Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Structural Bioinformatics

ConsAlign: simultaneous RNA structural aligner based on rich transfer learning and thermodynamic ensemble model of alignment scoring

Masaki Tagashira [1,*]

[1] Department of Computational Biology and Medical Sciences, University of Tokyo, Chiba 277-8561, Japan

* To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: To capture structural homology in RNAs, predicting RNA structural alignments has been a fundamental framework around RNA science. Learning simultaneous RNA structural alignments in their rich scoring is an undeveloped subject because evaluating them is computationally expensive.
Results: We developed ConsTrain—a gradient-based machine learning method for rich structural alignment scoring. We also implemented ConsAlign—a simultaneous RNA structural aligner composed of ConsTrain's learned scoring parameters. To aim for better structural alignment quality, ConsAlign employs (1) transfer learning from well-defined scoring models and (2) the ensemble model between the ConsTrain model and a mature thermodynamic scoring model. Keeping comparable running time, ConsAlign demonstrated competitive alignment prediction quality among current RNA structural aligners.
Availability and implementation: Our code and our data are freely available at `https://github.com/heartsh/consalign`.
Contact: heartsh@heartsh.io
Supplementary information: Supplementary data are available at *Bioinformatics*

1 Introduction

Machine learning is applied to each significant bioinformatics subject, including structural bioinformatics. Also, machine learning has attracted computational scientists' attention around folding single RNA sequences—or *mono-folding*. In mono-folding, modern parameter training methods optimize secondary structure scoring parameters based on both training RNA sequences and training RNA secondary structures (Do *et al*., 2006b; Zakov *et al*., 2011; Rivas *et al*., 2012; Singh *et al*., 2019; Sato *et al*., 2021). Folding RNA homolog sequences and aligning them at the same time—or *simultaneous RNA structural alignment* (Sankoff, 1985)—are an advanced solution, extending mono-folding. Simultaneous RNA structural alignment improves the folding quality of a focused RNA homolog sequence, incorporating other RNA homologs concerned. The machine learning methods of simultaneous RNA structural alignment have been developed less initiatively than those of mono-folding due to the complexity of the former (Dowell and Eddy, 2006; Do *et al*., 2008).

This study proposes *ConsTrain*—an *alignment score learning extension* of our sparse inside-outside algorithm (Tagashira and Asai, 2022). In Tagashira and Asai (2022), our sparse inside-outside algorithm scores possible simultaneous structural alignments by the scoring parameters of (1) Turner's (nearest neighbor physics) model for RNA secondary structures (Turner and Mathews, 2010) and (2) the CONTRAlign model for RNA sequence alignments (Do *et al*., 2006b). By ConsTrain, we intend to improve the predictive performance of our sparse inside-outside algorithm implemented using these two independent models. ConsTrain is based on the *pairwise conditional log-linear model* (= pair-CLLM)—a flexible discriminative training framework (Do *et al*., 2006a). Scoring parameter training for simultaneous RNA structural alignment has been proposed using not the pair-CLLM but the pairwise stochastic context-free grammar (= the pair-SCFG)—a classical generative training framework (Dowell and Eddy, 2006). Regarding mono-folding accuracy, CLLMs are generally superior to SCFGs due to SCFGs' model constraints (Do *et al*., 2006a).

To test the prediction performance of ConsTrain's trained alignment parameters, we build *ConsAlign*—a simultaneous RNA structural aligner parameterized with them. ConsAlign conducts sparse dynamic

© The Author 2015.

programming according to the *progressive alignment scheme* (Feng and Doolittle, 1987). ConsAlign depends on its hyper-parameters to control simultaneous structural alignment quality but corrects them automatically by our original scheme without exploiting any validation data. As our first essential contribution, we introduce *transfer learning* (Singh *et al*., 2019) based on conventional trained models into ConsTrain. As our second essential contribution, we also introduce a *model ensemble* (Aghaeepour and Hoos, 2013; Singh *et al*., 2019) between our thermodynamic probability inference and ConsTrain's trained probability inference into ConsAlign. We expect that ConsAlign will acquire higher precision and more robustness through our essential contributions.

2 Methods

2.1 Computing alignment scoring parameters' gradient

We can optimize the scoring parameters of each CLLM (including our structural alignment pair-CLLM), given their cost functions (Do *et al*., 2006a; Andronescu *et al*., 2007). From both the empirical occurrence counts and the expected occurrence counts of pair-CLLM scoring parameters in observed pairwise structural alignments (Fig. 1), we can derive the *gradient* of a pair-CLLM cost function. We compute the approximated expected counts of scoring parameters appearing in training pairwise structural alignments by conducting our inside-outside algorithm on sparse pairwise structural alignment (Tagashira and Asai, 2022) iteratively.

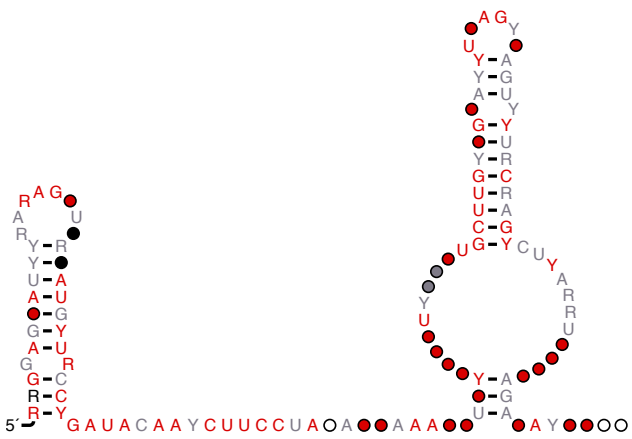

Fig. 1. RNA structural alignment is depicted in R-scape (Rivas et al., 2017).

2.1.1 Forming the cost of training pairwise structural alignments

We derive a *convex* pair-CLLM cost function fully parameterized with tunable alignment scoring parameters from training pairwise structural alignments. We score any pairwise structural alignment $\mathbb{A}$ by using a simple inner product:

$$s(\mathbb{A};\boldsymbol{\theta}) \stackrel{\text{def}}{=} \boldsymbol{\theta}^{\mathrm{T}} \cdot \boldsymbol{\phi}(\mathbb{A})$$

Here, $\boldsymbol{\theta}$ is any scoring parameter vector in the real F-space. In addition, a function $\boldsymbol{\phi}(\mathbb{A})$ maps any pairwise structural alignment $\mathbb{A}$ to the vector that counts the occurrence of each f-th scoring parameter $\theta_f : (\theta_f) \equiv \boldsymbol{\theta}, f \in \{1,\ldots,F\}$ in $\mathbb{A}$. This study assumes that any structural alignment $\mathbb{A}$ does not maintain any pseudoknot (Sato *et al*., 2011; Will *et al*., 2018) to simplify our proposed algorithms; we do not consider any base-pairing indels, i.e., we must match each base-pairing in one RNA secondary structure contained in $\mathbb{A}$ with some base-pairing in the other RNA secondary structure contained in $\mathbb{A}$, and vice versa, as implemented in many RNA structural aligners (Hofacker *et al*., 2004; Kiryu *et al*., 2007; Will *et al*., 2007; Do *et al*., 2008; Sato *et al*., 2012). The pairwise structural alignment scoring $s(\mathbb{A};\boldsymbol{\theta})$ obeys an underlying pair-CLLM. As a result, each alignment scoring parameter θ_f does not have to take the form of a log probability nor correspond only to a transition/emission (Do *et al*., 2006a).

Onto the training structural alignment $\mathbb{A}_d$ of each d-th RNA sequence pair, we obtain a convex pair-CLLM cost (Do *et al*., 2006a; Andronescu *et al*., 2007)

$$c(D;\boldsymbol{\theta}) \stackrel{\text{def}}{=} -\sum_{d:d\in\{1,\ldots,|D|\}} \ln \frac{\mathrm{e}^{s(\mathbb{A}_d;\boldsymbol{\theta})}}{Z_d(\boldsymbol{\theta})} \equiv \sum_d [\ln Z_d(\boldsymbol{\theta}) - s(\mathbb{A}_d;\boldsymbol{\theta})].$$

Here, any training alignment dataset D is $D \stackrel{\text{def}}{=} \{\mathbb{A}_d\}$. Furthermore, any alignment *partition function* $Z_d(\boldsymbol{\theta})$ is $Z_d(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d} \mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})}$. ($\mathcal{A}_d$ is the structural alignment *Boltzmann ensemble*—i.e., a set of all possible structural alignments—of each d-th sequence pair.) Our convex pair-CLLM cost $c(D;\boldsymbol{\theta})$ holds its gradient

$$\nabla c(D;\boldsymbol{\theta}) \stackrel{\text{def}}{=} (\frac{\partial c(D;\boldsymbol{\theta})}{\partial \theta_f}) \equiv \sum_{d:d\in\{1,\ldots,|D|\}} \{\mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}] - \boldsymbol{\phi}(\mathbb{A}_d)\}$$

because any log function gradient $(\frac{\partial \ln Z_d(\boldsymbol{\theta})}{\partial \theta_f})$ is

$$\begin{aligned}(\frac{\partial \ln Z_d(\boldsymbol{\theta})}{\partial \theta_f}) &\equiv (\frac{1}{Z_d(\boldsymbol{\theta})} \frac{\partial \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d} \mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})}}{\partial \theta_f}) \\ &\equiv \frac{1}{Z_d(\boldsymbol{\theta})} \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d} \boldsymbol{\phi}(\mathbb{A}')\mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})} \qquad (1) \\ &\equiv \mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}].\end{aligned}$$

Here, any expected count vector $\mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}]$ is

$$\mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}] \stackrel{\text{def}}{=} \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d} p_d(\mathbb{A}';\boldsymbol{\theta})\boldsymbol{\phi}(\mathbb{A}') \equiv \text{Equation (1)}.$$

[Any alignment *Boltzmann (probability) distribution* $p_d(\mathbb{A}';\boldsymbol{\theta})$ is $p_d(\mathbb{A}';\boldsymbol{\theta}) \stackrel{\text{def}}{=} \frac{\mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})}}{Z_d(\boldsymbol{\theta})}$.]

By using our sparse inside-outside algorithm, we can estimate the approximate amount of any expected count vector $\mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}]$ (Do *et al*., 2006a; Andronescu *et al*., 2007):

$$\mathbb{E}_d[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}] \equiv \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d} p_d(\mathbb{A}';\boldsymbol{\theta})\boldsymbol{\phi}(\mathbb{A}') \approx \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d^*} p_d^*(\mathbb{A}';\boldsymbol{\theta})\boldsymbol{\phi}(\mathbb{A}').$$

Here, $\mathcal{A}_d^*$ is a sparse version of any alignment Boltzmann ensemble $\mathcal{A}_d$. In addition, any sparse Boltzmann distribution $p_d^*(\mathbb{A}';\boldsymbol{\theta})$ is as follows:

$$\mathbb{A}' \sim p_d(\mathbb{A}';\boldsymbol{\theta}) \approx p_d^*(\mathbb{A}';\boldsymbol{\theta}) \stackrel{\text{def}}{=} \frac{\mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})}}{Z_d^*(\boldsymbol{\theta})}$$

$$Z_d(\boldsymbol{\theta}) \approx Z_d^*(\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d^*} \mathrm{e}^{s(\mathbb{A}';\boldsymbol{\theta})}.$$

This approximated vector estimation requires the same running time $O(L^2)$ and memory usage $O(L^2)$ as our sparse inside-outside algorithm (Tagashira and Asai, 2022). Here, L is the longer length of each d-th sequence pair.

2.2 Optimizing structural alignment scoring parameters

Our parameter optimization goal is to find structural alignment scoring parameters that minimize their convex pair-CLLM cost. We can utilize

general gradient-based optimization methods to complete our scoring parameter optimization. We aim to compute an optimized parameter vector

$$\boldsymbol{\theta}^* \stackrel{\text{def}}{=} \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^F} c^*(D;\boldsymbol{\theta})$$

$$c^*(D;\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{d:d\in\{1,\ldots,|D|\}} [\ln Z_d^*(\boldsymbol{\theta}) - s(\mathbb{A}_d;\boldsymbol{\theta})]$$

(where $\mathbb{R}^F$ is the real F-space) using the gradient of an approximated pair-CLLM cost $c^*(D;\boldsymbol{\theta})$

$$\nabla c^*(D;\boldsymbol{\theta}) \stackrel{\text{def}}{=} \sum_{d:d\in\{1,\ldots,|D|\}} \{\mathbb{E}_d^*[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}] - \boldsymbol{\phi}(\mathbb{A}_d)\}$$

$$\mathbb{E}_d^*[\boldsymbol{\phi}(\cdot);\boldsymbol{\theta}] \stackrel{\text{def}}{=} \sum_{\mathbb{A}':\mathbb{A}'\in\mathcal{A}_d^*} p_d^*(\mathbb{A}';\boldsymbol{\theta})\boldsymbol{\phi}(\mathbb{A}').$$

By Algorithm S1, we can conduct the above convex parameter optimization iteratively. Algorithm S1 is an instance of the Broyden-Fletcher-Goldfarb-Shanno (= BFGS) algorithm—a quasi-Newton optimization algorithm widely used (Fletcher, 1987).

Our approximated pair-CLLM cost $c^*(D;\boldsymbol{\theta})$ does not retain any regularization term. In other words, an optimized parameter vector $\boldsymbol{\theta}^*$ is prone to be overfitted for each training alignment dataset D. Thus, we minimize an L2-regularized non-convex cost instead of our unregularized convex cost $c^*(D;\boldsymbol{\theta})$:

$$\boldsymbol{\theta}^* \stackrel{\text{def}}{=} \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^F} c^{\mathrm{R}}(D;\boldsymbol{\theta},\alpha,\beta)$$

$$c^{\mathrm{R}}(D;\boldsymbol{\theta},\alpha,\beta) \stackrel{\text{def}}{=} c^*(D;\boldsymbol{\theta}) + \frac{1}{2}\boldsymbol{\lambda}^{\mathrm{T}}\cdot(\theta_f^2)$$

$$\boldsymbol{\lambda} \stackrel{\text{def}}{=} \left(\frac{\frac{|G(f)|}{2}+\alpha}{\beta+\frac{1}{2}\sum_{f':f'\in\{1,\ldots,F\}|G(f')=G(f)}\theta_{f'}^2}\right).$$

Here, α,β are the two gamma distribution parameters. Also, a function $G(f)$ returns the group where each alignment scoring parameter θ_f belongs. Our adaptive L2-regularization vector $\boldsymbol{\lambda}$ introduces a prior alignment parameter (probability) distribution:

$$\boldsymbol{\theta} \sim p(\boldsymbol{\theta};\alpha,\beta) \propto \Pi_{f=1}^F \Gamma\left(\frac{|G(f)|}{2}+\alpha\right)\Sigma_f(\boldsymbol{\theta};\beta)^{-\frac{|G(f)|}{2}-\alpha}$$

$$\Sigma_f(\boldsymbol{\theta};\beta) \stackrel{\text{def}}{=} \beta + \frac{1}{2}\sum_{f':f'\in\{1,\ldots,F\}|G(f')=G(f)}\theta_{f'}^2.$$

Here, the gamma function $\Gamma(r)$ given any real number r is $\Gamma(r) \stackrel{\text{def}}{=} \int_{t=0}^{\infty} t^{r-1}\mathrm{e}^{-t}dt$. We implement the above non-convex parameter optimization iteratively by Algorithm S2 based on Algorithm S1. Algorithm S2 is an example of *majorization-minimization algorithms*—iterative optimization algorithms that minimize some convex upper bound of any non-convex cost function (Langed *et al*., 2000). Algorithm S2 groups each alignment scoring parameter θ_f into a cluster and penalizes each scoring parameter cluster by a shared regularization constant (Foo *et al*, 2009). The advantage of introducing our prior parameter distribution $p(\boldsymbol{\theta};\alpha,\beta)$ is that we do not have to predetermine the value of our L2-regularization constant vector $\boldsymbol{\lambda}$ (e.g., by a grid search upon a validation structural alignment dataset) since we (in addition to our alignment scoring parameters $\boldsymbol{\theta}$ to be optimized) optimize $\boldsymbol{\lambda}$ in the same procedure (Foo *et al*, 2009). We implemented Algorithm S2 as the ConsTrain version 0.1.1 (`https://github.com/heartsh/consprob-trained`).

2.3 Configuring structural alignment scoring parameters

Borrowing knowledge from conventional scoring parameter optimizations, we specify the concrete settings of our alignment scoring parameters. We specify all the groups (or types) of our alignment scoring parameters, as described in Table S1. The *CONTRAfold model* is a CLLM that (from the training structure of each single RNA sequence) learns secondary structure scoring parameters based on RNA loop structures (Do *et al*., 2006a). The *CONTRAlign model* is a pair-conditional random field (Lafferty *et al*., 2001; Sato and Sakakibara, 2005) that trains sequence alignment scoring parameters from each sequence-aligned RNA pair (Do *et al*., 2006b). Our pair-CLLM inherits scoring parameter formats from both the CONTRAfold and the CONTRAlign models.

2.4 Our proposed structural aligner to maximize accuracy

2.4.1 Quadratic pairwise alignment guaranteeing its optimality

In the γ-centroid estimator principle (Ding *et al*., 2005; Hamada *et al*., 2009a,b,c, 2011), we define a pairwise structural alignment recursion like LocARNA (Will *et al*., 2007) and RAF (Do *et al*., 2008), as in Fig. 2:

$$a_{ijkl} \equiv b_{ik}(j-1,l-1) + \alpha_{ijkl}$$

$$b_{ik}(j,l) \equiv \max\begin{cases} b_{ik}(j-1,l) \\ b_{ik}(j,l-1) \\ b_{ik}(j-1,l-1) + \gamma^{\mathrm{M}} p_{jl}^{\mathrm{M}}(\boldsymbol{\theta}^*) - 1 \\ \max_{uv:i<u<j,k<v<l}[b_{ik}(u-1,v-1) + a_{ujvl}] \end{cases}$$

$$\alpha_{ijkl} \stackrel{\text{def}}{=} \gamma^{\mathrm{M}}[p_{ik}^{\mathrm{M}}(\boldsymbol{\theta}^*) + p_{jl}^{\mathrm{M}}(\boldsymbol{\theta}^*)] + \gamma^{\mathrm{P}}[p_{ij}^{\mathrm{P}}(\boldsymbol{\theta}^*) + p_{kl}^{\mathrm{P}}(\boldsymbol{\theta}^*)] - 4$$

starting from the initial condition $b_{ik}(i,k) \leftarrow 0$. Here, $p_{ik}^{\mathrm{M}}(\boldsymbol{\theta}^*)$ is the sparse posterior probability that any nucleotide (position) pair (i,k) matches in pairwise structural alignment given our trained scoring parameters $\boldsymbol{\theta}^*$; $p_{ij}^{\mathrm{P}}(\boldsymbol{\theta}^*)$ is the average sparse probabilistic consistency that every two nucleotides i,j base-pair in pairwise structural alignment given $\boldsymbol{\theta}^*$. The new ConsProb (Tagashira and Asai, 2022) parameterized with our trained scoring parameters $\boldsymbol{\theta}^*$ can compute trained sparse probabilities $p_{ik}^{\mathrm{M}}(\boldsymbol{\theta}^*), p_{ij}^{\mathrm{P}}(\boldsymbol{\theta}^*)$. Also, $\gamma^{\mathrm{M}},\gamma^{\mathrm{P}}$ are any two real numbers greater than 1. Hyper-parameters γ^{M} and γ^{P} control the predictive significance of nucleotide matching and nucleotide base-pairing, respectively.

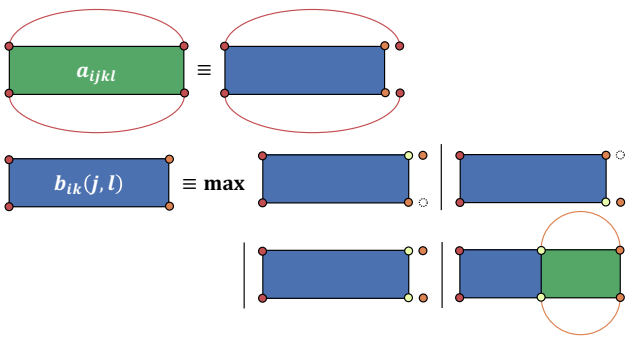

Fig. 2. Recursive diagram of our pairwise structural alignment.

We need to conduct our pairwise predictive recursion with $O(N^3M^3)$ running time and $O(N^2M^2)$ memory usage in a naive sense. (N, M are the lengths of every two RNA sequences.) However, we can perform our pairwise predictive recursion only with both $O(L^2)$ running time and $O(L)$ memory usage if hyper-parameters $\gamma^{\mathrm{M}},\gamma^{\mathrm{P}}$ are sufficiently small. [L is $L \stackrel{\text{def}}{=} \max(N,M)$.] We can realize this computational complexity reduction since our pairwise predictive recursion accepts only nucleotide

matches (i, k) satisfying the condition

$$p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*) \geq \frac{1}{\gamma^{\mathrm{M}}} \tag{2}$$

and nucleotide base-pairings $(i, j), (k, l)$ satisfying the conditions

$$p^{\mathrm{P}}_{ij}(\boldsymbol{\theta}^*) \geq \frac{1}{\gamma^{\mathrm{P}}}, p^{\mathrm{P}}_{kl}(\boldsymbol{\theta}^*) \geq \frac{1}{\gamma^{\mathrm{P}}}. \tag{3}$$

LocARNA and RAF impose Equations (2) and (3) upon possible pairwise structural alignments to enable both $O(L^2)$ running time and $O(L)$ memory usage (Do *et al*., 2008). Our pairwise predictive recursion assures that its obtained structural alignment is optimum (i.e., retaining the best score of all possible structural alignments) even if we impose Equations (2) and (3) upon possible pairwise structural alignments.

2.4.2 Extending pairwise alignment into multiple-alignment

Our structural multiple-alignment is based on the progressive alignment scheme (Feng and Doolittle, 1987) like conventional RNA structural aligners (Will *et al*., 2007; Do *et al*., 2008; Sato *et al*., 2012; Will *et al*., 2015; Li *et al*., 2021). When we build a *UPGMA guide tree* (Sneath and Sokal, 1962) from RNA homolog sequences to be aligned, we measure the efficient similarity between every two RNA homologs

$$\sigma(\boldsymbol{\theta}^*) \stackrel{\text{def}}{=} \sum_{ik} p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*) I[p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*) \geq \frac{1}{\gamma^{\mathrm{M}}}].$$

Here, $I[c]$ returns 1 if some condition c is true; otherwise, $I[c]$ returns 0. Given a hyper-parameter γ^{M}, our proposed metric $\sigma(\boldsymbol{\theta}^*)$ estimates the stochastic similarity between every two RNA homologs without aligning them. Our proposed metric $\sigma(\boldsymbol{\theta}^*)$ considers only nucleotide matches (i, k) satisfying the condition $p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*) \geq \frac{1}{\gamma^{\mathrm{M}}}$ to be eligible.

We extend our pairwise predictive recursion into a generalized multiple predictive recursion, replacing pairwise nucleotide matching probabilities $p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*)$ with alignment column matching probabilities

$$p^{\mathrm{M}}_{IK}(\boldsymbol{\theta}^*; X, Y) \stackrel{\text{def}}{=} \frac{1}{|X||Y|} \sum_{xy:x \in X, y \in Y} p^{\mathrm{M}}_{I_x K_y}(\boldsymbol{\theta}^*; x, y)$$

and replacing alignment-based average base-pairing probabilities $p^{\mathrm{P}}_{ij}(\boldsymbol{\theta}^*), p^{\mathrm{P}}_{kl}(\boldsymbol{\theta}^*)$ with alignment column base-pairing probabilities

$$p^{\mathrm{P}}_{IJ}(\boldsymbol{\theta}^*; X) \stackrel{\text{def}}{=} \frac{1}{|X|} \sum_{x:x \in X} p^{\mathrm{P}}_{I_x J_x}(\boldsymbol{\theta}^*; x)$$

$$p^{\mathrm{P}}_{KL}(\boldsymbol{\theta}^*; Y) \stackrel{\text{def}}{=} \frac{1}{|Y|} \sum_{y:y \in Y} p^{\mathrm{P}}_{K_y L_y}(\boldsymbol{\theta}^*; y).$$

Here, X, Y are every two aligned sets of RNA homolog sequences to be aligned. Moreover, I, J, K, L are every four alignment columns, and I_x is any RNA homolog x's nucleotide stored in I. $p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*; x, y)$ is the explicit form of each nucleotide matching probability $p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*)$ regarding every two RNA homologs x, y; and $p^{\mathrm{P}}_{ij}(\boldsymbol{\theta}^*; x)$ is the explicit form of each average base-pairing probability $p^{\mathrm{P}}_{ij}(\boldsymbol{\theta}^*)$ regarding x.

2.4.3 Auto-correction of structural alignment hyper-parameters

The disadvantage of γ-centroid estimators is that they rely on hyper-parameters (e.g., hyper-parameters $\gamma^{\mathrm{M}}, \gamma^{\mathrm{P}}$ in our structural alignment). In practice, we have to determine hyper-parameters before performing γ-centroid estimators. This study proposes a reasonable way to determine hyper-parameters in performing γ-centroid simultaneous structural alignments. The *sum-of-pairs score* (= SPS) of any predicted structural alignment $\mathbb{A}$ against any possible multiple alignment $\mathbb{A}'$ is

$$\sigma(\mathbb{A}, \mathbb{A}') \stackrel{\text{def}}{=} \frac{1}{C} \sum_{xy:x<y} \{ \sum_{ik:(i,k) \in M_{xy}(\mathbb{A})} I[(i,k) \in M_{xy}(\mathbb{A}')] + \sum_{i:i \in I_{xy}(\mathbb{A})} I[i \in I_{xy}(\mathbb{A}')] + \sum_{k:k \in D_{xy}(\mathbb{A})} I[k \in D_{xy}(\mathbb{A}')] \}.$$

Here, an SPS denominator C is

$$C \stackrel{\text{def}}{=} \sum_{xy:x<y} [\sum_{ik:(i,k) \in M_{xy}(\mathbb{A})} 1 + \sum_{i:i \in I_{xy}(\mathbb{A})} 1 + \sum_{k:k \in D_{xy}(\mathbb{A})} 1].$$

In addition, $M_{xy}(\mathbb{A})$ is a set of all aligned nucleotide pairs between every two RNA homologs x, y in any structural alignment $\mathbb{A}$; $I_{xy}(\mathbb{A})$ is a set of all inserted nucleotides between x, y in $\mathbb{A}$; $D_{xy}(\mathbb{A})$ is a set of all deleted nucleotides between x, y in $\mathbb{A}$.

As a result, we can measure the approximated expected SPS of any predicted structural alignment $\mathbb{A}$ against all possible multiple alignments

$$\mathbb{E}[\sigma(\mathbb{A}, \cdot); \boldsymbol{\theta}^*] \stackrel{\text{def}}{=} \frac{1}{C} \sum_{xy:x<y} [\sum_{ik:(i,k) \in M_{xy}(\mathbb{A})} p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*; x, y) + \sum_{i:i \in I_{xy}(\mathbb{A})} p^{\mathrm{I}}_{i}(\boldsymbol{\theta}^*; x, y) + \sum_{k:k \in D_{xy}(\mathbb{A})} p^{\mathrm{D}}_{k}(\boldsymbol{\theta}^*; x, y)].$$

Here, $p^{\mathrm{I}}_{i}(\boldsymbol{\theta}^*; x, y)$ is $p^{\mathrm{I}}_{i}(\boldsymbol{\theta}^*; x, y) \stackrel{\text{def}}{=} 1 - \sum_{k:1 \leq k \leq M} p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*; x, y)$, and $p^{\mathrm{D}}_{k}(\boldsymbol{\theta}^*; x, y)$ is $p^{\mathrm{D}}_{k}(\boldsymbol{\theta}^*; x, y) \stackrel{\text{def}}{=} 1 - \sum_{i:1 \leq i \leq N} p^{\mathrm{M}}_{ik}(\boldsymbol{\theta}^*; x, y)$. Finally, we can optimize hyper-parameters $\gamma^{\mathrm{M}}, \gamma^{\mathrm{P}}$ as well as a predicted structural alignment by enumerating possible values of $\gamma^{\mathrm{M}}, \gamma^{\mathrm{P}}$ as in Algorithm S3. Our alignment hyper-parameter auto-correction is similar to hyper-parameter auto-correction used in mono-folding—called pseudo-expected accuracy optimization (Hamada *et al*., 2010; Sato and Kato, 2022)—because both these methods optimize γ-centroid hyper-parameters based on posterior probabilities.

2.4.4 *Ali-folding* post-process of predicted structural alignment

Folding RNA alignment—or ali-folding—is one of the most potent ways to find *conserved base pairing*s in RNA alignment as well as simultaneous RNA structural alignment (Bernhart *et al*., 2008; Seemann *et al*., 2008; Hamada *et al*., 2011; Tagashira and Asai, 2022). As LocARNA adopts RNAalifold (Bernhart *et al*., 2008) as the ali-folding post-processor of LocARNA's predicted structural alignment (Will *et al*., 2007), we adopt ConsAlifold (Tagashira and Asai, 2022) parameterized with our trained alignment parameters as the ali-folding post-processor of our predicted structural alignment.

2.4.5 Ensemble of different alignment scoring models

Unifying different stochastic models for a prediction—referred to as a model ensemble—has been utilized in mono-folding to achieve more accurate structure prediction (Aghaeepour and Hoos, 2013; Singh *et al*., 2019). In our structural alignment prediction, we combine our sparse inside-outside algorithm of Tagashira and Asai (2022) and that of this study to create an alignment scoring model ensemble. More specifically, we use average column matching probabilities

$$p^{\mathrm{M}}_{IK}(\boldsymbol{\theta}^{\mathrm{old}}, \boldsymbol{\theta}^*; X, Y) \stackrel{\text{def}}{=} \frac{1}{2}[p^{\mathrm{M}}_{IK}(\boldsymbol{\theta}^{\mathrm{old}}; X, Y) + p^{\mathrm{M}}_{IK}(\boldsymbol{\theta}^*; X, Y)]$$

and average column base-pairing probabilities

$$p^{\mathrm{P}}_{IJ}(\boldsymbol{\theta}^{\mathrm{old}}, \boldsymbol{\theta}^*; X) \stackrel{\text{def}}{=} \frac{1}{2}[p^{\mathrm{P}}_{IJ}(\boldsymbol{\theta}^{\mathrm{old}}; X) + p^{\mathrm{P}}_{IJ}(\boldsymbol{\theta}^*; X)].$$

Here, $\boldsymbol{\theta}^{\mathrm{old}}$ is the alignment scoring parameters used in Tagashira and Asai (2022). Our old scoring parameters $\boldsymbol{\theta}^{\mathrm{old}}$ are from both Turner's and

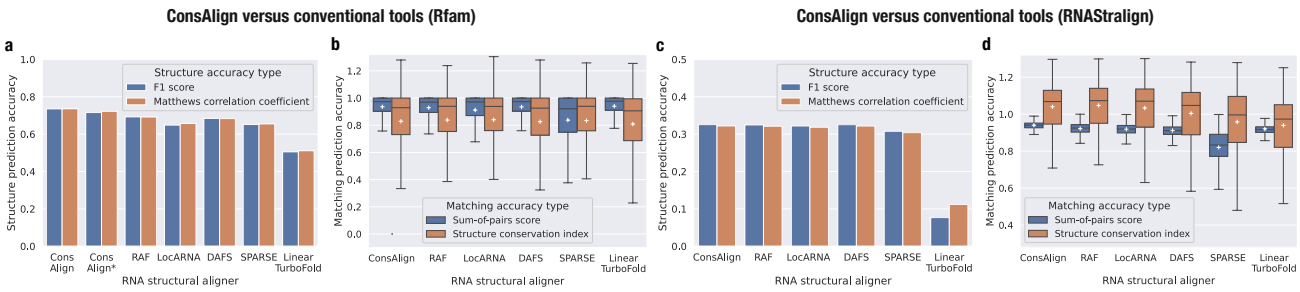

Fig. 3. Alignment prediction accuracy comparison among ConsAlign and conventional structural aligners. (a, c) Methods were compared in terms of F1 scores and MCCs. (b, d) Each method's distributions of SPSs and SCIs are visualized in box plots. Outliers were omitted from box plots. Each white cross indicates a corresponding method's average. The datasets (a, b) "test multiple" and (c, d) "RNAStralign processed" were used. ConsAlign was based on our proposed ensemble model and our proposed transfer learning. (ConsAlign*) ConsAlign disabled its RNAalifold subroutine from producing alignment column base-pairing probabilities (Bernhart et al., 2008) to check ConsAlign's pure prediction performance.

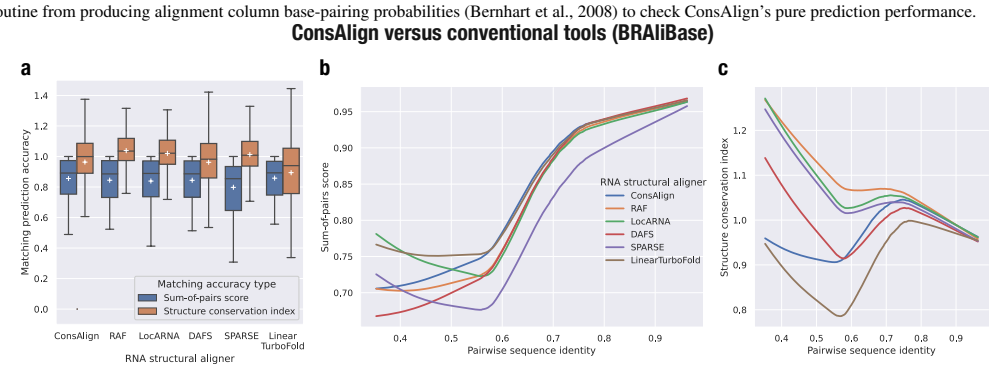

Fig. 4. Predicted alignments' quality among modern structural aligners. (a) Each method's SPSs and SCIs are visualized in box plots. Each method's (b) SPSs and (c) SCIs are expressed in lowess curves (Cleveland, 1981) to the pairwise sequence identity of reference alignments. BRAliBase's dataset 1 was used.

the CONTRAlign models, while our trained scoring parameters $\boldsymbol{\theta}^*$ are from the CONTRAfold and the CONTRAlign models. We implemented our RNA structural aligner explained above as the ConsAlign version 0.1.1 (`https://github.com/heartsh/consalign`).

2.5 Benchmarking data

We used the Rfam version 14.4 (Kalvari *et al*., 2018) for our alignment parameter training and software benchmarking test. From Rfam, we collected 3129 structurally aligned families as the dataset "origin." The seed reference structural alignment of each collected RNA family retained at most 500 columns and 20 sequences. We randomly split the dataset "origin" into both the equal datasets "train origin" and "test multiple." From the dataset "train origin," we collected each RNA family whose seed reference consensus structure is not any prediction product [e.g., neither by Pfold (Knudsen and Hein, 2003) nor by RNAalifold (Bernhart *et al*., 2008)] as the dataset "train pure." We random-sampled 10 RNA sequences from each RNA family holding at least 10 RNA sequences in the dataset "train pure." Then, we retrieved the all-to-all pairwise structural alignments of these 10 RNA sequences, finally obtaining the dataset "train pairwise." This random sequence sampling prevents our alignment parameter learning from overfitting due to the overwhelming number of the RNA homologs belonging to the same large RNA family (Do *et al*., 2006a, 2008).

As a result, (1) the dataset "train pairwise" is composed of 2250 pairwise structural alignments, and (2) the dataset "test multiple" is composed of 1565 multiple structural alignments. We used the dataset "train pairwise" in order to train ConsTrain. We used the dataset "test multiple" to assess the prediction performances of each benchmarked prediction method. Any entry in the dataset "train pairwise" and any entry in the dataset "test multiple" are not derived from the same family to prevent structural homology from leaking between training data and test data (Rivas *et al*., 2012; Sato *et al*., 2021).

As another test data source, we adopted reference RNA structural alignments registered in RNAStralign (Tan *et al*., 2017). We split each complete structural alignment comprising all RNA sequences into the sub-alignment comprising every 20 or fewer sequences. At last, we obtained the dataset "RNAStralign processed," composed of 917 multiple structural alignments. From the BRAliBase version 2 (Wilm *et al*., 2006), we also downloaded dataset 1, containing reference RNA sequence alignments, as a test dataset.

2.6 Conventional structural aligners for experiments

Using default command line arguments, we compared (1) the RAF version 1.00, (2) the LocARNA version 2.0.0, (3) the SPARSE version 2.0.0 (Will *et al*., 2015), (4) the DAFS version 0.0.3 (Sato *et al*., 2012), and (5) the LinearTurboFold version 1 (Li *et al*., 2021) to ConsAlign. Each conventional aligner's characteristics are described in Section S4.

2.7 Alignment prediction accuracy metrics

Across all test structural alignments, we find the numbers of true positives, true negatives, false positives, and false negatives TP, TN, FP, FN regarding structure quality while benchmarking each tool. We compute an F1 score and a Matthews correlation coefficient (= MCC) in each test alignment dataset:

$$F1 \stackrel{\text{def}}{=} \frac{2 \times TP}{2 \times TP + FP + FN}$$

$$MCC \stackrel{\text{def}}{=} \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}.$$

To measure the matching quality of each predicted structural alignment, we obtain both the SPS and the structure conservation index (= SCI) of each predicted structural alignment. Each SCI quantifies the structural

conservation of a corresponding RNA sequence alignment from a thermodynamic viewpoint (Washietl *et al*., 2005) and, therefore, diagnoses the matching quality of each predicted RNA alignment considering potential structure quality.

2.8 Implementations and benchmark environments

Employing multi-threading to provide a more comfortable user experience, we implemented ConsTrain and ConsAlign in Rust. ConsTrain used the two gamma distribution parameters $\alpha \leftarrow 0, \beta \leftarrow 1$ proposed by Foo *et al* (2009). In ConsTrain, we implemented two different initialization methods for structural alignment scoring parameters to be trained. (1) In our first initialization method, corresponding scoring parameter values were transferred from the CONTRAfold and the CONTRAlign models. (2) In the other initialization method, structural alignment scoring parameters were initialized by random values derived from the normal distribution with both the mean 0 and the variance $\frac{1}{F}$. ConsAlign used the four lower/upper bounds $i^{\min} \leftarrow 0, i^{\max} \leftarrow 7, j^{\min} \leftarrow 0, j^{\max} \leftarrow 3$ found in Algorithm S3. Also, ConsAlifold performed in ConsAlign took the γ-centroid hyper-parameter $2^0 + 1 = 2$ because this γ-centroid hyper-parameter value showed the highest structure prediction accuracy among the possible γ-centroid hyper-parameter values $2^{-4} + 1, \ldots, 2^{10} + 1$ in Tagashira and Asai (2022). For our prediction software benchmarks, we used an "Intel Xeon" CPU computer with $240\,\mathrm{GB}$ of RAM, 64 threads, and a clock rate of $2.30\,\mathrm{GHz}$. (We refer to this computer as the environment "Xeon.")

3 Results

3.1 Benchmarking ConsAlign to conventional methods

ConsAlign consistently performed better structure prediction accuracy than the other RNA structural aligners, whereas LinearTurboFold (Li *et al*., 2021) had the lowest rank regarding secondary structure prediction accuracy (Fig. 3a). ConsAlign showed comparable SPSs among state-of-the-art RNA structural aligners, while SPARSE (Will *et al*., 2015) indicated the poorest SPSs (Fig. 3b). Regarding SCIs, we did not observe any significant difference among all compared structural aligners except LinearTurboFold (Fig. 3b). ConsAlign took the median prediction running time among available RNA structural aligners (Table 1).

Table 1. Prediction type and running time of benchmarked structural aligners.

Structural aligner	Structural alignment type	Total running time
ConsAlign	Simultaneous	442 s
RAF		1697 s
LocARNA	⋮	70 s
SPARSE		618 s
DAFS	Simultaneous	84 s
LinearTurboFold	Iterative	1362 s

The center column indicates each aligner's prediction type. The right column shows the total running time of each benchmarked structural aligner on the dataset "test multiple." Utilizing `multiprocessing.Pool` (`https://docs.python.org/3/library/multiprocessing.html`) to relieve the large volume of the dataset "test multiple," each benchmarked structural aligner other than ConsAlign was run on the environment "Xeon" by mapping 64 processes. ConsAlign was based on our proposed ensemble model and our proposed transfer learning. Moreover, ConsAlign was conducted by providing each of 16 processes with four different threads.

When we switched our test dataset from the dataset "test multiple" to the dataset "RNAStralign processed," ConsAlign competed with most RNA structural aligners regarding predicted structures' quality, separating LinearTurboFold from the others (Fig. 3c). ConsAlign was superior to conventional RNA structural aligners from the viewpoint of SPS, and SPARSE suffered lower SPSs compared to the others (Fig. 3d). The SCIs of ConsAlign were comparable to those of RAF, LocARNA, and DAFS (Fig. 3d). Surprisingly, LinearTurboFold was more disadvantageous to SCI than the others, albeit performing moderate SPSs (Fig. 3d).

In BRAliBase's dataset 1, ConsAlign and the existing structural aligners other than SPARSE were almost equally precise regarding SPSs (Fig. 4a–b). However, SPARSE's predicted structural alignments were graded highly together with RAF and LocARNA in terms of SCIs (Fig. 4a,c).

3.2 Verifying our proposed training and model ensemble

ConsTrain took only about 30 epochs when using our proposed transfer learning (Fig. S1). On the other hand, ConsTrain consumed around 60 epochs when we initialized our alignment scoring parameters to random values (Fig. S1). ConsTrain continuously reduced our L2-regularized non-convex cost in both parameter initialization methods (Fig. S1). In each test dataset, ConsAlign's expected SPSs highly correlated with its observed SPSs (Figs. S2 and 5), and ConsAlign's tuned hyper-parameters were principally distributed around lower bounds (Figs. S2 and 6).

In the dataset "test multiple," ConsTrain with our proposed transfer learning demonstrated the best structure prediction accuracy among available parameter training schemes (Fig. 7a). Demonstrated parameter training schemes showed the same level of predicted matches' quality (Fig. 7b). Our proposed ensemble model drew advantages from the old ConsProb, which outperformed the new ConsProb, in structural accuracy (Fig. 8a). In contrast, performed alignment scoring models were nearly equivalent for SPS and SCI (Fig. 8b).

Shifting to the dataset "RNAStralign processed," ConsAlign was not enhanced at any alignment quality criterion when we focused on our transfer learning against our scoring parameters that were only transferred (Fig. S3). Regarding both structural and matching quality, our proposed ensemble model capitalized on the new ConsProb, which surpassed the old ConsProb (Fig. S4). From BRAliBase's dataset 1, predicted matches' quality was raised in ConsAlign when we compared our transfer learning with the others (Fig. 9a–b, Fig. S5a). Our proposed ensemble model used the new ConsProb and the old ConsProb to perform the best SPSs and SCIs (Fig. 9c–d, Fig. S5b).

4 Conclusion and discussion

We invented ConsTrain—an efficient gradient-based learning method of simultaneous alignment scoring parameters—and applied it to ConsAlign—a model ensemble-based structural aligner. ConsAlign attained competitive alignment quality in our alignment tool benchmarks. In our alignment software comparison, LinearTurboFold and SPARSE were the weakest in structural prediction accuracy and SPSs, respectively. LinearTurboFold is an iterative method different from the other benchmarked simultaneous methods. Therefore, we can interpret that simultaneous solutions enhanced obtained structures' quality by the more direct transmission of structural conservation among homologs than iterative solutions. SPARSE realizes its efficient computation by employing only structure-based constraints, whereas the other simultaneous structural aligners—ConsAlign, LocARNA, RAF, and DAFS—employ matching-based constraints in addition to structure-based constraints. We can explain that leveraging both matching-based and

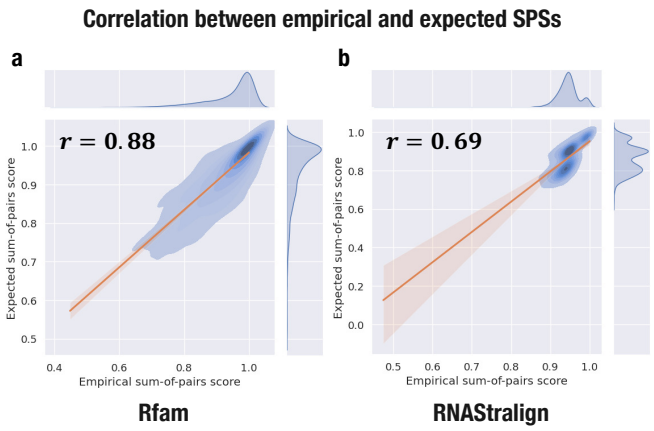

Fig. 5. Contrast between ConsAlign's empirical and expected SPSs in the datasets (a) "test multiple" and (b) "RNAStralign processed." (r) Each Pearson correlation coefficient was noted. ConsAlign was based on our proposed ensemble model and our proposed transfer learning.

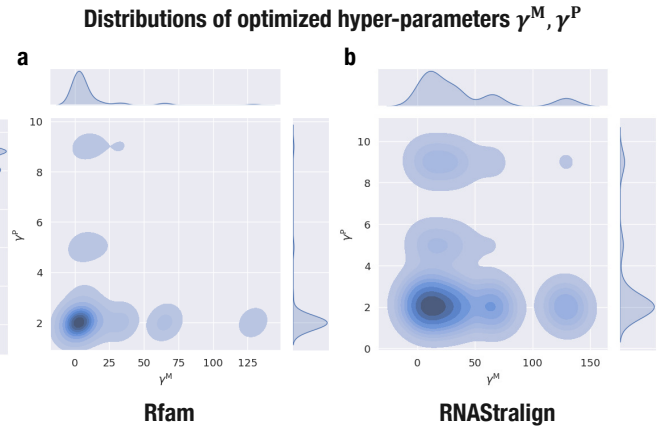

Fig. 6. Distributions of ConsAlign's optimized hyper-parameters $\gamma^{\mathrm{M}}, \gamma^{\mathrm{P}}$ in the datasets (a) "test multiple" and (b) "RNAStralign processed." ConsAlign was based on our proposed ensemble model and our proposed transfer learning.

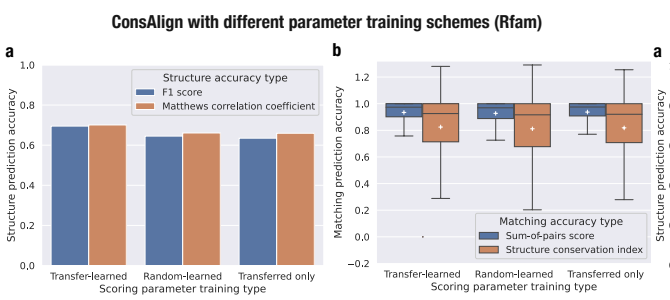

Fig. 7. Alignment prediction accuracy comparison among the different parameter training schemes of ConsTrain. (Transfer-learned) Structural alignment scoring parameters were transfer-learned. (Random-learned) Random values initialized structural alignment scoring parameters before training. (Transferred only) Structural alignment scoring parameters were transferred but then were not trained. This figure's configuration is the same as Fig. 3. The dataset "test multiple" was used. ConsAlign was based on the new ConsProb (Tagashira and Asai, 2022) alone and disabled an RNAalifold subroutine.

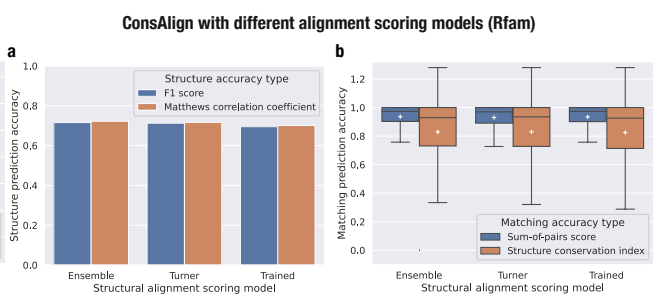

Fig. 8. Alignment prediction accuracy comparison among the different alignment scoring models of ConsAlign. (Ensemble) Our proposed ensemble model was used. (Turner) Only the old ConsProb was used as a posterior probability inference method. (Trained) Only the new ConsProb was used as a posterior probability inference method. This figure's configuration is the same as Fig. 3. The dataset "test multiple" was used. ConsAlign was based on our proposed transfer learning and disabled an RNAalifold subroutine.

structure-based constraints is more effective regarding obtained nucleotide matches' quality than only structure-based constraints.

ConsTrain with our transfer learning substantiated double parameter training efficiency and better structural alignment quality than our random parameter initialization. ConsTrain's iterative parameter update is not convex due to our attached training hyper-parameters. Thus, we can conclude that providing well-defined scoring parameter values guided ConsTrain to better machine learning solutions. Our proposed ensemble model made the flexible use of both the new ConsProb and the old ConsProb to achieve ConsAlign's robust alignment quality. In other words, both thermodynamics-based and entirely statistical models compensated each other for predictive weak points.

Recently, deep learning-based methods for mono-folding—such as SPOT-RNA (Singh *et al*., 2019), MXfold (Sato *et al*., 2021), and Ufold (Fu *et al*., 2022)—emerged, replacing non-deep learning methods for mono-folding—such as CONTRAfold (Do *et al*., 2006b) and TORNADO (Rivas *et al*., 2012). Expanding ConsTrain into deep learning for simultaneous RNA structural alignments has the potential to improve their quality by in-depth features unable to capture in any CLLMs. In another scenario, we have a chance to generalize the standard RNA base alphabet (`A`, `C`, `G`, and `U`) used in ConsTrain by including modified bases—such as `I` and Ψ (Harcourt *et al*., 2017). However, we need well-defined training datasets containing modified bases in this scenario because the base composition of training datasets directly affects ConsTrain's trained scoring parameters via gradients. Constructing reliable large structural databases for modified bases can assist machine learning methods involving RNA secondary structures in data-driven avenues.

Acknowledgments

We would like to thank Dr. Kiyoshi Asai, Dr. Martin Frith, and their lab members for a year of discussion. We incredibly appreciate valuable comments from Dr. Kengo Sato and Dr. Akito Taneda regarding points to be improved. We performed most of the computations in this study on the NIG supercomputer at the ROIS National Institute of Genetics, Japan.

Funding

This work was not supported by any funding.

Conflict of Interest

None declared.

Data Availability

Our underlying data are available at our shown GitHub repositories.

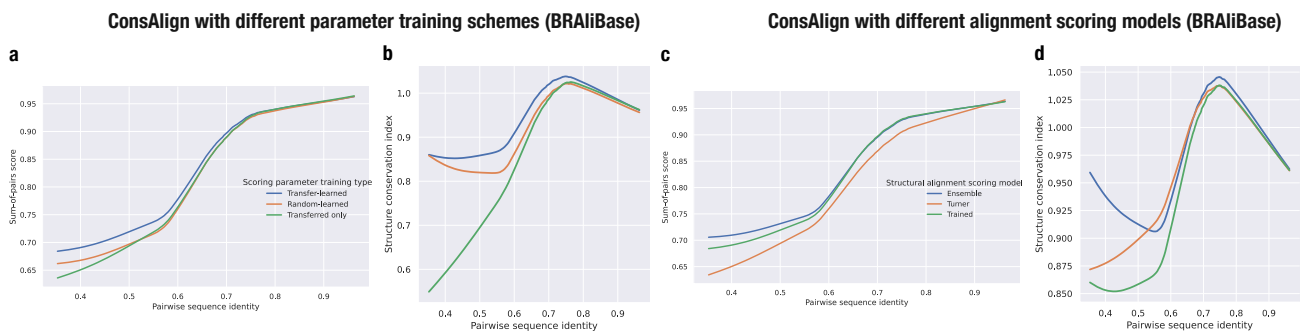

Fig. 9. Alignment prediction accuracy comparison among (a, b) the different parameter training schemes and (c, d) the different alignment scoring models of ConsAlign. This figure's configuration is the same as Figs. 7 and 8. BRAliBase's dataset 1 was used.

References

Aghaeepour,N. and Hoos,H.H. (2013) Ensemble-based prediction of RNA secondary structures. *BMC Bioinf.*, **14**, 139.

Andronescu,M. *et al*. (2007) Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, **23**, i19–i28.

Bernhart,S.H. *et al*. (2008) RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinf.*, **9**, 474.

Cleveland,W.S. (1981) LOWESS: A Program for Smoothing Scatterplots by Robust Locally Weighted Regression. *Am. Stat.*, **35**, 54.

Ding,Y., Chi,Y.C. and Lawrence,C.E. (2005) RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, **11**, 1157–1166.

Do,C.B., Woods,D.A. and Batzoglou,S. (2006a) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**, e90–e98.

Do,C.B., Gross,S.S. and Batzoglou,S. (2006b) *CONTRAlign: Discriminative Training for Protein Sequence Alignment*. In Proceedings of the Tenth Annual International Conference on Computational Molecular Biology, RECOMB.

Do,C.B., Foo,C. and Batzoglou,S. (2008) A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics*, **24**, i68–i76.

Dowell,R.D. and Eddy,S.R. (2006) Efficient pairwise RNA structure prediction and alignment using sequence alignment constraints. *BMC Bioinf.*, **7**, 400.

Feng,D. and Doolittle,R.F. (1987) Progressive sequence alignment as a prerequisitetto correct phylogenetic trees. *J. Mol. Evol.*, **25**, 351–360.

Fletcher,R. (1987) *Practical Methods of Optimization*. John Wiley & Sons, Ltd, New York.

Foo,C.S., Do,C.B. and Ng,A.Y. (2009) *A majorization-minimization algorithm for (multiple) hyperparameter learning.* ACM International Conference Proceeding Series.

Fu,L., Cao,Y., Wu,J., Peng,Q., Nie,Q., Xie,X. (2022) UFold: fast and accurate RNA secondary structure prediction with deep learning. *Nucleic Acids Res.*, **50**, e14.

Hamada,M. *et al*. (2009a) Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, **25**, 465–473.

Hamada,M. *et al*. (2009b) Predictions of RNA secondary structure by combining homologous sequence information. *Bioinformatics*, **25**, i330–i338.

Hamada,M. *et al*. (2009c) CentroidAlign: fast and accurate aligner for structured RNAs by maximizing expected sum-of-pairs score.

Hamada,M., Sato,K. and Asai,K. (2010) Prediction of RNA secondary structure by maximizing pseudo-expected accuracy. *BMC Bioinf.*, **11**, 586. *Bioinformatics*, **25**, 3236–3243.

Hamada,M., Sato,K. and Asai,K. (2011) Improving the accuracy of predicting secondary structure for aligned RNA sequences. *Nucleic Acids Res.*, **39**, 393–402.

Harcourt,E.M., Kietrys,A.M. and Kool,E.T. (2017) Chemical and structural effects of base modifications in messenger RNA. *Nature*, **541**, 339–346.

Hofacker,I.L., Bernhart,S.H.F. and Stadler,P.F. (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**, 2222–2227.

Kalvari,I. *et al*. (2018) Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res.*, **46**, D335–D342.

Kiryu,H. *et al*. (2007) Murlet: a practical multiple alignment tool for structural RNA sequences. *Bioinformatics*, **23**, 1588–1598.

Knudsen,B. and Hein,J. (2003) Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, **31**, 3423–3428.

Lafferty,J., McCallum,A. and Pereira,F. (2001) *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data.* Proceedings of the 18th International Conference on Machine Learning 2001, 282–289.

Langed,K., Hunter,A.R. and Yang,I. (2000) Optimization Transfer Using Surrogate Objective Functions. *J. Comput. Graph. Stat.*, **9**, 1–20.

Li,S. *et al*. (2021) LinearTurboFold: Linear-time global prediction of conserved structures for RNA homologs with applications to SARS-CoV-2. *Proc. Natl. Acad. Sci. U. S. A.*, **118**.

Rivas,E., Lang,R. and Eddy,S.R. (2012) A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA*, **18**, 193–212.

Rivas,E., Clements,J., Eddy,S.R. (2017) A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat. Methods*, **14**, 45–48.

Sankoff,D. (1985) Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Sato,K. and Sakakibara,Y. (2005) RNA secondary structural alignment with conditional random fields. *Bioinformatics*, **21**, ii237–ii242.

Sato,K. *et al*. (2011) IPknot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, **27**, i85–i93.

Sato,K. *et al*. (2012) DAFS: simultaneous aligning and folding of RNA sequences via dual decomposition. *Bioinformatics*, **28**, 3218–3224.

Sato,K., Akiyama,M. and Sakakibara,Y. (2021) RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat Commun.*, **12**, 1–9.

Sato,K. and Kato,Y. (2022) Prediction of RNA secondary structure including pseudoknots for long sequences. *Brief Bioinform.*, **23**, 1–9.

Seemann,S.E., Gorodkin,J. and Backofen,R. (2008) Unifying evolutionary and thermodynamic information for RNA folding of multiple alignments. *Nucleic Acids Res.*, **36**, 6355–6362.

Singh,J. *et al*. (2019) RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nat. Commun.*, **10**, 1–13.

Sneath,P.H. and Sokal,R.R. (1962) Numerical taxonomy. *Nature*, **193**, 855–860.

Tagashira,M. and Asai,K. (2022) ConsAlifold: considering RNA structural alignments improves prediction accuracy of RNA consensus secondary structures. *Bioinformatics*, **38**, 710–719.

Tan,Z. *et al*. (2017) TurboFold II: RNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res.*, **45**, 11570–11581.

Turner,D.H. and Mathews,D.H. (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, **38**, D280–D282.

Washietl,S., Hofacker,I.L. and Stadler,P.F. (2005) Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. U. S. A.*, **102**, 2454–2459.

Will,S., Reiche,K., Hofacker,I.L., Stadler,P.F. and Backofen,R. (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, 680–691.

Will,S. *et al*. (2015) SPARSE: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics. *Bioinformatics*, **31**, 2489–2496.

Jabbari,H., Wark,I., Montemagno,C. and Will,S. (2018) Knotty: efficient and accurate prediction of complex RNA pseudoknot structures. Valencia A, editor. *Bioinformatics*, **34**, 3849–3856.

Wilm,A., Mainz,I. and Steger,G. (2006) An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol. Biol.* **1**, 19.

Zakov,S., Goldberg,Y., Elhadad,M. and Ziv-Ukelson, M. (2011) Rich parameterization improves RNA structure prediction. *J. Comput. Biol.*, **18**, 1525–1542.