrimAlign`.

2012 ACM Subject Classification Applied computing → Molecular sequence analysis

Keywords and phrases Sequence alignment, Substitution matrix, Logistic regression

Digital Object Identifier 10.4230/LIPIcs.WABI.2026.17

Supplementary Material *Software (Source Code)*: `https://github.com/BioGeMT/DiscrimAlign` [8] archived at `swh:1:dir:e70ee75187e06eb93adc1da6022502a80230d4b0`

Funding This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101244218. We acknowledge the BioGeMT Team (HORIZON-WIDERA-2022 Grant ID: 101086768).

Acknowledgements We thank Mr Dave Galea for the help in running simulations and preparing figures.

[1] Corresponding author

1 Introduction

Any kind of alignment of biological sequences requires a substitution matrix and penalties for gap opening and extension. The quality of the resulting alignment and its usefulness for any given task depends greatly on the choice of these parameters [2, 40]. Alignment of DNA sequences can often be performed with a simple three- or four-parameter scoring scheme; for example, the default option in the BLASTN suite (megablast algorithm) is to assign one point to a match, minus two to a mismatch, and minus one and a half for any gap (a so-called linear gap penalty). However, scoring schemes that account for differences in rates of different mutations can improve the performance of database searches [63]. For protein sequences, more complex scoring schemes are needed to account for similarities of amino acids, such as the PAM and BLOSUM series of substitution matrices. In particular, the BLOSUM62 matrix is the *de facto* standard choice [58, 65, 32, 55], at least in part due to being the default option in BLASTP [7, 62], MUSCLE [18], MAFFT [37], MMseqs2 [64] and others. Despite its popularity, it has been reported that, for some protein families, the pairwise alignment with BLOSUM62 aligns less than 50% of residues correctly (evaluated by comparing alignments of sequences and structures), partly because it averages out diverse substitution patterns across families [40, 32, 44]. Furthermore, there is a known bug in the software used to calculate the BLOSUM series, which was found to increase, rather than decrease, its performance in database searches [65]. Accordingly, there is still much to discover and improve about substitution matrices, and numerous alternative substitution matrices have been developed, both general-purpose and application-specific, both for amino acid and nucleotide sequences [69].

There are several approaches to the estimation of substitution matrices and gap penalties, with different optimization criteria and intended applications. Some matrices summarize substitutions observed in a collection of ground-truth alignments, others describe mutations observed in phylogenetic trees, and still others optimize the similarity between sequence alignment and structural alignment of proteins (see the Related work section). Despite the variety of approaches, to our knowledge, there is no approach specifically designed for a binary classification of pairs of sequences. Binary classification covers multiple applications of sequence alignment, including homology search, in which sequences are classified as either related or not; prediction of protein or RNA interactions; read mapping, in which a read can be classified as correctly or incorrectly placed on the genome; detection of duplicated sequences and sequence clustering; motif or domain searches; and others.

Our contribution. In this work, we explore the task of discriminative learning of substitution matrices and gap penalties for binary classification of pairs of sequences. We consider a data set $\mathcal{D} = \{(A_i, B_i, y_i) \colon i = 1, 2, \ldots, |\mathcal{D}|\}$, where A_i and B_i are sequences over a common alphabet (typically either DNA, RNA, or amino acid) and $y_i \in \{0, 1\}$ is a binary label associated with each pair. Our goal is to design a supervised learning procedure that can distinguish between positive pairs ($y_i = 1$) and negative pairs ($y_i = 0$) using alignment score. We analyze a logistic link between the sequences and the label, defined as

$$\mathbb{P}(Y_i = 1 |\ \alpha, \theta, A_i, B_i) = \left(1 + e^{-\alpha - S_\theta(A_i, B_i)}\right)^{-1}, \tag{1}$$

where θ denotes a vector of alignment parameters, $S_\theta(A, B)$ denotes the score of a given alignment model (global, local or glocal) between sequences A and B under parameters θ, and α is an intercept parameter that controls the probability of the pair being positive when the alignment score is zero.

Using a logistic function to classify pairs of sequences after the alignment score has been calculated under a fixed choice of parameters is a well-established idea [16]. In this work, we take a reverse approach: we derive an optimization procedure that, given a dataset of positive and negative examples of unaligned sequences, uses the logistic link to infer the parameters.

Our approach is applicable to local, global and semi-global (glocal) alignment schemes, linear and affine gap penalties, symmetric and asymmetric substitution matrices, and DNA and amino acid alphabets. We present a derivation of the optimization procedure, analyze the properties of the proposed method theoretically and in computational simulation experiments, and present a case study of prediction of microRNA-target interactions, where we show that the proposed method performs comparably to neural networks while retaining the interpretability of sequence alignment. A Python implementation of our method is available at `https://github.com/BioGeMT/DiscrimAlign`, together with Jupyter notebooks with simulation experiments and Python scripts with the case study.

2 Related work

The two main approaches to developing amino acid substitution matrices can be described as alignment-based and phylogeny-based. The alignment-based matrices, such as the original BLOSUM [27] and the bug-fixed RBLOSUM and CorBLOSUM series [65, 28, 21], are obtained by counting substitutions in a collection of ground-truth alignments, typically obtained from conserved blocks or by aligning protein structures. This approach has recently been used to obtain the CBM and CCF matrices for *Plasmodium* species [5], asymmetric dirAtPf matrices that capture the compositional bias of *Plasmodium* proteomes [3], the MOLLI series for *Mollicutes* bacteria [46], or the RHOD matrix for microbial rhodopsins [52]. Some alignment-based approaches combine the observed substitutions with other information, for example structural features, which can also account for correlated mutations [68, 32, 39].

Phylogeny-based matrices describe patterns of ancestral substitutions inferred from phylogenetic trees, and can either be in the form of log-odds, like the PAM [15] or VTML [51, 50] series, or in the form of instantaneous rates of mutations, like JTT [35], WAG [70] or LG [45, 12] matrices, the latter being used mostly in maximum likelihood phylogenetic inference. Both forms can be converted to one another by fixing the assumed evolutionary time and taking the matrix exponent of the instantaneous rate matrix to calculate the probabilities of mutations, which can then be transformed to log-odds [42, 50]. The main difference between phylogeny-based and alignment-based matrices is that the former focus on mutations occurring only between an ancestral and a descendent sequence; thus, if an ancestral character A changes to C in one descendant and to D in another, a phylogenetic approach like PAM will consider only the inferred mutations $A \to C$ and $A \to D$, while an alignment-based approach like BLOSUM will consider only the observed exchange $C \leftrightarrow D$ (see e.g. the example discussed in [15]). The phylogenetic approaches focus mostly on general-purpose rather than application-specific matrices, likely due to a considerably more involved estimation procedure [69]. Some approaches combine phylogenetic and alignment-based matrices using principal component analysis [72].

Apart from amino acid matrices, researchers also investigated scoring schemes for nucleotide sequences for homology search [63] and read mapping to reference genomes [20, 25]. Here, ground-truth alignments are often not available, necessitating a different approach. For example, LAST-TRAIN [25] iteratively realigns pairs of sequences and recalculates the matrix, a procedure mathematically based on a Markov Chain formalism of sequence

alignment [16, 19]. A similar approach has been used for disordered proteins [6, 59, 48]. Other applications also require substitution matrices obtained from pairs of sequences rather than multiple sequence alignments or trees, such as predicting protein-protein interactions [31], microRNA-target interactions [67], or DNA-to-protein alignment [73].

The selection of appropriate gap penalties for a given substitution matrix is often considered a separate task. The gap penalties for MOLLI matrices were optimized for ortholog prediction [46]. The POP procedure was developed to optimize the accuracy of alignments on a benchmark of structural alignments [17]. A notable exception is the Inverse Parametric Alignment (IPA) approach, which estimates the substitution matrix jointly with gap penalties by maximizing the score of a given collection of example alignments [40].

While the approaches described above are often used interchangeably, they have different goals and purposes. Phylogeny-based matrices describe the evolutionary process of substitutions, log-odds matrices summarize patterns observed in a collection of alignments, while IPA and POP optimize the similarity of structure-based and sequence-based alignments. While some theoretical guarantees exist for log-odds matrices applied to homology search with an ungapped local alignment [36], overall there is no guarantee that, for example, matrices describing ancestral substitutions, combined with gap penalties optimized for the structural accuracy of an alignment, will perform optimally when applied to the task of homology search with a local alignment. This motivates designing estimation procedures for clearly defined end goals with theoretical guarantees for convergence and optimality.

3 Methods

3.1 Mathematical properties of pairwise alignment score

In this work, we focus on pairwise sequence alignments that optimize a linear combination of parameters, denoted θ, and counts of associated features such as substitutions or gaps, denoted x and referred to as the *configuration* of the alignment. In the simplest setting, θ can be a vector of match, mismatch, and linear gap weights, in which case $\theta = (w_{\text{match}}, w_{\text{mismatch}}, w_{\text{gap}})$; the associated configuration vector is then $x = (x_{\text{match}}, x_{\text{mismatch}}, x_{\text{gap}})$, with the numbers of matches, mismatches and gaps in some given alignment. In a setting more commonly used for protein sequences, θ consists of either 190 or 400 substitution weights (depending on whether the substitution matrix is symmetric) and two gap penalties, and the alignment configuration x counts the associated substitutions, gap openings, and gap extensions.

Let Θ denote the space of possible parameters and let $\mathcal{X}(A, B)$ denote the set of all alignment configurations that can be obtained from A and B. As the scoring scheme influences the shape of the parameter space Θ, the alignment scheme (local, global or glocal) influences the space $\mathcal{X}(A, B)$. For example, the zero vector corresponding to an empty alignment always belongs to the space of configurations of a local alignment, but not of the global one (and, in general,$\mathcal{X}_{\text{global}}(A, B)$ is a proper subset of $\mathcal{X}_{\text{local}}(A, B)$).

The specification of Θ and $\mathcal{X}(A, B)$ is equivalent to the specification of alignment and scoring schemes. In this work, we will assume that $\Theta = \mathbb{R}^n$ for some n. Let $\langle x, y\rangle = \sum_i x_i y_i$ denote the inner product. Given a fixed configuration space $\mathcal{X}(A, B)$ and some $\theta \in \Theta$, the alignment score of sequences A, B can be defined as

$$S_\theta(A, B) = \langle \theta, x^* \rangle = \max_{x \in \mathcal{X}(A,B)} \langle \theta, x \rangle. \tag{2}$$

This kind of formulation has previously been used in research on Parametric and Inverse-Parametric Alignment [38, 40, 33]. In general, the space $\mathcal{X}(A, B)$ has a complex structure and its size grows exponentially with the length of the sequences [23]. The classical alignment

algorithm based on dynamic programming serves as a so-called *oracle* that can return an optimal configuration in polynomial time [40]. We remark that different alignments can have the same configuration, and different configurations can potentially yield the same optimal score, so x^* in the definition above denotes one of the optimal configurations that may itself correspond to several alignments.

The formulation of pairwise alignment with Equation (2) allows us to state the following Proposition, which describes the shape of regions of optimality for a given configuration. A set U is *convex* if $x, y \in U$ implies $px + (1-p)y \in U$ for all $p \in [0,1]$; a set U is a *cone* if $x \in U$ implies $sx \in U$ for all $s > 0$.

▶ **Proposition 1.** *Let $U_{A,B}(x) \subseteq \Theta$ denote the subset of the parameter space on which the alignment configuration x is optimal, and let $U^*_{A,B}(x) \subseteq \Theta$ denote the subset in which x is uniquely optimal. Then, $U_{A,B}(x)$ is a non-empty closed convex cone, and $U^*_{A,B}(x)$ is an open convex cone.*

This result was shown previously in the context of parametric alignment [54] and generalizes previous results about the decomposition of the parameter space [66, 24]. The proof is based on comparing x to every other configuration and expressing $U_{A,B}(x)$ as an intersection of the resulting hyperplanes (see the Appendix for details). The cone property is a mathematical expression of the fact that scaling all parameters by a positive constant does not change the relative scores of different alignments, thus preserving the optimal one [2]. The next Proposition describes the dependence of $S_\theta(A,B)$ on θ, and will later be necessary to derive generalized gradients for our optimization procedure. Let $||\cdot||$ denote the Euclidean (L2) norm. A function $f\colon \mathbb{R}^n \to \mathbb{R}$ is said to be Lipschitz continuous with a constant L if for any $\theta_1, \theta_2 \in \mathbb{R}^n$ we have $|f(\theta_1) - f(\theta_2)| \leq L||\theta_1 - \theta_2||$ (i.e. the growth of f is linearly bounded by L). A function f is said to be convex if, for any $\theta_1, \theta_2 \in \mathbb{R}^n$ and $p \in [0,1]$, we have $f(p\theta_1 + (1-p)\theta_2) \leq pf(\theta_1) + (1-p)f(\theta_2)$.

▶ **Proposition 2.** *The function $\theta \to S_\theta(A,B)$ is convex and Lipschitz continuous with the Lipschitz constant* $\max_{x \in \mathcal{X}(A,B)} ||x||$.

Both properties stem from the fact that for any vector x, the scalar product $\langle \theta, x \rangle$ is a convex and Lipschitz continuous function of θ, and the maximum operator over a finite number of functions preserves these properties (see the Appendix for a detailed proof). The Rademacher theorem states that Lipschitz-continous functions are differentiable almost everywhere (i.e. the set of points of non-differentiability has a Lebesgue measure zero). In the case of $S_\theta(A,B)$, if $\theta \in U^*(x)$ for some x, then $\nabla_\theta S_\theta(A,B)$ exists at θ and is equal to x. On the other hand, if two or more different configurations are optimal at θ, then $S_\theta(A,B)$ is non-differentiable at θ. It follows that the optimal configurations are unique for almost all values of θ.

The non-existence of gradients of $S_{A,B}(\theta)$ complicates the use of gradient descent methods for parameter learning. This problem has recently been overcome by designing differentiable alignment algorithms [57, 53, 41, 49]. In this work, we will derive a learning procedure for the standard alignment algorithms using *subdifferentials*, a generalization of gradients for non-differentiable convex functions.

▶ **Definition 3** (Adapted from [4]). *A subdifferential of a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ at a point $\theta \in \mathbb{R}^n$, denoted $\partial f(\theta)$, is the set of coefficient vectors of linear functions which bound f from below (also referred to as supporting hyperplanes of f at θ):*

$$\partial f(\theta) = \{v \in \mathbb{R}^n \colon \forall_{\theta' \in \mathbb{R}^n} \; f(\theta') \geq f(\theta) + \langle v, \theta' - \theta \rangle\}. \tag{3}$$

A subgradient of f at θ is any element of the subdifferential $\partial f(\theta)$.

For example, if $n = 1$ and $f(\theta) = |\theta|$, then $\partial f(\theta) = \{-1\}$ for all $\theta < 0$, $\partial f(\theta) = \{1\}$ for all $\theta > 0$, and $\partial f(0) = [-1, 1]$. Similarly to standard gradients, subgradients indicate the direction of growth of the function: for any $v \in \partial f(\theta)$, setting $\theta' = \theta + \lambda v$ in Equation (3) results in $f(\theta + \lambda v) - f(\theta) \geq \langle v, \lambda v \rangle = \lambda ||v||^2$, indicating that $f(\theta + \lambda v) \geq f(\theta)$ whenever $\lambda \geq 0$. If follows that θ is the minimum of $f(\theta)$ if and only if $0 \in \partial f(\theta)$ [4]. However, unlike standard gradients of differentiable functions, the definition does not guarantee that $-v$ indicates the direction in which the function decreases.

▶ **Proposition 4.** *Let $S_\theta(A, B)$ be the optimal score of alignment of sequences A and B with parameter vector θ. Let $\mathrm{Conv}(X)$ denote the convex hull of a set X, i.e. the minimal convex set containing all points from X (equivalently, the set of all convex combinations of points from X). Then, the subdifferential of the function $\theta \to S_{A,B}(\theta)$ is the convex hull of all configurations optimal at θ:*

$$\partial S_\theta(A, B) = \mathrm{Conv}(\{x^* \in \mathcal{X}(A, B) : \langle \theta, x^* \rangle = S_\theta(A, B)\}).$$

The Proposition follows from the fact that for a function f defined as $f(\theta) = \max_{i \in I} f_i(\theta)$, if I is a finite set and f_i are convex, then the subdifferential of f is the convex hull of subdifferentials of functions f_i active at θ, i.e. $\partial f(\theta) = \mathrm{Conv} \bigcup_i \{\partial f_i(\theta) \colon \ f_i(\theta) = f(\theta)\}$ (see e.g. Theorem 3.50 in ref. [4]). From the Proposition it follows that any optimal configuration is a subgradient of $S_\theta(A, B)$, and can be obtained by calculating the score with the dynamic programming algorithm, backtracking the optimal alignments, and calculating the configurations. If only one subgradient is necessary for any given θ, it can be calculated without backtracking by keeping track of the choices during the calculation of the score with the dynamic programming algorithm.

3.2 The likelihood function of the proposed model

Let $p(\alpha, \theta;\ A_i, B_i) = \mathbb{P}(Y_i = 1|\ \theta, \alpha, A_i, B_i)$. In the same way as in logistic regression models, we assume that the likelihood of parameters α and θ given a dataset $\mathcal{D}$ is equal to the product of probabilities defined by Equation (1):

$$\mathcal{L}(\alpha, \theta; \mathcal{D}) = \prod_{i=1}^{|\mathcal{D}|} \mathbb{P}(Y_i = y_i|\ \alpha, \theta, A_i, B_i) \tag{4}$$

$$= \prod_{i=1}^{|\mathcal{D}|} p(\alpha, \theta;\ A_i, B_i)^{y_i} (1 - p(\alpha, \theta;\ A_i, B_i))^{1 - y_i} \tag{5}$$

Given a dataset $\mathcal{D}$, the parameters α and θ can be estimated by maximizing the likelihood, equivalent to maximizing the log-likelihood $l(\alpha, \theta; \mathcal{D}) = \ln \mathcal{L}(\alpha, \theta; \mathcal{D})$. Let $\mathcal{X}_i = \mathcal{X}(A_i, B_i)$. After taking the logarithm of Equation (5), substituting Equations (1) and (2), and doing some algebraic manipulations, the log-likelihood of the proposed model can be expressed as

$$l(\alpha, \theta; \mathcal{D}) = \sum_{i=1}^{|\mathcal{D}|} \Big(y_i \ln p(\alpha, \theta;\ A_i, B_i) + (1 - y_i) \ln(1 - p(\alpha, \theta;\ A_i, B_i)) \Big)$$

$$= \sum_{i=1}^{|\mathcal{D}|} y_i \Big(\alpha + \max_{x \in \mathcal{X}_i} \langle \theta, x \rangle \Big) - \sum_{i=1}^{|\mathcal{D}|} \ln \Big(1 + \exp(\alpha + \max_{x \in \mathcal{X}_i} \langle \theta, x \rangle) \Big)$$

The main difference between the proposed model and the standard logistic regression is the presence of the maximum operator within the log-likelihood. In fact, fixing the alignments

reduces the model to a standard logistic regression. Namely, given a list of fixed alignments $\mathcal{A}_i = \mathcal{A}(A_i, B_i)$, the probability of a pair being positive can be expressed as

$$\mathbb{P}(Y_i = 1|\ \alpha, \theta, \mathcal{A}_i) = \left(1 + e^{-\alpha - \langle \theta, x(\mathcal{A}_i)\rangle}\right)^{-1} \tag{6}$$

This equation corresponds to a standard logistic regression model with intercept α, vector of parameters θ, and independent variables $x(\mathcal{A}_i)$. If a list of ground-truth alignments is available, this model can be used to estimate the parameters, but does not guarantee that the alignments will be optimal for parameters estimated this way. Conversely, the model in Equation (5) uses unaligned pairs of sequences and estimates the alignments jointly with the parameters, allowing both to change during optimization.

Note that functions $y_i(\alpha + S_\theta(A_i, B_i))$ are convex as affine transformations of a convex function $S_\theta(A_i, B_i))$, functions $\ln(1 + \exp(\alpha + S_\theta(A_i, B_i)))$ are convex as compositions of a strictly increasing convex function $z \to \ln(1 + \exp(\alpha + z))$ and convex functions $S_\theta(A_i, B_i)$, and convexity is preserved under summation. The log-likelihood of the proposed model is therefore a difference of two convex functions, which may be neither convex nor concave (as opposed to the concave log-likelihood of the standard logistic regression). However, it is concave locally almost everywhere, i.e. for almost all $\theta \in \mathbb{R}^n$ (in the sense of Lebesgue measure) there exists an open neighborhood $U \ni \theta$ such that the function restricted to U is concave (i.e. $-l(\alpha, \theta; \mathcal{D})_{|\theta \in U}$ is convex).

▶ **Proposition 5.** *The function $l(\alpha, \theta; \mathcal{D})$ is smooth and locally concave with respect to θ almost everywhere.*

The Proposition stems from the fact that the log-likelihood is differentiable almost everywhere, and in the points of differentiability it locally reduces to the log-likelihood of a standard logistic regression. A detailed proof is presented in the Appendix. Since $l(\alpha, \theta; \mathcal{D})$ is not globally concave, the model may not be uniquely identifiable (i.e. there may be more than one maximizer of the log-likelihood). An intuitive explanation of this phenomenon is that the added flexibility of alignment means that different substitution matrices may perform equally well in classifying pairs of sequences, as the differences in substitution matrices can potentially be offset by adjusting the alignments. As a consequence, we will look for example maximizers $\alpha^*, \theta^* \in \arg\max_{\alpha,\theta} \mathcal{L}(\alpha, \theta; \mathcal{D})$. An exact characterization of the geometry of the set of maximizers remains an open problem. The following Proposition states that this set is small in the sense of Lebesgue measure:

▶ **Proposition 6.** *The set of maximizers $\arg\max_{\alpha,\theta} \mathcal{L}(\alpha, \theta; \mathcal{D})$ has Lebesgue measure zero.*

The Proposition follows from the fact that Θ is composed of a finite number of open subsets on which $l(\alpha, \theta; \mathcal{D})$ is concave (and thus has a unique maximizer), and the boundaries of these subsets, which have measure zero. The next Proposition states that a necessary condition for the boundedness of the set of maximizers is the lack of a complete separation of the labels, similarly as in the standard logistic regression [1]:

▶ **Proposition 7.** *Assume there exists a threshold $c \in \mathbb{R}$ and a parameter vector θ such that $S_\theta(A_i, B_i) > c$ for any i such that $y_i = 1$ and $S_\theta(A_i, B_i) < c$ for any i such that $y_i = 0$. Then, $\sup_{\alpha,\theta} \mathcal{L}(\alpha, \theta; \mathcal{D}) = 1$. Furthermore, for any sequence α_k, θ_k such that $\lim_{k\to\infty} \mathcal{L}(\alpha_k, \theta_k; \mathcal{D}) = 1$, the alignment parameters diverge to infinity: $||\theta_k|| \to \infty$.*

Proof. Let $\alpha_k = -kc$, $\theta_k = k\theta$ and observe that $S_{\theta_k}(A_i, B_i) = kS_\theta(A_i, B_i)$. We have

$$\mathbb{P}(Y_i = 1|A_i, B_i, \alpha_k, \theta_k) = (1 + \exp(k(c - S_\theta(A_i, B_i))))^{-1},$$
$$\mathbb{P}(Y_i = 0|A_i, B_i, \alpha_k, \theta_k) = (1 + \exp(-k(c - S_\theta(A_i, B_i))))^{-1}.$$

Since $c - S_\theta(A_i, B_i) < 0$ for $y_i = 1$ and $c - S_\theta(A_i, B_i) > 0$ for $y_i = 0$, for every i we have $\mathbb{P}(Y_i = y_i | A_i, B_i, \alpha_k, \theta_k) \to 1$ as $k \to \infty$, which proves the first part of the Proposition. The second part follows from the fact that the likelihood function attains 1 only if all the logistic probabilities attain 1, which can only happen in the limit of infinite alignment scores. ◀

We also conjecture, motivated by similar results in standard logistic regression [1] and the results of our numerical experiments, that a so-called overlap of points (i.e. a dataset in which any set of parameters results in at least one misclassified label) is sufficient for there to exist finite maximizers α^*, θ^* (and, possibly, for the set of maximizers to be bounded). In what follows, we will assume that the set of maximizers of $l(\alpha, \theta; \mathcal{D})$ is non-empty.

3.3 The Clarke subdifferential of the log-likelihood

As opposed to the function $\theta \to S_\theta(A, B)$, the functions $\ln p(\theta)$ and $\ln(1 - p(\theta))$ may not be convex nor concave. This is because the logarithm of the logistic function is concave, while the alignment score is convex, and a composition of a concave and a convex function may be neither. Thus, these functions may not have standard subdifferentials, and in order to analyze them we need to use a further generalization called the Clarke subdifferentials [9, 10]. A function is said to be locally Lipschitz continuous if, for every $\theta \in \mathbb{R}^n$, there exists an open neighbourhood $U \ni \theta$ such that f is Lipschitz continuous on U.

▶ **Definition 8** (Adapted from [9]). *For a locally Lipschitz function $f\colon \mathbb{R}^n \to \mathbb{R}$, the Clarke subdifferential at a point θ, denoted $\partial^C f(\theta)$, is defined as:*

$$\partial^C f(\theta) = \mathrm{Conv}\{v \in \mathbb{R}^n \colon \exists_{\theta_n \to \theta}\ \nabla f(\theta_n) \to v\}. \tag{7}$$

Intuitively, one may think about the Clarke subdifferential as the set of all gradients of f, and their convex combinations, around θ. This intuition is substantiated by the fact that locally Lipschitz functions are differentiable almost everywhere, so their gradients exist almost everywhere. Points θ such that $0 \in \partial^C f(\theta)$ are called *Clarke-stationary* or *Clarke-critical*. If f has a local minimum or maximum at θ, then θ is a Clarke-stationary point, but the converse may not hold (Proposition 2.3.2 in [10]). As opposed to standard calculus, the rules of Clarke calculus often provide only inclusions rather than equalities.

▶ **Lemma 9** (Consequence of Theorem 2.3.9 in [10]). *Let $f, g\colon \mathbb{R}^n \to \mathbb{R}$ be locally Lipschitz, and let $h\colon \mathbb{R} \to \mathbb{R}$ be locally Lipschitz and continuously differentiable. Let $\oplus$ denote the Minkovsky sum of sets defined as $A \oplus B = \{a + b \colon a \in A, b \in B\}$. Then,*

$$\partial^C(f(\theta) + g(\theta)) \subseteq \partial^C f(\theta) \oplus \partial^C g(\theta) \tag{8}$$

$$\partial^C h(f(\theta)) = h'(f(\theta))\partial^C f(\theta) \tag{9}$$

We remark that the chain rule above works only in the special case that h is a continuously differentiable function of a single variable; in general, only an inclusive chain rule is satisfied with $\partial^C h(f(x)) \subseteq h'(f(x))\partial^C f(x)$. In what follows, we will denote the Clarke subdifferential with respect to θ as ∂^C_θ, treating the parameter α as a constant.

▶ **Proposition 10.** *The Clarke subdifferentials of $\ln p(\alpha, \theta; A, B)$ and $\ln(1 - p(\alpha, \theta; A, B))$ with respect to θ are equal to*

$$\partial^C_\theta \ln p(\alpha, \theta; A, B) = (1 - p(\alpha, \theta; A, B))\partial S_\theta(A, B),$$
$$\partial^C_\theta \ln(1 - p(\alpha, \theta; A, B)) = -p(\alpha, \theta; A, B)\partial S_\theta(A, B).$$

The proof is based on showing that the functions are locally Lipschitz continuous and applying Lemma 9. Details are presented in the Appendix. Using a variable $y \in \{0, 1\}$, we can combine both equations from the Proposition into

$$\partial_\theta^C \big(y \ln p(\alpha, \theta; A, B) + (1 - y) \ln(1 - p(\alpha, \theta; A, B))\big) = (y - p(\alpha, \theta; A, B))\partial S_\theta(A, B).$$

This equality follows from the fact that any choice of $y \in \{0, 1\}$ reduces this equation to either of the two from the Proposition. From Lemma 9 we now have

$$\partial_\theta^C \; l(\theta; \mathcal{D}) \subseteq \bigoplus_{(A_i, B_i, y_i) \in \mathcal{D}} (y_i - p(\alpha, \theta; A_i, B_i))\, \partial S_\theta(A_i, B_i).$$

In general, $\partial_\theta^C \; l(\theta; \mathcal{D})$ is a proper subset of the Minkovsky sum on the right-hand side. This stems from the fact that the Minkovsky sum allows us to select any optimal configuration for each sequence pair, while the Clarke subdifferential requires a single convergent sequence of parameters $\theta_n \to \theta$ for which the configurations are jointly uniquely optimal:

▶ **Lemma 11.** *Let* $\mathcal{B}(\theta; \mathcal{D}) = \{(x_1, x_2, \ldots, x_{|\mathcal{D}|}) \colon x_i \in \partial_\theta S(\theta; A_i, B_i)\}$. *Then,*

$$\exists_{\theta_n \to \theta} \forall_n \mathcal{B}(\theta_n; \mathcal{D}) = \{(x_1, \ldots, x_{|\mathcal{D}|})\} \;\Rightarrow\; \sum_{i=1}^{|\mathcal{D}|} (y_i - p(\alpha, \theta; A_i, B_i)) x_i \in \partial_\theta^C l(\alpha, \theta; \mathcal{D})$$

A proof of the Lemma is presented in the Appendix. The Lemma can be used to show the following Proposition. We remark that both the Lemma and the Proposition state an inclusion rather than equality due to the technical fact that the sum of non-differentiable functions may be differentiable. This issue is discussed further in the proof of the Lemma.

▶ **Proposition 12.** *Let* $(x_1, x_2, \ldots, x_{|\mathcal{D}|}) \in \mathcal{B}(\theta; \mathcal{D})$. *For* $x_i \in \mathcal{X}_i$, *let* $U_i^*(x_i) \subseteq \Theta$ *be the set of parameters* θ *for which* x_i *is a uniquely optimal configuration for* A_i, B_i. *Then,*

$$\bigcap_{i=1}^{|\mathcal{D}|} U_i^*(x_i) \neq \emptyset \;\Rightarrow\; \sum_{i=1}^{|\mathcal{D}|} (y_i - p(\alpha, \theta; A_i, B_i)) x_i \in \partial_\theta^C l(\alpha, \theta; \mathcal{D})$$

We note that the above subgradients of $l(\alpha, \theta; \mathcal{D})$ have an intuitive interpretation. The weights $y_i - p(\alpha, \theta; A_i, B_i)$ are the errors of the model, which are positive for the positive pairs and negative for the negative ones. Thus, they prioritize increasing the parameters corresponding to common substitutions in low-scoring positive pairs and decreasing the ones corresponding to common substitutions in high-scoring negative pairs. We also note a connection to the problem of separability: if there exists a sequence θ_k such that $\lim_{k\to\infty} p(\alpha, \theta_k; A_i, B_i) = y_i$ for all i, then $\lim_{k\to\infty} \partial_\theta^C l(\alpha, \theta_k; \mathcal{D}) = 0$, resulting in "Clarke-stationarity in infinity".

We remark that functions whose gradients point towards the direction of increase are termed *pseudoconvex*. The following Proposition establishes that $\ln p(\alpha, \theta; A, B)$ is a Clarke-pseudoconvex function and $\ln(1 - p(\alpha, \theta; A, B))$ is a Clarke-pseudoconcave function. Whether $\partial_\theta^C l(\alpha, \theta; \mathcal{D})$ admits a similar property remains an open question.

▶ **Proposition 13.** *Let* $d_{\ln p} \in \partial_\theta^C \ln p(\alpha, \theta; A, B)$ *and* $d_{\ln(1-p)} \in \partial_\theta^C \ln(1 - p(\alpha, \theta; A, B))$. *Then, for any* $\theta' \in \Theta$*:*

- *If* $\langle d_{\ln p}, \theta' - \theta \rangle \geq 0$, *then* $\ln p(\alpha, \theta'; A, B) \geq \ln p(\alpha, \theta; A, B)$,
- *If* $\langle d_{\ln(1-p)}, \theta' - \theta \rangle \leq 0$, *then* $\ln(1 - p(\alpha, \theta'; A, B)) \leq \ln(1 - p(\alpha, \theta; A, B))$.

The proof is based on an observation that $d_{\ln p}$ is proportional to a subgradient of the alignment score, and thus points in the direction of increasing alignment score, which in turn is the direction of increase of $\ln p(\alpha, \theta; A, B)$ for a fixed α.

3.4 Fitting the model with the subgradient method

While there are specialized methods of optimizing differences of convex functions such as the log-likelihood of the proposed model [29, 34], in this work we present a general subgradient method because it results in a particularly simple algorithm that can be implemented in a few lines of computer code. The subgradient method is a generalization of gradient ascent that essentially replaces the gradient with any subgradient. While the definition of subgradients does not guarantee that the function value increases in each step (hence the name "subgradient method" rather than "subgradient ascent"), the method converges to a Clarke stationary point under some general assumptions. We will first present the conditions for finding optimal θ assuming that the α parameter is known. Below, "almost surely" means "with probability 1" and refers to the randomness of the trajectory, and "limit points" of θ_k refer to limits of subsequences of θ_k (which may exist even if θ_k does not converge to a single point).

▶ **Theorem 14.** *For $k \geq 1$, let ξ_k be an n-element vector of independent random variables with zero expected value and standard deviation σ_k, where n is the dimension of Θ. Let $\theta_0 \in \Theta$ and let the sequence θ_k be defined as*

$$\theta_{k+1} = \theta_k + \eta_k g_k + \xi_k \text{ for } g_k \in \partial^C_\theta l(\alpha, \theta_k; \mathcal{D}) \tag{10}$$

Assume that $\eta_k > 0, \sum \eta_k = \infty, \sum \eta_k^2 < \infty, \sigma_k/\eta_k \to 0$, and that the sequence θ_k is bounded. Then, almost surely, every limit point of θ_k is a Clarke-stationary point of $l(\alpha, \theta; \mathcal{D})$.

The proof is based on showing that the sequence of parameters satisfies Assumptions C and D from ref. [14], and presented in the Appendix. An example sequence of step-lengths that satisfies the assumptions of the Theorem is $\eta_k = \eta/k$ for some $\eta > 0$ and $\sigma_k = \sigma/k^2$ for some $\sigma \geq 0$. Note that while $\sigma_k \equiv 0$ also satisfies the assumptions of the Theorem, some amount of random noise may be beneficial in practice e.g. to avoid local stationary points. The assumption $\sigma_k/\eta_k \to 0$ can be considerably weakened to still preserve convergence, but having the random noise vanish faster than the step size allows the method to refine the estimates without random perturbations in final iterations. Finally, while subgradients of the log-likelihood can be theoretically constructed using Proposition 12, it might be computationally expensive in practice. However, if ξ_k are continuous random variables, then for $k \geq 1$ the log-likelihood is differentiable at θ_k almost surely. This reduces the condition $g_k \in \partial^C_\theta l(\alpha, \theta_k; \mathcal{D})$ to $g_k = \nabla_\theta l(\alpha, \theta_k; \mathcal{D}) = \sum_i (y_i - p(\alpha, \theta_k; A_i, B_i))x_i$ almost surely.

We remark that the update step above uses a standard gradient of the log-likelihood. The algorithm can be derived heuristically by simply differentiating the log-likelihood where possible and using the gradient in a heuristic gradient ascent method. However, gradient ascent methods can oscillate or diverge for some "pathological" non-smooth non-convex functions [13, 60]. The formalism presented in this paper establishes convergence guarantees of the proposed method, which are, however, more subtle than for smooth functions. For example, Theorem 14 guarantees that limits of subsequences of θ_k are Clarke-stationary, and a similar proof can be made to show that the whole sequence θ_k accumulates around the set of Clarke-stationary points, but, in general, does not guarantee that the whole sequence θ_k converges to a single point (see e.g. the example in ref. [60] in which the trajectory oscillates around a circle of stationary points). The existence of stronger convergence guarantees for the proposed model, and sufficient conditions for the existence of stationary points and boundedness of θ_k, remains an open question. In particular, the lack of complete separability is a necessary condition for the existence of finite global maximizers and boundedness of θ_k, but does not preclude the existence of local maxima or other Clarke-stationary points.

Theorem 14 requires that all alignments be recalculated at each step, making it computationally intensive for large datasets. This can be mitigated by taking a random sample of sequence pairs at each iteration, similar to a batch learning strategy in neural networks:

▶ **Theorem 15.** *For $k \geq 1$ and $N \in \mathbb{N}_+$, let $\mathcal{D}_k$ be a sequence of independent N-element random subsets of $\mathcal{D}$ such that the elements of each subset are selected uniformly with replacement. Assume the same conditions on η_k and ξ_k as in Theorem 14, but let the sequence θ_k now be defined as*

$$\theta_{k+1} = \theta_k + \eta_k g_k + \xi_k \text{ for } g_k \in \partial_\theta^C l(\alpha, \theta_k; \mathcal{D}_k) \tag{11}$$

Assume that subgradients g_k are obtained using Proposition (12) applied to $l(\alpha, \theta_k; \mathcal{D}_k)$. Then, with probability 1, every limit point of θ_k is a Clarke-stationary point of $l(\alpha, \theta; \mathcal{D})$.

The proof is presented in the Appendix. Theorems 14 and 15 provide ways of estimating θ under a known α. In order to optimize the latter, we note that the log-likelihood is differentiable and concave (due to a negative second derivative) with respect to α for any fixed θ. It follows that for any fixed θ the optimal value of α is unique and can be efficiently found with standard optimization methods such as the Newton-Rhapson procedure. We use this observation to replace the free parameter α with $\alpha^*(\theta) = \arg\max_\alpha l(\alpha, \theta; \mathcal{D})$ and optimize $l(\alpha^*(\theta), \theta; \mathcal{D})$ using the subgradient method for θ. This procedure is summarized in Algorithm 1.

4 Results

While theoretical guarantees of convergence require a decreasing sequence of steps, in practice we generally see a better performance with a constant step size $\eta_k \equiv \eta$ due to a faster convergence, and we use this sequence in the Results section with $\sigma_k = \sigma/k$.

Algorithm 1 Discriminative learning of alignment parameters from pairs of labeled sequences.

Input: List of sequence pairs with binary labels $\mathcal{D} = \{(A_i, B_i, y_i) \colon i = 1, 2, \ldots, |\mathcal{D}|\}$;
Hyperparameters: Maximum number of iterations $I_{\max}$; Sequence of step lengths η_k; Sequence of standard deviations σ_k; Initial alignment parameters
Output: Alignment parameters θ^*, logistic intercept α^*;
Let $\mathcal{A}_0 \leftarrow$ alignments of (A_i, B_i) under initial parameters
Let $\alpha_0, \theta_0 \leftarrow$ Initial estimate using Equation (6)
for $k = 1, 2, \ldots, I_{\max}$ **do**
  Let $\mathcal{A}_k \leftarrow$ Alignments of $\mathcal{D}$ under θ_{k-1}
  Let $x^* \leftarrow$ configurations of $\mathcal{A}_k$
  Let $g_k \leftarrow \sum_i (y_i - p(\theta_{k-1}; \alpha_{k-1}))x_i^*$
  Let $\xi_k \sim \mathcal{N}(0, \sigma_k I)$
  Let $\theta_k \leftarrow \theta_{k-1} + \eta_k g_k + \xi_k$
  Let $\alpha_k \leftarrow \max_\alpha l(\theta_k, \alpha)$ ▷ Newton-Rhapson procedure
end for
Return: $\theta^* = \theta_k, \alpha^* = \alpha_k$

4.1 Convergence and accuracy of estimation

To experimentally validate the proposed method, we performed simulation experiments with fixed substitution matrices and gap penalties using a data set of microRNA (miRNA) sequences. MiRNAs are short RNA molecules (typically 18-25 nt) that regulate gene

expression by recognizing mRNA transcripts, or other RNA molecules, through partial complementary pairing and inhibiting translation and/or promoting degradation [47, 43, 30]. Recently, a set of benchmark datasets called miRBench was released, which contains several datasets of pairs of miRNA sequences and 50nt-long RNA sequences labeled based on whether the RNA sequence contains a target site for the miRNA [61, 26]. We used the train split of the Hejret dataset from miRBench, which contains 4084 positive and 4109 negative instances.

In order to obtain a dataset with a known ground truth, we replaced the original binary labels with simulated ones generated as follows. We investigated two local alignment scoring schemes, referred to as "simple model" and "full model". The "simple" model had match weight 1, mismatch weight -1, gap open -1.2 and gap extend -0.1 The more challenging "full" model had an asymmetric DNA substitution matrix M with manually selected entries to obtain a diversity of scores:

$$M = \left(\begin{array}{c|cccc} & A & C & U & G \\ \hline A & 1.0 & -0.3 & -1.0 & -0.8 \\ C & -0.6 & 1.2 & -0.3 & -1.0 \\ U & -1.2 & -0.4 & 1.0 & -0.8 \\ G & -0.4 & -1.4 & -0.9 & 1.3 \end{array}\right) \qquad G = \left(\begin{array}{r|c} \text{Gap open} & -1.2 \\ \text{Gap extend} & -0.1 \end{array}\right)$$

To generate labels, we calculated the scores of pairwise local alignments of miRNA sequences and RNA fragments using Biopython [11]. We used the scores to calculate the probabilities of interactions using Equation (1), with α equal to the negative of the median score to obtain a roughly balanced dataset. Finally, for each pair, we used the corresponding probability to simulate a binary label. We run DiscrimAlign on the simulated dataset for 200 iterations with a stochastic factor $\sigma = 0.001$ and different learning rates η ranging from 5e-06 to 5e-05 (using a constant step size in order to obtain a faster convergence, as discussed in Methods).

The results are shown in Figure 1. Step sizes which resulted in an initial drop in the log-likelihood often corresponded to a faster convergence. Excessive step sizes resulted in an unstable behavior, while insufficient step sizes result in a very slow convergence. The step sizes $\eta = 2 \cdot 10^{-5}$ for the simple model and $\eta = 4 \cdot 10^{-5}$ for the full model were among the smallest step sizes that resulted in a sufficiently fast convergence, thus offering a balance between speed and numerical stability. Fig. 2 shows the estimated weights of the asymmetric matrix for $\eta = 4 \cdot 10^{-5}$. The correlation between the entries of the true substitution matrix M and the estimated one $\hat{M}$ was equal to over 0.99, and the mean absolute error of estimation was 0.06. To analyze the robustness of the method, we run DiscrimAlign with $\eta = 4 \cdot 10^{-5}$ for 20 replicated datasets with the asymmetric matrix. The results show that the speed of convergence varies, but the true value of the log-likelihood was reached (or nearly reached) after 200 iterations in all replicates (Fig. 1). The mean absolute error varied between 0.06 and 0.12.

4.2 Dataset requirements for amino acid matrices

A natural question is the size of dataset necessary for an accurate estimation depending on the size of the model and the alphabets. For example, a full model for amino acid sequences would require 403 parameters (400 substitution weights, 2 gap penalties, and the intercept α). The error of estimation can be decomposed into error resulting from the training procedure (which includes learning the alignments) and the inherent statistical error resulting from the finite size of the dataset and randomness of labels. In this section, we show how to estimate the latter error in simulated experiments using a standard logistic regression model applied to ground-truth alignments (Equation (6)). This provides a lower bound on the error of

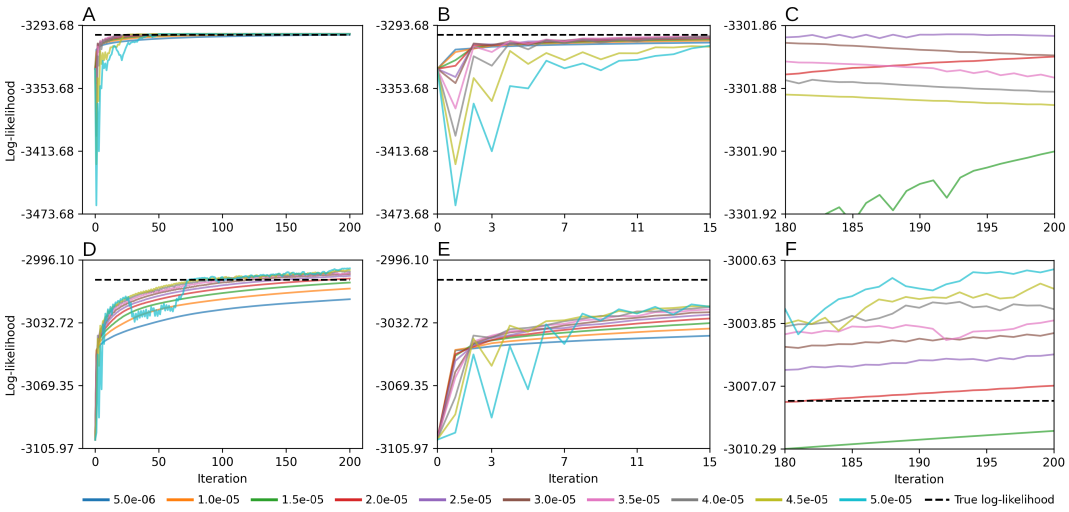

Figure 1 Estimation of discriminative substitution matrices with DiscrimAlign for pairs of RNA sequences with simulated labels. Plots show trajectories of the log-likelihood for different step sizes η for a constant sequence of step sizes. A,B,C: Simple substitution model with match/mismatch weights and an affine gap penalty (4 parameters plus intercept). D,E,F: Asymmetric substitution matrix and an affine gap penalty (18 parameters plus intercept). A,D: Full trajectories; B,E: Initial iterations; C,F: Final iterations. All trajectories shown in panel C are above the true log-likelihood; the remaining trajectories did not reach it or were unstable. The light blue line in D-F shows the method's behavior for excessive step sizes (numerical instability), while the dark blue line shows insufficient step sizes (slow convergence).

DiscrimAlign without the need for a full training procedure, while also allowing one to use an extensive body of statistical results including confidence intervals. In turn, a lower bound on the error provides a lower bound on dataset requirements.

To generate simulated ground-truth labels to study the lower bound on estimation error for protein alignments, we used a centered and scaled BLOSUM62 matrix with gap penalties $G = (-1, -0.1)$. Scaling the matrix was done so that individual substitutions do not have an overwhelming impact on the predicted probability, as large values of θ mean that any differences in alignment configurations result in large changes of the predicted probability. To generate sequence pairs, we downloaded proteomes of *Homo sapiens* (`GCF_000001405.40`, 136 807 sequences) and *Gallus gallus* (`GCF_016699485.2`, 68 683 sequences) from the NCBI Genome database. We selected proteins up to 1 000 amino acids in length to avoid excessive computational times, used a local installation of BLASTP to identify homologs with an E-value threshold of 1e-6 (resulting in 2 238 760 pairs of proteins), and created three datasets by randomly selecting pairs with $|\mathcal{D}| = 10^4, 10^5$ or 10^6. We remark that these sequence pairs have highly varying degrees of similarity, because an E-value threshold of 1e-6 is relatively high when comparing two proteomes rather than a proteome against a large database (since E-values are proportional to the number of compared sequences). We calculated global alignments and discarded those which consisted of more than 20% of gaps. We used the scores of remaining alignments to generate labels as in the previous section. Finally, we arranged the configurations of the alignments and the labels into arrays and estimated the parameters using the `LogisticRegression` function implemented in the scikit-learn library (`penalty='None', solver='newton-cg'`) [56].

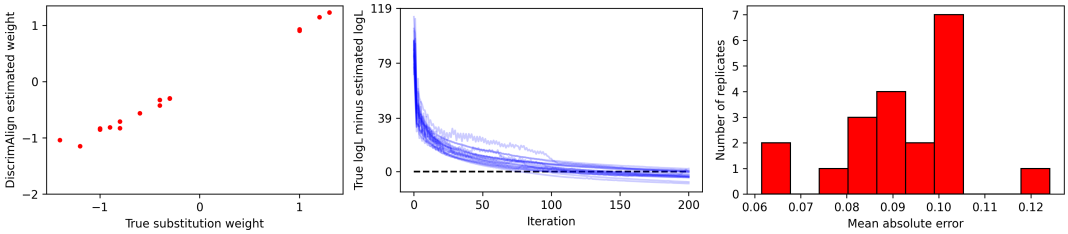

Figure 2 Accuracy of estimation in replicated experiments on RNA sequences with simulated labels. Left: Estimated vs ground-truth substitution weights in an example run. Middle: Trajectories of the difference between current log-likelihood and the ground-truth log-likelihood for 20 replicated experiments. Right: A histogram of mean absolute errors of estimation for 20 replicated experiments. All plots correspond to a model with an asymmetric substitution matrix and an affine gap penalty.

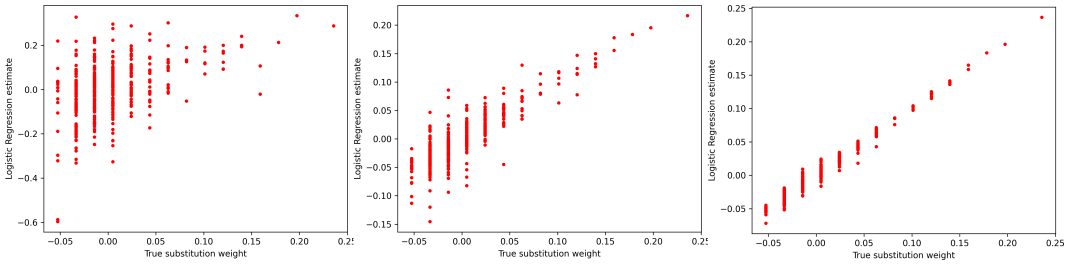

Figure 3 A lower bound on the estimation error for asymmetric amino acid substitution matrices for different dataset sizes. Logistic regression estimation of amino acid substitution matrices directly from optimal alignments obtained with the true parameters (without DiscrimAlign optimization). From left to right, $|\mathcal{D}| = 10^4, 10^5, 10^6$.

The large dataset $|\mathcal{D}| = 10^6$ resulted in a clear correlation between the estimated and ground-truth substitution weights (Fig 3), providing a likely indication of the minimal requirements for an accurate estimation of discriminative substitution matrices for detecting homology between *Homo sapiens* and *Gallus gallus* proteins. For $|\mathcal{D}| = 10^4$, the estimated parameters were only loosely correlated with the ground-truth, which is to be expected considering the large number of parameters. This correlation depended not just on $|\mathcal{D}|$, but also on the scaling of the substitution matrix; in particular, small parameter values were more difficult to estimate because of a lower impact on the probability (Fig. 3). A realistic range of substitution weights for amino acid discriminative substitution matrices requires further studies, and most likely depends on the similarity of compared proteins. The construction of proper datasets for estimation of discriminative matrices for homology search also requires further studies, as selecting positive and negative instances simply as related and unrelated proteins results in a complete separation. This can be overcome either by a specific selection of the dataset or developing regularized models.

4.3 Case study: prediction of miRNA-target interactions

The task of predicting miRNA target interactions is typically addressed with black-box methods such as convolutional neural networks [61], although approaches based on alignments with position-specific substitution matrices have been explored with promising results [67]. In this case study, we investigated whether a pairwise alignment score with a discriminative substitution matrix can be used to identify interacting pairs. We used the $\mathcal{D}_{\text{Hejret}}$ dataset

Table 1 The area under the precision-recall curve (AUPRC) and final log-likelihood of miRNA interaction prediction for different alignment models trained with `DiscrimAlign`.

Training dataset	Hejret Train				Manakov Train			
Test dataset	Hejret Test	Manakov Test	Manakov Leave-Out	LogLik	Hejret Test	Manakov Test	Manakov Leave-Out	LogLik
Simple Linear	0.852	0.746	0.741	-3321.7	0.851	0.752	0.747	-1219390.0
Simple Affine	0.848	0.761	**0.757**	-3304.9	0.847	0.764	0.761	-1202172.6
Symmetric Linear	0.852	0.754	0.751	-3326.7	**0.853**	0.760	0.759	-1208419.1
Symmetric Affine	0.849	**0.761**	0.756	-3280.3	0.849	0.766	0.763	-1199209.3
Asymmetric Linear	**0.854**	0.753	0.750	-3302.6	0.852	0.761	0.759	-1208102.0
Asymmetric Affine	0.852	0.760	0.756	-3256.0	0.849	**0.767**	**0.764**	-1197558.5

Table 2 The Area under Precision-Recall curves for different miRNA-target prediction algorithms. Data for TargetScanCNN, miRBind and miRNA_CNN_Hejret2023 comes from ref. [61]. Full model refers to an asymmetric substitution matrix and affine gap penalty.

Test dataset	Hejret Test	Manakov Test	Manakov Leave-Out
`TargetScanCNN`	0.71	0.77	0.76
`miRBind`	0.80	0.71	0.71
`miRNA_CNN_Hejret2023`	0.77	0.71	0.71
`miRNA_CNN_Hejret2023` retrained on Hejret Train	0.86	0.79	0.78
`miRNA_CNN_Hejret2023` retrained on Manakov Train	0.84	0.84	0.81
`DiscrimAlign` full model trained on Hejret Train	0.85	0.76	0.76
`DiscrimAlign` full model trained on Manakov Train	0.85	0.77	0.76

and another, larger dataset from the miRBench package [61], referred to as Manakov, with 1 253 320 positive instances in the training split. This data set contains three splits: train, which we used to train our model; test, which we used to evaluate the model; and an additional left-out split which contains miRNAs not present in either the train or test split. We run DiscrimAlign on pairs of miRNA sequences and reverse-complemented target sequences with labels indicating experimentally confirmed interactions. We investigated all combinations of linear and affine gap penalties and simple, symmetric and asymmetric substitution matrices. We evaluated the performance using the area under the precision-recall curve (AUPRC) on the test split of each dataset in the same way as reported in the miRBench article [61].

For each training dataset, the training split was divided into fitting and validation subsets. The fitting subset was used to estimate alignment parameters for each candidate model, while the validation subset was used to select the best configuration. We considered local and global alignment, linear and affine gap penalties, and three substitution models: simple match/mismatch scores, symmetric substitution matrices, and general asymmetric substitution matrices. Performance was measured using the area under the precision-recall curve (AUPRC), while the final log-likelihood was used to assess model fit.

Asymmetric substitution matrices provided the best AUPRC when evaluated on the held-out test splits of their respective datasets, but not in cross-dataset evaluation, indicating some degree of overfitting to the experimental technique used in each dataset (Table 1). Models with a symmetric substitution matrix seemed to generalize better between datasets while having only slightly lower performance within datasets. Overall, models trained on the larger $\mathcal{D}_{\text{Manakov}}$ generalized well to the Hejret test set, achieving AUPRC values close to those of the Hejret-trained models, which was not the case for models trained on $\mathcal{D}_{\text{Hejret}}$ when evaluated on $\mathcal{D}_{\text{Manakov}}$. This suggests that the larger Manakov training set may capture broader miRNA-target pairing patterns.

Next, we refitted the full model (19 parameters) on full training splits and evaluated on the held-out test datasets (step size $\eta = 5 \times 10^{-4}$, 500 iterations). These models achieved AUPRC values comparable to neural-network baselines (Table 2). The substitution matrix obtained from the Hejret dataset was considerably asymmetric, likely reflecting the directional alignment of miRNA sequences to reverse-complemented target sequences.

$$M_{\text{Hejret}} = \left(\begin{array}{c|cccc} & A & C & T & G \\ \hline A & 0.66 & -0.60 & -0.45 & -0.60 \\ C & -0.72 & 0.65 & -0.77 & -0.86 \\ T & -0.57 & -0.36 & 0.62 & -0.58 \\ G & -0.36 & -0.78 & -0.58 & 0.63 \end{array}\right) \quad G_{\text{Hejret}} = \left(\begin{array}{c|c} \text{Gap open} & -1.12 \\ \text{Gap extend} & -0.11 \end{array}\right).$$

5 Discussion

Some of the first substitution matrices, both for amino acids and nucleotides, were derived from first principles, such as the physical properties of amino acids or the properties of the genetic code [55, 22, 36]. When databases of example alignments became available, researchers started to derive matrices that describe observed patterns of substitutions. In this work, we propose a new type of substitution matrices that we call *discriminative substitution matrices*, which describe the differences observed between two sets of pairs of sequences.

We described a general theoretical framework of learning discriminative matrices through a logistic link between the alignment score and the label associated with each pair. We analyzed the mathematical properties of alignment score and the logistic link as functions of substitution and gap weights. We derived subgradients of the alignment score with respect to the parameters and subgradients of the logistic link function and the log-likelihood. We used these results to implement a learning procedure and validated it on simulated data sets to show convergence and accuracy of estimation. Finally, we applied the method to learn a discriminative matrix for the task of predicting microRNA-target interactions. The discriminative alignment performed similarly to black-box classifiers based on neural networks while offering more biological insight into the interactions thanks to the interpretable nature of the model, which can be used in a downstream analysis to identify binding patterns by investigating the optimal alignments.

We remark that most amino acid substitution matrices are relatively similar, because certain pairs of amino acids align more commonly than others regardless of the data set (e.g. serine and threonine is more likely to align than serine and tryptophan, driving the differences in substitution weights). While discriminative matrices are thus not expected to be highly different than e.g. BLOSUM62 in terms of absolute differences of their values, even relatively minor changes can impact alignments and the preformance of database searches. A seemingly understudied problem is a quantitative analysis of sensitivity of alignments to parameter values in practical applications with commonly used substitution weights, which could answer how accurate the estimation of parameters needs to be.

The proposed model offers many topics for future investigations, some of which were mentioned throughout the paper. Other topics include improvements in computational efficiency and derivation of statistical tests for parameters, such as a likelihood ratio test to determine whether an affine gap penalty gives a statistically significant improvement in classification compared to a linear one in a particular application. Finally, while the proposed method is not explicitly designed to optimize the accuracy of alignments, it would be interesting to see how it performs compared to methods such as IPA or POP.

References

1 Adelin Albert and John A Anderson. On the existence of maximum likelihood estimates in logistic regression models. *Biometrika*, 71(1):1–10, 1984.

2 Stephen F Altschul. Amino acid substitution matrices from an information theoretic perspective. *Journal of molecular biology*, 219(3):555–565, 1991.

3 Olivier Bastien, Sylvaine Roy, and Éric Maréchal. Construction of non-symmetric substitution matrices derived from proteomes with biased amino acid distributions. *Comptes rendus. Biologies*, 328(5):445–453, 2005.

4 Amir Beck. *First-order methods in optimization*. SIAM, 2017.

5 Kevin Brick and Elisabetta Pizzi. A novel series of compositionally biased substitution matrices for comparing plasmodium proteins. *BMC bioinformatics*, 9(1):236, 2008. doi:10.1186/1471-2105-9-236.

6 Celeste J Brown, Audra K Johnson, and Gary W Daughdrill. Comparing models of evolution for ordered and disordered proteins. *Molecular biology and evolution*, 27(3):609–621, 2010.

7 Christiam Camacho, George Coulouris, Vahram Avagyan, Ning Ma, Jason Papadopoulos, Kevin Bealer, and Thomas L Madden. Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009. doi:10.1186/1471-2105-10-421.

8 Michał Aleksander Ciach, Elissavet Zacharopoulou, Michał Piotr Startek, Błażej Miasojedow, and Panagiotis Alexiou. DiscrimAlign. Software, version 1.0., swhId: swh:1:dir:e70ee75187e06eb93adc1da6022502a80230d4b0 (visited on 2026-08-12). URL: https://github.com/BioGeMT/DiscrimAlign, doi:10.4230/artifacts.27630.

9 Frank H Clarke. Generalized gradients and applications. *Transactions of the American Mathematical Society*, 205:247–262, 1975.

10 Frank H Clarke, Yu S Ledyaev, Ronald J Stern, and RR Wolenski. *Nonsmooth analysis and control theory*. Springer, 1998.

11 Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422, 2009.

12 Cuong Cao Dang, Vincent Lefort, Vinh Sy Le, Quang Si Le, and Olivier Gascuel. Replacementmatrix: a web server for maximum-likelihood estimation of amino acid replacement rate matrices. *Bioinformatics*, 27(19):2758–2760, 2011. doi:10.1093/BIOINFORMATICS/BTR435.

13 Aris Daniilidis and Dmitriy Drusvyatskiy. Pathological subgradient dynamics. *SIAM Journal on Optimization*, 30(2):1327–1338, 2020. doi:10.1137/19M1298147.

14 Damek Davis, Dmitriy Drusvyatskiy, Sham Kakade, and Jason D Lee. Stochastic subgradient method converges on tame functions. *Foundations of computational mathematics*, 20(1):119–154, 2020. doi:10.1007/S10208-018-09409-5.

15 Margaret Dayhoff, R Schwartz, and B Orcutt. A model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5:345–352, 1978.

16 Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998. doi:10.1017/CBO9780511790492.

17 Robert C Edgar. Optimizing substitution matrix choice and gap parameters for sequence alignment. *Bmc Bioinformatics*, 10(1):396, 2009. doi:10.1186/1471-2105-10-396.

18 Robert C Edgar. Muscle5: High-accuracy alignment ensembles enable unbiased assessments of sequence homology and phylogeny. *Nature communications*, 13(1):6968, 2022.

19 Martin C Frith. How sequence alignment scores correspond to probability models. *Bioinformatics*, 36(2):408–415, 2020. doi:10.1093/BIOINFORMATICS/BTZ576.

20 Martin C Frith, Michiaki Hamada, and Paul Horton. Parameters for accurate genome alignment. *BMC bioinformatics*, 11(1):80, 2010. doi:10.1186/1471-2105-11-80.

21 Renganayaki Govindarajan, Biji Christopher Leela, and Achuthsankar S Nair. Rblosum performs better than corblosum with lesser error per query. *BMC Research Notes*, 11(1):328, 2018.

22 Richard Grantham. Amino acid difference formula to help explain protein evolution. *science*, 185(4154):862–864, 1974.

23 Jerrold R Griggs, Phil Hanlon, Andrew M Odlyzko, and Michael S Waterman. On the number of alignments of k sequences. *Graphs and Combinatorics*, 6(2):133–146, 1990. doi:10.1007/BF01787724.

24 Dan Gusfield and Paul Stelling. Parametric and inverse-parametric sequence alignment with xparal. In *Methods in enzymology*, volume 266, pages 481–494. Elsevier, 1996.

25 Michiaki Hamada, Yukiteru Ono, Kiyoshi Asai, and Martin C Frith. Training alignment parameters for arbitrary sequencers with last-train. *Bioinformatics*, 33(6):926–928, 2017. doi:10.1093/BIOINFORMATICS/BTW742.

26 Vaclav Hejret, Nandan Mysore Varadarajan, Eva Klimentova, Katarina Gresova, Ilektra-Chara Giassa, Stepanka Vanacova, and Panagiotis Alexiou. Analysis of chimeric reads characterises the diverse targetome of ago2-mediated regulation. *Scientific Reports*, 13(1):22895, 2023.

27 Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the national academy of sciences*, 89(22):10915–10919, 1992.

28 Martin Hess, Frank Keul, Michael Goesele, and Kay Hamacher. Addressing inaccuracies in blosum computation improves homology search performance. *Bmc Bioinformatics*, 17(1):189, 2016. doi:10.1186/S12859-016-1060-3.

29 Reiner Horst and Nguyen V Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

30 Hyeonseo Hwang, Hee Ryung Chang, and Daehyun Baek. Determinants of functional microrna targeting. *Molecules and cells*, 46(1):21–32, 2023.

31 Sumaiya Islam and Robert J Pantazes. Developing similarity matrices for antibody-protein binding interactions. *Plos one*, 18(10):e0293606, 2023.

32 Kejue Jia and Robert L Jernigan. New amino acid substitution matrix brings sequence alignments into agreement with structure matches. *Proteins: Structure, Function, and Bioinformatics*, 89(6):671–682, 2021.

33 Thorsten Joachims, Tamara Galor, and Ron Elber. Learning to align sequences: A maximum-margin approach. In *New algorithms for macromolecular simulation*, pages 57–69. Springer, 2005.

34 Kaisa Joki, Adil M Bagirov, Napsu Karmitsa, Marko M Makela, and Sona Taheri. Double bundle method for finding clarke stationary points in nonsmooth dc programming. *SIAM Journal on Optimization*, 28(2):1892–1919, 2018. doi:10.1137/16M1115733.

35 David T Jones, William R Taylor, and Janet M Thornton. The rapid generation of mutation data matrices from protein sequences. *Bioinformatics*, 8(3):275–282, 1992. doi:10.1093/BIOINFORMATICS/8.3.275.

36 Samuel Karlin and Stephen F Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proceedings of the National Academy of Sciences*, 87(6):2264–2268, 1990.

37 Kazutaka Katoh, John Rozewicki, and Kazunori D Yamada. Mafft online service: multiple sequence alignment, interactive sequence choice and visualization. *Briefings in bioinformatics*, 20(4):1160–1166, 2019. doi:10.1093/BIB/BBX108.

38 John Kececioglu and Eagu Kim. Simple and fast inverse alignment. In *Annual International Conference on Research in Computational Molecular Biology*, pages 441–455. Springer, 2006. doi:10.1007/11732990_37.

39 Frank Keul, Martin Hess, Michael Goesele, and Kay Hamacher. Pfasum: a substitution matrix from pfam structural alignments. *BMC bioinformatics*, 18(1):293, 2017. doi:10.1186/S12859-017-1703-Z.

40 Eagu Kim and John Kececioglu. Learning scoring schemes for sequence alignment from partial examples. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(4):546–556, 2008. doi:10.1109/TCBB.2008.57.

41 Satoshi Koide, Keisuke Kawano, and Takuro Kutsuna. Neural edit operations for biological sequences. *Advances in Neural Information Processing Systems*, 31, 2018.

42 Carolin Kosiol and Nick Goldman. Different versions of the dayhoff rate matrix. *Molecular biology and evolution*, 22(2):193–199, 2005.

43 Jacek Krol, Inga Loedige, and Witold Filipowicz. The widespread regulation of microrna biogenesis, function and decay. *Nature reviews genetics*, 11(9):597–610, 2010.

44 Igor B Kuznetsov. Protein sequence alignment with family-specific amino acid similarity matrices. *BMC Research Notes*, 4(1):296, 2011.

45 Si Quang Le and Olivier Gascuel. An improved general amino acid replacement matrix. *Molecular biology and evolution*, 25(7):1307–1320, 2008.

46 Claire Lemaitre, Aurélien Barré, Christine Citti, Florence Tardy, François Thiaucourt, Pascal Sirand-Pugnet, and Patricia Thébault. A novel substitution matrix fitted to the compositional bias in mollicutes improves the prediction of homologous relationships. *BMC bioinformatics*, 12(1):457, 2011. doi:10.1186/1471-2105-12-457.

47 Sean E McGeary, Kathy S Lin, Charlie Y Shi, Thy M Pham, Namita Bisaria, Gina M Kelley, and David P Bartel. The biochemical basis of microrna targeting efficacy. *Science*, 366(6472):eaav1741, 2019.

48 Uros Midic, A Keith Dunker, and Zoran Obradovic. Protein sequence alignment and structural disorder: a substitution matrix for an extended alphabet. In *Proceedings of the KDD-09 Workshop on Statistical and Relational Learning in Bioinformatics*, pages 27–31, 2009. doi:10.1145/1562090.1562096.

49 James T Morton, Charlie EM Strauss, Robert Blackwell, Daniel Berenberg, Vladimir Gligorijevic, and Richard Bonneau. Protein structural alignments from sequence. *BioRxiv*, pages 2020–11, 2020.

50 Tobias Müller, Rainer Spang, and Martin Vingron. Estimating amino acid substitution models: a comparison of dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Molecular biology and evolution*, 19(1):8–13, 2002.

51 Tobias Müller and Martin Vingron. Modeling amino acid replacement. *Journal of Computational Biology*, 7(6):761–776, 2000. doi:10.1089/10665270050514918.

52 VN Novoseletsky, GA Armeev, and KV Shaitan. Construction and analysis of amino acid substitution matrices for optimal alignment of microbial rhodopsin sequences. *Moscow University Biological Sciences Bulletin*, 74(1):21–25, 2019.

53 Evgenii Ofitserov, Vasily Tsvetkov, and Vadim Nazarov. Soft edit distance for differentiable comparison of symbolic sequences. *arXiv preprint arXiv:1904.12562*, 2019. arXiv:1904.12562.

54 Lior Pachter and Bernd Sturmfels. *Algebraic statistics for computational biology*, volume 13. Cambridge university press, 2005.

55 William R Pearson. Selecting the right similarity-scoring matrix. *Current protocols in bioinformatics*, 43(1):3–5, 2013.

56 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. doi:10.5555/1953048.2078195.

57 Samantha Petti, Nicholas Bhattacharya, Roshan Rao, Justas Dauparas, Neil Thomas, Juannan Zhou, Alexander M Rush, Peter Koo, and Sergey Ovchinnikov. End-to-end learning of multiple sequence alignments with differentiable smith–waterman. *Bioinformatics*, 39(1):btac724, 2023. doi:10.1093/BIOINFORMATICS/BTAC724.

58 Valery Polyanovsky, Alexander Lifanov, Natalia Esipova, and Vladimir Tumanyan. The ranging of amino acids substitution matrices of various types in accordance with the alignment accuracy criterion. *BMC bioinformatics*, 21(Suppl 11):294, 2020. doi:10.1186/S12859-020-03616-0.

59 Predrag Radivojac, Zoran Obradovic, Celeste J Brown, and A Keith Dunker. Improving sequence alignments for intrinsically disordered proteins. In *Biocomputing 2002*, pages 589–600. World Scientific, 2001.

60 Rodolfo Rios-Zertuche. Examples of pathological dynamics of the subgradient method for lipschitz path-differentiable functions. *Mathematics of Operations Research*, 47(4):3184–3206, 2022. doi:10.1287/MOOR.2021.1241.

61 Stephanie Sammut, Katarina Gresova, Dimosthenis Tzimotoudis, Eva Marsalkova, David Cechak, and Panagiotis Alexiou. mirbench: novel benchmark datasets for microrna binding site prediction that mitigate against prevalent microrna frequency class bias. *Bioinformatics*, 41(Supplement_1):i542–i551, 2025. doi:10.1093/BIOINFORMATICS/BTAF233.

62 Eric W Sayers, Jeffrey Beck, Evan E Bolton, J Rodney Brister, Jessica Chan, Ryan Connor, Michael Feldgarden, Anna M Fine, Kathryn Funk, Jinna Hoffman, et al. Database resources of the national center for biotechnology information in 2025. *Nucleic acids research*, 53(D1):D20–D29, 2025.

63 David J States, Warren Gish, and Stephen F Altschul. Improved sensitivity of nucleic acid database searches using application-specific scoring matrices. *Methods*, 3(1):66–70, 1991.

64 Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028, 2017.

65 Mark P Styczynski, Kyle L Jensen, Isidore Rigoutsos, and Gregory Stephanopoulos. Blosum62 miscalculations improve search performance. *Nature biotechnology*, 26(3):274–275, 2008.

66 Fangting Sun, David Fernández-Baca, and Wei Yu. Inverse parametric sequence alignment. *Journal of Algorithms*, 53(1):36–54, 2004. doi:10.1016/J.JALGOR.2004.04.008.

67 Amlan Talukder, Xiaoman Li, and Haiyan Hu. Position-wise binding preference is important for mirna target site prediction. *Bioinformatics*, 36(12):3680–3686, 2020. doi:10.1093/BIOINFORMATICS/BTAA195.

68 Octavian Teodorescu, Tamara Galor, Jaroslaw Pillardy, and Ron Elber. Enriching the sequence substitution matrix by structural information. *Proteins: Structure, Function, and Bioinformatics*, 54(1):41–48, 2004.

69 Rakesh Trivedi and Hampapathalu Adimurthy Nagarajaram. Substitution scoring matrices for proteins-an overview. *Protein Science*, 29(11):2150–2163, 2020.

70 Simon Whelan and Nick Goldman. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular biology and evolution*, 18(5):691–699, 2001.

71 Alex J Wilkie. Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function. *Journal of the American Mathematical Society*, 9(4):1051–1094, 1996.

72 Kazunori Yamada and Kentaro Tomii. Revisiting amino acid substitution matrices for identifying distantly related proteins. *Bioinformatics*, 30(3):317–325, 2014. doi:10.1093/BIOINFORMATICS/BTT694.

73 Yin Yao and Martin C Frith. Improved dna-versus-protein homology search for protein fossils. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 20(3):1691–1699, 2022. doi:10.1109/TCBB.2022.3177855.

A Proofs

Proof of Proposition 1. Using Equation (2), we can express $U_{A,B}(x)$ as

$$U_{A,B}(x^*) = \{\theta \in \Theta \colon \forall_{x \in \mathcal{X}_{A,B}} \langle \theta, x^* \rangle \geq \langle \theta, x \rangle\}.$$

From the linearity of the inner product, $\langle \theta, x^* \rangle \geq \langle \theta, x \rangle$ is equivalent to $\langle \theta, x^* - x \rangle \geq 0$. This allows us to write

$$U_{A,B}(x^*) = \{\theta \in \Theta \colon \forall_{x \in \mathcal{X}_{A,B}} \langle \theta, x^* - x \rangle \geq 0\} = \bigcap_{x \in \mathcal{X}_{A,B}} \{\theta \in \Theta \colon \langle \theta, x^* - x \rangle \geq 0\}.$$

For any vector y, the set defined by the inequality $\langle \theta, y \rangle \geq 0$ is a closed half-space, which is a convex set. Therefore, $U_{A,B}(x)$ is a closed convex set as an intersection of a finite number of such sets. Furthermore, since for any scalar s we have $\langle s\theta, y \rangle = s\langle \theta, y \rangle$, it follows that for $s > 0$ we have $\langle s\theta, y \rangle \geq 0$ if and only if $\langle \theta, y \rangle \geq 0$, which shows that $U_{A,B}(x)$ is a cone. The zero vector of parameters always belongs to $U_{A,B}(x)$, because $\langle 0, y \rangle = 0$ for any vector y. The proof for $U^*_{A,B}(x)$ is similar; however, this set is open as an intersection of a finite number of open half-spaces. Furthermore, the zero vector never belongs to $U^*_{A,B}(x)$, and the set may be empty. ◀

Proof of Lemma 2. For the proof of Lipschitz continuity, consider two vectors of parameters θ and θ'. First, we note that

$$\max_{x \in \mathcal{X}_{A,B}} \langle \theta, x \rangle - \max_{x \in \mathcal{X}_{A,B}} \langle \theta', x \rangle \leq \max_{x \in \mathcal{X}_{A,B}} (\langle \theta, x \rangle - \langle \theta', x \rangle) = \max_{x \in \mathcal{X}_{A,B}} \langle \theta - \theta', x \rangle.$$

The inequality follows from the fact that

$$\begin{aligned} \max_{x \in \mathcal{X}_{A,B}} \langle \theta, x \rangle &= \max_{x \in \mathcal{X}_{A,B}} \langle (\theta - \theta') + \theta', x \rangle = \\ &= \max_{x \in \mathcal{X}_{A,B}} (\langle \theta - \theta', x \rangle + \langle \theta', x \rangle) \leq \max_{x \in \mathcal{X}_{A,B}} \langle \theta - \theta', x \rangle + \max_{x \in \mathcal{X}_{A,B}} \langle \theta', x \rangle. \end{aligned}$$

It follows that $S_\theta(A,B) - S_{\theta'}(A,B) \leq S_{\theta-\theta'}(A,B)$. Next, using the Cauchy-Schwarz inequality $\langle a, b \rangle \leq ||a|| \cdot ||b||$, we have

$$S_{\theta-\theta'}(A,B) = \max_{x \in \mathcal{X}_{A,B}} \langle \theta - \theta', x \rangle \leq \max_{x \in \mathcal{X}_{A,B}} (||\theta - \theta'|| \cdot ||x||) = ||\theta - \theta'|| \max_{x \in \mathcal{X}_{A,B}} ||x||.$$

We therefore have $S_\theta(A,B) - S_{\theta'}(A,B) \leq ||\theta - \theta'|| \max_{x \in \mathcal{X}_{A,B}} ||x||$. Swapping θ and θ' (and noting that $||\theta - \theta'|| = ||\theta' = \theta||$), we obtain $S_{\theta'}(A,B) - S_\theta(A,B) \leq ||\theta - \theta'|| \max_{x \in \mathcal{X}_{A,B}} ||x||$, and thus we conclude that

$$|S_\theta(A,B) - S_{\theta'}(A,B)| \leq ||\theta - \theta'|| \max_{x \in \mathcal{X}_{A,B}} ||x||.$$

For the proof of convexity, consider $p \in [0,1]$ and two vectors of parameters θ and θ'. Then,

$$\begin{aligned} S_{p\theta+(1-p)\theta'}(A,B) &= \max_{x \in X_{A,B}} \langle p\theta + (1-p)\theta', x \rangle = \\ &= \max_{x \in X_{A,B}} (p\langle \theta, x \rangle + (1-p)\langle \theta', x \rangle) \leq \\ &\leq p \max_{x \in X_{A,B}} \langle \theta, x \rangle + (1-p) \max_{x \in X_{A,B}} \langle \theta', x \rangle = \\ &= pS_\theta(A,B) + (1-p)S_{\theta'}(A,B) \end{aligned}$$ ◀

Proof of Proposition 5. Since the functions $\theta \to S_\theta(A_i, B_i)$ is differentiable almost everywhere, so is $l(\alpha, \theta; \mathcal{D})$. Accordingly, let θ be such that every pair of sequences A_i, B_i has a unique optimal configuration x_i, and let $U = \bigcap_i U^*(x_i)$ be the set of θ where all of these configurations are uniquely optimal. Then, the log-likelihood restricted to U can be expressed as

$$\begin{aligned} l(\alpha, \theta; \mathcal{D})_{|\theta \in U} = &\sum_{(A_i,B_i,y_i) \in \mathcal{D}} y_i(\alpha + \langle \theta, x_i \rangle) \\ &- \sum_{(A_i,B_i,y_i) \in \mathcal{D}} \ln\left(1 + \exp(\alpha + \langle \theta, x_i \rangle)\right). \end{aligned}$$

Now, $\sum_{(A_i,B_i,y_i) \in \mathcal{D}} y_i(\alpha + \langle \theta, x_i \rangle)$ is linear with respect to θ, making it both convex and concave. Therefore, $l(\alpha, \theta; \mathcal{D})_{|\theta \in U}$ is concave with respect to θ as a sum of concave functions. ◀

Proof of Propostition 10. First, we observe that the functions $\ln p(\alpha, \theta; A_i, B_i)$, $\ln(1 - p(\alpha, \theta; A_i, B_i))$ are locally Lipschitz continuous with respect to θ, so that their Clarke subdifferentials are well-defined. For a fixed intercept α, let $\varsigma(x; \alpha) \colon \mathbb{R} \to \mathbb{R}$ be the logistic function of x defined as $\varsigma(x; \alpha) = 1/(1 + \exp(-\alpha - x))$. This function is globally Lipschitz continuous with respect to x, because for any $x \in \mathbb{R}$ we have $\varsigma'(x) = \varsigma(x)(1 - \varsigma(x)) < 1$. Next, since $p(\alpha, \theta; A_i, B_i) = \varsigma(S_\theta(A_i, B_i); \alpha)$, and a composition of Lipschitz functions is Lipschitz, it follows that $p(\alpha, \theta; A_i, B_i)$ is also globally Lipschitz continuous with respect to θ. Finally, $\ln(x)$ is locally (but not globally) Lipschitz continuous. The proof for $\ln(1 - p(\theta))$ follows the same lines. We remark that the function $l(\alpha, \theta)$ is also locally Lipschitz with respect to θ as a sum of locally Lipschitz functions.

Now, the first equality in the Proposition follows directly from the chain rule in Lemma 9 applied to $f(\theta) = S_\theta(A, B)$ and $g_\alpha(x) = -\ln(1 + \exp(-\alpha - x))$ for a fixed α. The second equality follows from the chain rule applied to $g_\alpha(x) = -\ln(1 + \exp(\alpha + x))$. ◀

Proof of Lemma 11. First, we note that from the definition of the Clarke subdifferential, we have $\partial^C_\theta l(\alpha, \theta; \mathcal{D}) = \mathrm{Conv}(\{v_i\})$ for a set of vectors v_i, such that for each v_i there exists a sequence $\theta^i_n \to \theta$ such that $l(\alpha, \theta; \mathcal{D})$ is differentiable w.r.t. θ at each θ_n and $\nabla_\theta l(\alpha, \theta^i_n; \mathcal{D}) \to v_i$.

Assume $\theta_n \to \theta$ is such that $\mathcal{B}(\theta_n; \mathcal{D}) = \{(x_1, \ldots, x_{|\mathcal{D}|})\}$ for all n. Then, for every pair of sequences A_i, B_i, the score $S_{\theta_n}(A_i, B_i)$ is differentiable w.r.t. θ at each θ_n (as the alignment configurations are unique). It follows that $\nabla_\theta l(\alpha, \theta_n; \mathcal{D})$ exists at all θ_n. From Proposition 10, we get

$$\begin{aligned}\nabla_\theta l(\alpha, \theta_n; \mathcal{D}) &= \sum_{(A_i, B_i, y_i) \in \mathcal{D}} y_i(1 - p(\theta_n))x_i + (1 - y_i)(-p(\theta_n))x_i \\ &= \sum_{(A_i, B_i, y_i) \in \mathcal{D}} (y_i - p(\theta_n))x_i.\end{aligned}$$

From the continuity of p it follows that $\nabla_\theta l(\alpha, \theta_n; \mathcal{D}) \to \sum_{(A_i, B_i, y_i) \in \mathcal{D}} (y_i - p(\theta))x_i$, and thus $\sum_{(A_i, B_i, y_i) \in \mathcal{D}} (y_i - p(\theta))x_i \in \partial^C_\theta l(\alpha, \theta_n; \mathcal{D})$.

We remark that the reason why the Lemma states an inclusion, rather than equality, is that a sum of non-differentiable functions can be differentiable (e.g. $f(x) = |x|$ and $g(x) = -|x|$). Therefore, even if the alignment scores are not differentiable at points of some sequence $\theta_n \to \theta$, the log-likelihood potentially may be differentiable at these points and result in additional limits of gradients belonging to the Clarke subdifferential. Whether cases like that can happen in alignments remains an open question. ◀

Proof of Theorem 14. First, we note that the log-likelihood function $l(\alpha, \theta; \mathcal{D})$ is a "tame" function in the sense of Davis, Drusvyatsky, Kakade and Lee and satisfies their Assumption D [14]. This follows from the fact that the function is composed of a finite number of algebraic operations, maxima over finite numbers of functions, and logarithmic and exponential functions, which makes it definable in the o-minimal structure $\mathbb{R}_{\exp} = (\mathbb{R}, +, ., \exp, 0, 1)$ [71].

We will now show that the procedure satisfies Assumption C form [14]. The assumptions $\eta_k > 0, \sum \eta_k = \infty, \sum \eta_k^2 < \infty$ correspond to Assumption C.1, and the assumption that θ_k is bounded corresponds to Assumption C.2. Now, let $\alpha_k = \xi_k/\eta_k$, so that the update step can be written as $\theta_{k+1} = \theta_k + \eta_k(g_k + \alpha_k)$. Now, since $\mathbb{E}\xi_k = 0$, we also have $\mathbb{E}\alpha_k = 0$. Furthermore, $\mathbb{E}||\alpha_k||^2 = \frac{1}{\eta_k^2}\mathbb{E}||\xi_k||^2 = n\sigma_k^2/\eta_k^2$. Therefore, α_k is a sequence of independent random variables with zero expected value and a bounded variance, which satisfies Assumption C.3. ◀

Proof of Theorem 15. Let $g_k \in \partial_\theta^C l(\alpha, \theta; \mathcal{D}_k)$. Let $I_1^k, I_2^k, \ldots, I_{|\mathcal{D}|}^k$ be a sequence of independent binary random variables such that $I_i^k = 1$ if the i-th pair was sampled in iteration k. By construction of $\mathcal{D}_k$, we have $\mathbb{P}(I_i^k = 1) = N/|\mathcal{D}|$.

Since the Theorem assumes that g_k is constructed using Proposition 12, conditioned on the sequence $I^k = (I_1^k, \ldots, I_{|\mathcal{D}|}^k)$ it can be expressed as

$$g_k | I^k = \sum_{i=1}^{|\mathcal{D}|} I_i^k (y_i - p(\alpha, \theta; A_i, B_i)) x_i.$$

Let $\tilde{g}_k$ be a hypothetical subgradient obtained using Proposition 12 on the full dataset,

$$\tilde{g}_k = \sum_{i=1}^{|\mathcal{D}|} (y_i - p(\alpha, \theta; A_i, B_i)) x_i.$$

We now have

$$\mathbb{E} g_k = \frac{N}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} (y_i - p(\alpha, \theta; A_i, B_i)) x_i = \frac{N}{|\mathcal{D}|} \tilde{g}_k.$$

Let $\alpha_k = \xi_k / \eta_k$ and let $\beta_k = \tilde{g}_k - |\mathcal{D}|(g_k + \alpha_k)/N$. We can now write the θ update step as

$$\theta_{k+1} = \theta_k + \frac{N}{|\mathcal{D}|} \eta_k (\tilde{g}_k - \beta_k).$$

The Theorem now follows from Theorem 14 applied to the above sequence, treating $N\eta_k/|\mathcal{D}|$ as the sequence of step lengths and β_k as the sequence of random errors. ◀