Appendix

In this Additional file we provide a detailed description of the algorithms involved in implementing the three probabilistic models components of our comparative method. Here we give the most general description of the scoring/parsing algorithms. We indicate at some different points how to obtain some simplifications that are part of the software implementation.

Assume we have two aligned sequences X and Y from two related organisms of respective lengths n and m. Unprimed coordinates "i" (with $0 \leq i < n$) will describe positions in sequence X, and primed coordinates "i'" (with $0 \leq i' < m$) will describe positions in sequence Y.

The IID Model algorithm

The IID model is used to report the probability scores of the OTH, COD, and RNA models in a log-odds version. The IID model allows us to emit two sequences X, Y of varying length independently from each other. The IID model is the aggregation of two single HMMs as described in Figure 4. The probability of sequences X and Y being emitted by the IID model in Figure 4 is

$$P(X, Y \mid \text{IID}) = \eta^2 (1-\eta)^{n+m} \prod_{i=0}^{n-1} P^X(a_i) \prod_{i'=0}^{m-1} P^Y(b_{i'}), \tag{1}$$

where the single nucleotide emission probabilities are the ones defined in (**??**) and (**??**).

Because the sequences are emitted independently, $P(X, Y \mid \text{IID})$ can be factorized. Introduce the following quantities,

$$F_\eta^X(i,j) \quad = \quad \eta(1-\eta)^{(j-i+1)} \prod_{k=i}^{j} P^X(a_k), \quad \text{for} \quad \text{i} \leq \text{j}, \tag{2}$$

$$F_\eta^Y(i',j') \quad = \quad \eta(1-\eta)^{(j'-i'+1)} \prod_{k'=i'}^{j'} P^Y(b_{k'}), \quad \text{for} \quad \mathrm{i}' \leq \mathrm{j}'. \tag{3}$$

Use the initialization conditions,

$$\begin{aligned} F_\eta^X(-1,-1) &= \eta, \\ F_\eta^X(i,i-1) &= \eta, \\ F_\eta^X(-1,i) &= F_\eta^X(0,i), \quad 0 \leq i < n. \end{aligned} \tag{4}$$

(5)

Perform a similar construction for F_α^Y, with $\alpha = \eta_L, \eta_R, \eta_J$. In this notation, we have for the IID model,

$$P(X,Y \mid \text{IID}) = F_\eta^X(0,n-1)F_\eta^Y(0,m-1) \equiv F_\eta(\vec{0},(n-1,m-1)). \tag{6}$$

The IID is identical in design to the flanking models (F_L, F_R, and F_J) that we use to generate local alignments in the OTH, COD, and RNA models.

OTH Model Forward algorithm

Using the vectorial notation introduced previously in Section 2.3.3 of the paper, the pair-HMM OTH model Forward algorithm can be cast into the form,

$$\begin{aligned} T_{F_L}(\vec{\imath}) &= F_L(\vec{0},\vec{\imath}); \\ T_{F_J}(\vec{\imath}) &= (1-\eta)\sum_{\vec{k}=\vec{0}}^{\vec{\imath}} E(\vec{k})F_J(\vec{k}+\vec{1},\vec{\imath}); \\ B(\vec{\imath}) &= \xi T_{F_L}(\vec{\imath}) + T_{F_J}(\vec{\imath}); \\ XY(\vec{\imath}) &= P^{XY}(\vec{s}_{\vec{\imath}})\left[(1-2\kappa)B(\vec{\jmath}) + (1-2\delta-\tau)XY(\vec{\jmath}) + \right. \\ &\qquad \left. (1-\epsilon-\tau-\gamma)X(\vec{\jmath}) + (1-\epsilon-\tau-\gamma)Y(\vec{\jmath})\right], \end{aligned}$$

$$
\begin{aligned}
& \text{with} \quad \vec{\jmath} \equiv \vec{\imath} - (1,1); \\
X(\vec{\imath}) &= P^X(s^x_{\vec{\imath}}) \left[\kappa B(\vec{\jmath}) + \delta XY(\vec{\jmath}) + \epsilon X(\vec{\jmath}) + \gamma Y(\vec{\jmath})\right], \\
& \text{with} \quad \vec{\jmath} \equiv \vec{\imath} - (1,0); \\
Y(\vec{\imath}) &= P^Y(s^y_{\vec{\imath}}) \left[\kappa B(\vec{\jmath}) + \delta XY(\vec{\jmath}) + \gamma X(\vec{\jmath}) + \epsilon Y(\vec{\jmath})\right], \\
& \text{with} \quad \vec{\jmath} \equiv \vec{\imath} - (0,1); \\
E(\vec{\imath}) &= \tau \left[XY(\vec{\imath}) + X(\vec{\imath}) + Y(\vec{\imath})\right]; \\
T_{F_R}(\vec{\imath}) &= \sum_{\vec{k}=\vec{0}}^{\vec{\imath}} \left[(1-\xi) T_{F_L}(\vec{k}) + \eta E(\vec{k})\right] \cdot F_R(\vec{k} + \vec{1}, \vec{\imath}).
\end{aligned} \tag{7}
$$

The initialization conditions are

$$
XY(-\vec{1}) = 0, \tag{8}
$$

in addition to

$$
\begin{aligned}
XY(i,-1) = Y(i,-1) = 0, \qquad & 0 \le i < n; \\
XY(-1,i') = X(-1,i') = 0, \qquad & 0 \le i' < m.
\end{aligned} \tag{9}
$$

To obtain the Viterbi algorithm, one just has to replace all sums in (7) with a maximization operation.

The probability that sequences X and Y are related according to the OTH model by *any* local alignment $\overline{XY}$ is

$$
P(X, Y \mid \text{OTH}) = \sum_{\overline{XY}} P(\overline{XY} \mid \text{OTH}) = T_{F_R}(\vec{\imath} = (n,m)), \tag{10}
$$

for a sequence X of length n and a sequence Y of length m.

The log-odds ratios of the OTH model respect to the IID model (1) are given by

$$
LOD(X, Y, \text{OTH}) = \log_2 \frac{P(X, Y \mid \text{OTH})}{P(X, Y \mid \text{IID})}. \tag{11}
$$

This general algorithm has a cost O(nm) in storage and a cost O(n^2m^2) in time, for two sequences of length n and m.

The implemented version of the algorithm usually holds an input alignment fixed, instead of allowing the model to optimally align the input sequences. This simplified version can be obtained from the previous description simply by eliminating the vectorial character of the recursions in (7), by imposing the constraints $i = i'$ and $j = j'$. In its fixed–alignment (diagonal) version the OTH model parsing algorithm becomes cost O(L) in storage and cost O(L^2) in time, for an alignment of length L.

A further simplification of the OTH model that renders its cost linear both in time and memory requires trivializing the pass by the F_J meta-state,

$$F_J(\vec{\imath}, \vec{\imath}) = \eta^2(1-\eta)^2, \qquad F_J(\vec{\imath}, \vec{\jmath}) = 0 \quad \text{if} \quad \vec{\imath} \neq \vec{\jmath}. \tag{12}$$

In this way, no unaligned nucleotide is emitted outside the two flanking ends of the alignment.

Generalization of the OTH model Forward algorithm

In the previous section we have introduced the Forward algorithm for the OTH model assuming that we score both sequences from beginning to end. We need to generalize the algorithm so we can score sequences from a given start position (i_0, i'_0) to an end position (j_0, j'_0) with $0 \leq i_0 \leq j_0 < n$ and $0 \leq i'_0 \leq j'_0 < m$.

We will need this generalization in the algorithms of the COD and RNA models.

The generalization of the forward algorithm in (7) when we score between position

$i_0 \rightarrow i$ in sequence X and positions $i'_0 \rightarrow i'$ in sequence Y is,

$$\begin{aligned} T_{F_L}^{\vec{i_0}}(\vec{\imath}) &= F_L(\vec{i_0}, \vec{\imath}), \\ T_{F_J}^{\vec{i_0}}(\vec{\imath}) &= (1-\eta) \sum_{\vec{k}=\vec{i_0}}^{\vec{\imath}} E^{\vec{i_0}}(\vec{k}) F_J(\vec{k}+\vec{1}, \vec{\imath}), \\ T_{F_R}^{\vec{i_0}}(\vec{\imath}) &= \sum_{\vec{k}=\vec{\imath_0}}^{\vec{\imath}} \left[(1-\xi) T_{F_L}^{\vec{i_0}}(\vec{k}) + \eta E^{\vec{i_0}}(\vec{k}) \right] \cdot F_R(\vec{k}+\vec{1}, \vec{\imath}). \end{aligned} \tag{13}$$

The $F_l, F_R,$ and F_J functions are the ones defined in (6). Formally the recursions for $B^{\vec{i_0}}(\vec{\imath})$, $XY^{\vec{i_0}}(\vec{\imath})$, $X^{\vec{i_0}}(\vec{\imath})$ and $Y^{\vec{i_0}}(\vec{\imath})$ are identical to the ones in (7). However the initialization conditions are a little different

$$XY(\vec{\imath_0} - \vec{1}) = 0, \tag{14}$$

in addition to

$$\begin{aligned} XY(i, i'_0 - 1) = Y(i, i'_0 - 1) = 0, \qquad & i_0 \leq i < j_0; \\ XY(i_0 - 1, i') = X(i_0 - 1, i') = 0, \qquad & i'_0 \leq i' < j'_0. \end{aligned} \tag{15}$$

The probability that sub-sequences $X : (i_0, j_0)$ and $Y : (i'_0, j'_0)$ are related according to the OTH model by *any* local alignment is

$$P(i_0, i'_0 \rightarrow j_0, j'_0 \mid \text{OTH}) = T_{F_R}^{\vec{i_0}}(\vec{\imath} = (j_0, j'_0)), \tag{16}$$

for a sequence X of length n and a sequence Y of length m.

To obtain the Viterbi algorithm, one just has to replace all sums in (13) with a maximization operation.

For a fixed start (i_0, i'_0) and stop positions (j_0, j'_0), the complexity of this generalized algorithm is the same as that described before for the OTH model, simply substituting $n \rightarrow j_0 - i_0 + 1$ and $m \rightarrow j'_0 - i'_0 + 1$.

COD Model Forward algorithm

Let us define the following quantities:

$$\begin{aligned} COB(\vec{\imath}) &= P(0,0 \to i,i' \mid \mathrm{O_B}), \\ COE(\vec{\imath}) &= P(i,i' \to n-1,m-1 \mid \mathrm{O_E}), \\ COJ(\vec{\imath},\vec{\jmath}) &= P(i,i' \to j,j' \mid \mathrm{O_J}). \end{aligned} \tag{17}$$

Where the probabilities are calculated as in equation (16) with the transition probabilities appropriate for each OTH model: O_B, O_E, and O_J.

The Forward algorithm for this COD pair-HMM is,

$$\begin{aligned} T_{O_B}(\vec{\imath}) &= COB(\vec{\imath}), \\ C_E(\vec{\imath}) &= \sum_{\alpha,\beta} \xi_{\alpha,\beta} \cdot P^{\alpha,\beta}(x_{i-\alpha} \cdots x_{i-1}, y_{i'-\beta} \cdots y_{i'-1}) C_B(i-\alpha, i'-\beta), \\ T_{O_J}(\vec{\imath}) &= (1-\eta) \sum_{\vec{k}=\vec{0}}^{\vec{\imath}} C_E(\vec{k}) \cdot COJ(\vec{k}+\vec{1},\vec{\imath}), \\ C_B(\vec{\imath}) &= \varphi T_{O_B}(\vec{\imath}) + T_{O_J}(\vec{\imath}), \\ T_{O_E}(\vec{\imath}) &= \sum_{\vec{k}=\vec{0}}^{\vec{\imath}} \Big[(1-\varphi) T_{O_B}(\vec{k}) + \eta C_E(\vec{k}) \Big] \cdot COE(\vec{k}+\vec{1}). \end{aligned} \tag{18}$$

To obtain the Viterbi algorithm, one just has to substitute all sums in (17) and (18) with a maximization operation.

We use hexamer-dependent statistics for the COD model. An approximation analogous to the one introduced for the emission of two base pairs in the RNA model (equation (6) of paper)) can be used to incorporate dicodon probabilities for the COD model. If we assume that codons A and B (in sequences X and Y respectively) are aligned, and preceded by the aligned codons $\widetilde{A}$, $\widetilde{B}$, we can write

$$P(AB \,|\, \widetilde{A}\widetilde{B}\, t) \simeq P^{COD}(AB \,|\, t)$$

$$\times \frac{1}{2}\left[\frac{P_X^{\mathsf{hex}}(\widetilde{A}A)}{P(\widetilde{A}\,|\,t)P(A\,|\,t)} + \frac{P_Y^{\mathsf{hex}}(\widetilde{B}B)}{P(\widetilde{B}\,|\,t)P(B\,|\,t)}\right], \qquad (19)$$

where $P_X^{\mathsf{hex}}(\widetilde{A}A)$ and $P_Y^{\mathsf{hex}}(\widetilde{B}B)$ are the 64×64 dicodon frequencies for the relevant two species under comparison, which we assume are time independent.

The probability that sequences X and Y are related according to the COD model by *any* local alignment is

$$P(X, Y \mid \mathrm{COD}) = T_{O_E}(n, m), \qquad (20)$$

for a sequence X of length n and a sequence Y of length m. The log-odds ratios of the COD model respect to the IID model given by (1) are

$$LOD(X, Y, \mathrm{COD}) = \log_2 \frac{P(X, Y \mid \mathrm{COD})}{P(X, Y \mid \mathrm{IID})}. \qquad (21)$$

This general algorithm has a cost O(n^2m^2) in storage and a cost O(n^3m^3) in time, for two sequences of length n and m. The fixed–alignment (diagonal) version the COD model parsing algorithm becomes cost O(L^2) in storage and cost O(L^3) in time, for an alignment of length L. The actual implemented version in the programs uses the linear-cost version for the OTH models O_B, O_E, O_J, which reduces the complexity of the implemented COD algorithm to cost O(L^2) in storage and cost O(L^2) in time.

RNA Model Forward/Inside algorithm

Let us define the following quantities,

$$\begin{aligned} ROB(\vec{\imath}) &= P(0,0 \to i, i' \mid \mathrm{O_B}), \\ ROE(\vec{\imath}) &= P(i, i' \to n-1, m-1 \mid \mathrm{O_E}), \\ ROJ(\vec{\imath}, \vec{\jmath}) &= P(i, i' \to j, j' \mid \mathrm{O_J}), \end{aligned} \qquad (22)$$

where the probabilities are calculated as in equation (16) with transition probabilities appropriate for each IND model: O_B, O_E, and O_J.

The algorithm for the RNA model is,

$$\begin{aligned} T_{O_B}(\vec{\imath}) &= ROB(\vec{\imath}), \\ RNA(\vec{\imath}) &= \sum_{\vec{d}=\vec{0}}^{\vec{\imath}} W(\vec{\imath}-\vec{d},\vec{\imath}) \cdot \{\phi T_{O_B}(\vec{\imath}-\vec{d}-\vec{1}) + T_{O_J}(\vec{\imath}-\vec{d}-\vec{1})\}, \\ T_{O_J}(\vec{\imath}) &= (1-\theta)\sum_{\vec{k}=\vec{0}}^{\vec{\imath}} RNA(\vec{k}) \cdot ROJ(\vec{k}+\vec{1},\vec{\imath}), \\ T_{O_E}(\vec{\imath}) &= \sum_{\vec{k}=\vec{0}}^{\vec{\imath}} \Big[(1-\phi)T_{O_B}(\vec{k}) + \theta RNA(\vec{k})\Big] \cdot ROE(\vec{k}+\vec{1}). \end{aligned} \tag{23}$$

To obtain the "best-alignment" algorithm, one just has to substitute all sums in (22) and (23) by a maximization operation.

The probability that sequences X and Y are related according to the RNA model by *any* local alignment is

$$P(X, Y \mid \text{RNA}) = T_{O_E}(n, m), \tag{24}$$

for a sequence X of length n and a sequence Y of length m. The log-odds ratios of the RNA-alignment model respect to the IID model given by (1) are

$$LOD(X, Y, \text{RNA}) = \log_2 \frac{P(X, Y \mid \text{RNA})}{P(X, Y \mid \text{IID})}. \tag{25}$$

The RNA model is an SCFG. The SCFG-like part of the algorithm hides in the calculation of probabilities $W(\vec{\jmath}, \vec{d})$. That is, the probability of having a RNA structure between positions $\vec{\imath} = \vec{\jmath} - \vec{d}$ and $\vec{\jmath}$. The recursions involved in the calculation of the Inside algorithm for $W(\vec{\jmath}, \vec{d})$ are,

$$
\begin{aligned}
W(\vec{\jmath},\vec{d}) &= t_W^L \cdot \sum_{\vec{e}} P^{RNA}(\vec{e} * \vec{s}_{\vec{\imath}}) \cdot W(\vec{\jmath},\vec{d}-\vec{e}) \\
&+ t_W^R \cdot \sum_{\vec{e}} P^{RNA}(\vec{e} * \vec{s}_{\vec{\jmath}}) \cdot W(\vec{\jmath}-\vec{e},\vec{d}-\vec{e}) \\
&+ t_W^P \cdot \sum_{\vec{e}_1,\vec{e}_2} P^{RNA}(\vec{e}_1 * \vec{s}_{\vec{\imath}}, \vec{e}_2 * \vec{s}_{\vec{\jmath}}) \cdot V(\vec{\jmath}-\vec{e}_2,\vec{d}-\vec{e}_1-\vec{e}_2) \quad (26) \\
&+ t_W^{bif} \cdot \sum_{\vec{d}_1} W(\vec{\imath}+\vec{d}_1,\vec{d}_1) \cdot W(\vec{\jmath},\vec{d}-\vec{d}_1-1),
\end{aligned}
$$

for $\quad 1 \le j \le m, \quad 1 \le j' \le n, \quad 0 \le d \le j, \quad 0 \le d_1 \le d, \quad 0 \le d_1 + d_2 \le d,$

and similarly for primed d's. Here m and n are the respective lengths of the two sequences to be aligned. The vector $\vec{e}$ can take three different values: $(1,1)$, $(1,0)$ and $(0,1)$ that correspond to moving one position in at least one of the two sequences. The star product defined as $\vec{e} * \vec{s} \equiv (e_x s_x, e_y s_y)$ produces a "gap"—a zero instead of a nucleotide—when there is no movement for one of the sequences.

The symbol V again denotes the state we are in after emitting one pair in each sequence. The recursion for state V is,

$$
\begin{aligned}
V(\vec{\jmath},\vec{d}) &= t_V^{IS1} \cdot IS1(\vec{\jmath},\vec{d}) \\
&+ t_V^{IS2} \cdot \sum_{\vec{d}_1,\vec{d}_2} \sum_{\vec{e}_1,\vec{e}_2} \Big[IS2(\vec{\imath}+\vec{d}_1-1,\vec{d}_1-1;\vec{\jmath}-\vec{d}_2+1,\vec{d}_2-1) \cdot \\
&\qquad P^{RNA}(\vec{e}_1 * \vec{s}_{\vec{\imath}+\vec{d}_1}, \vec{e}_2 * \vec{s}_{\vec{\jmath}-\vec{d}_2}) V(\vec{\jmath}-\vec{d}_2-\vec{e}_2,\vec{d}-\vec{d}_1-\vec{d}_2-\vec{e}_1-\vec{e}_2) \Big] \\
&+ t_V^{bif} \cdot \sum_{\vec{d}_1} W_B(\vec{\imath}+\vec{d}_1,\vec{d}_1) \cdot W_B(\vec{\jmath},\vec{d}-\vec{d}_1-1), \qquad (27)
\end{aligned}
$$

for $\quad 1 \le j \le m, \quad 1 \le j' \le n, \quad 0 \le d \le j, \quad 0 \le d_1 \le d, \quad 0 \le d_1 + d_2 \le d,$

and similarly for primed d's.

The recursions for the states that take care of length distributions for hairpin loops

(IS1) and stems, bulges and internal loops (IS2) are,

$$\begin{aligned} IS1(\vec{j},\vec{d}) &= t_{IS1}^{\vec{d}+1}\cdot G^{IS1}(\vec{j},\vec{d}), \\ IS2(\vec{j}_1,\vec{d}_1;\vec{j}_2,\vec{d}_2) &= t_{IS2}^{\vec{d}_1+\vec{d}_2+2}\cdot G^{IS2}(\vec{j}_1,\vec{d}_1)\cdot G^{IS2}(\vec{j}_2,\vec{d}_2), \\ IS2(0) &= t_{IS2}^{(0,0)}, \end{aligned} \tag{28}$$

where $0 \leq d, d', d_1, d_1', d_2, d_2' \leq maxloop-1$. Here the functions G^{ISx} (for $x = 1, 2$) are given by the general expression,

$$\begin{aligned} G^{ISx}(\vec{j},\vec{d}) = \sum_k \sum_{\vec{e}_1,\ldots,\vec{e}_k} & P^{ISx}(\vec{e}_1 * \vec{s}_{\vec{j}})P^{ISx}(\vec{e}_2 * \vec{s}_{\vec{j}-\vec{e}_1}) \\ & P^{ISx}(\vec{e}_3 * \vec{s}_{\vec{j}-\vec{e}_1-\vec{e}_2})\ldots P^{ISx}(\vec{e}_k * \vec{s}_{\vec{j}-\sum_{s=1}^{k-1}\vec{e}_s}). \end{aligned} \tag{29}$$

The number of additive terms in this function reflects the number of possible alignments, and it verifies the condition $\sum_{s=1}^{k}\vec{e}_s = \vec{d}+1$. The initialization conditions are $s^x_{(-1,j)}$ =“gap”, and $s^y_{(j,-1)}$ =“gap”.

For the transition probabilities t^{trans}_{state}, we have

$$\sum_{trans \in state} t^{trans}_{state} = 1, \quad \forall \quad \text{state}. \tag{30}$$

For the particular grammar at hand, the previous conditions translate to

$$\begin{aligned} t^L_W + t^R_W + t^P_W + t^{bif}_W &= 1, \\ t^L_{W_B} + t^R_{W_B} + t^P_{W_B} + t^{bif}_{W_B} &= 1, \\ t^{IS1}_V + t^{IS2}_V + t^{bif}_V &= 1, \\ \sum_{l,l'=1}^{maxloop} t^{(l,l')}_{IS1} &= 1, \\ \sum_{l,l'=0}^{maxloop} t^{(l,l')}_{IS2} &= 1. \end{aligned} \tag{31}$$

These transition probabilities are calculated using a training set of RNA motifs that includes rRNAs and tRNAs [26, 27]. $P^{IS1}(\vec{s_i})$ and $P^{IS2}(\vec{s_i})$ are the mutation probabilities in hairpin loops and internal loops respectively [which we set to be equal to the OTH model probabilities defined in equaiton (4) of the paper].

Notice that the RNA model can be aligned by either Forward or Viterbi with respect to the HMM part of the model; however, we always use the Inside algorithm to evaluate the SCFG part of the pair grammar. This is necessary to assure that the RNA score is comparable to the COD and OTH scores (by removing the conditioning on one particular structure out of the combinatorially enormous number of possible structures), and also to avoid undesirable effects associated with choosing a best path when we have a grammar with ambiguity (R. Giegerich, personal communication).

This is the algorithm used when we allow the model to generate its own alignments. This general algorithm has a cost O(n^2m^2) in storage and a cost O(n^3m^3) in time, for two sequences of length n and m. For the scoring system in which a predetermined alignment is scored locally, since we are not adding additional gaps in the pairwise alignment, the algorithm loses its vectorial character ($j = j'$, $d = d'$), the vector $\vec{e}$ takes always the value $(1, 1)$, and the memory and time complexity of the algorithm reduces to O(n^2) storage and O(n^3) time.