The locality dilemma of Sankoff-like RNA alignments

—

Supplementary material

Teresa Müller, Milad Miladi, Frank Hutter,
Ivo Hofacker, Sebastian Will, and Rolf Backofen,

Contents

1 RNA sequence structure alignment

Alignments of RNA sequences

Global RNA alignments reflect the evolutionary relationship between RNA sequences via point mutations, insertions and deletions. Whereas RNA sequences are strings composed of the nucleotide symbols A,C,G, and U, an alignment of the two RNAs consists of two alignment strings over A,C,G,U, and the gap

symbol '-', such that both strings have the same length and each alignment string contains all the nucleotides of the respective sequence in the sequence order. The gap symbols indicate insertions and deletions of the nucleotide in the opposite sequence. RNA alignment algorithms, like LocARNA, typically compute the best alignment of two RNAs according to an alignment score, which combines a sequence and a structure component. In this additive composition, the influence of the structure component is controlled by a factor, here called the *structure weight*. A *local alignment* compares subsequences of two input sequences. Thus, local alignment algorithms (like LocARNA in local alignment mode) compute the best local alignment of two RNAs, which is defined as the best global alignment of some subsequences of the input RNAs, according to the alignment score.

Sankoff

Sankoff is an algorithm proposed in 1985, which does sequence- and structure alignment simultaneously [5]. The basic idea was to combine the classical dynamic programming sequence alignment with the structure prediction proposed by Zuker. The objective function given in Equation 1 shows that the minimum free energy for the structure P_a of sequence a, (minimum) free energy (FE) for the structure P_b of sequence b and the edit distance of the alignment A are optimized together. Therefore the objective function is optimized by finding the best combination of the equivalent structures and a constrained alignment. Following the concept of aligning shapes and not the whole sequence and by allowing 2-loops to be aligned either to another 2-loops or to gaps makes the Sankoff algorithm more flexible in respect to structure prediction.

$$FE(a, P_a) + FE(b, P_b) + EditDistance(A) \tag{1}$$

Sankoff-like algorithms

To achieve an optimal alignment, several algorithms were proposed providing faster approaches. One successfully applied heuristic of Sankoff is using precomputed base pair probabilities, calculated by McCaskill, instead of computing energies using the Zuker method. Using precomputed McCaskill base pair probability matrices enables a faster scoring of the consensus structure for the alignment A than it was done before.

For a pairwise sequence alignment of sequences a and b the base pair probabilities will be stored in a two-dimensional matrices. Algorithms that use this kind of technical progress are call PMcomp like algorithms. LocARNA is one of the PMcomp successors.

Comparison of Sankoff and LocARNA

Before comparing the two scoring functions it is important to note that the Sankoff algorithm computes a global alignment using energies and distances.

To compute a local alignment the distances would need to be transferred into similarities. Also the negative energies would need to be converted so that they can maximize the objective function. LocARNA uses base pair probabilities to compute the *structure score component*, which already fits to the local alignment scheme of a objective function. A possibility on how to use the Zuker algorithm for local alignment is given in the Foldalign tool, where the energy model is multiplied by $-10 mol/kcal$. This score transformations enables alignment score maximization [3]. Notice that the *sequence score component* of LocARNA could be negative, however using the base pair cutoff probability parameter in practice, this will not happen. Therefore a structure contribution to the score will be always positive.

Sankoff et al. already discussed that there is a trade-off between the free energy (structure contribution) and the alignment cost. This trade-off should be balanced using weights. However Sankoff's algorithm used an unweighted objective function, since alignment cost is in terms of arbitrary units x and y. But it was also stated that in practice the units should be calibrated using known secondary structures. LocARNA already weights the structural alignment by the structure weight parameter ω. LocARNA is more strict in including *structure components* into the alignment, because every base pair (i, j) in sequence a needs to be aligned to a base pair (k, l) in sequence b. In the Sankoff's algorithm not every base pair needs to be aligned to another one.

LocARNA recursions

LocARNA uses a modified dynamic programming scheme which takes advantage of the fact that the base pair probability matrices Pr^a and Pr^b are commonly sparse. The calculation of the alignment is based on the recursion function M. The first case of the recursion scores a (mis-)match, the second and third case penalizes an insertion or deletion and the last case evaluates if there are further structural elements and tries to find the maximum by calculating $D(j' j; l' l)$ for each base pair combination. The evaluation is only done for $Pr^a_{j',j} \geq p_*$ and $Pr^b_{l',l} \geq p_*$ which makes the calculation computationally less expensive. The maximum similarity score of A for subsequences $A[i...j]$ and $B[k...l]$ and the consensus secondary structure of $(ij; kl)$ is obtained from the scoring function $D(ij; kl)$ (equation: 4). The function only includes significant base pairs because of the structural pre-computation. To avoid redundant computation i and k are fixed while j and l are varying which gets denoted as $D_{i*;k*}$. In the end the recursion can be evaluated in $O(n^2)$ memory and $O(n^4)$ time, with the optimal global alignment score found at position $M_{0|A|;0|B|}$.

$$M_{ii-1;kk-1} = 0 \tag{2}$$

$$M_{ij;kl} = max \begin{cases} M_{ij-1;kl-1} + \sigma(j,l) \\ M_{ij-1;kl} + \gamma \\ M_{ij;kl-1} + \gamma \\ max_{j'l'} M_{ij'-1;kl'-1} + D_{j'j;l'l} \end{cases} \tag{3}$$

$$D_{ij;kl} = M_{i+1j+1;k-1l-1} + \omega(\Psi^A_{ij} + \Psi^B_{kl}) + \tau\sigma'(ij;kl) \tag{4}$$

Local alignment: LocARNA uses the same tricks as in the Smith-Waterman algorithm by adding an additional zero entry to the recursion function M which restricts to all prefixes of the sequence e.g. ($M_{0j;0l}$). This way the optimal local alignment score is located in $max_{jl}(M_{0j;0l})$ and the secondary structure can be obtained by a traceback in $O(n^2)$ time.

$$M_{0j;0l} = max \begin{cases} 0 \\ M_{0j-1;0l-1} + \sigma(j,l) \\ M_{0j-1;0l} + \gamma \\ M_{0j;0l-1} + \gamma \\ max_{j'l'} M_{0j'-1;0l'-1} + D_{j'j;l'l} \end{cases} \tag{5}$$

$$D_{ij;kl} = M_{i+1j+1;k-1l-1} + \omega(\Psi^A_{ij} + \Psi^B_{kl}) + \tau\sigma'(ij;kl) \tag{6}$$

Extending the LocARNA score by a position-wise penalty

The novel proposed position-wise penalty (λ) is an alignment length depended score penalization. For every position in the local alignment the position-wise penalty value will be subtracted form the score, which leads to a penalization of the length of the alignment.

$$\begin{aligned} &\sum_{(ij;kl)\in S} \big(\omega(\Psi^a_{ij} + \Psi^b_{kl}) + \tau\sigma'(i,j,k,l) - 4\lambda\big) \\ &+ \sum_{(i,k)\in A_s} (\sigma(i,k) - 2\lambda) - N_{gap}(\gamma + \lambda) - N^o_{gap}\beta \end{aligned} \tag{7}$$

Function 7 displays the extended objective function of LocARNA with the position-wise penalty λ. For each structure edge four positions i,j,k and l are scored. Therefore 4 times the λ value need to be reduced. For each sequence alignment edge, i.e. pairs (i,j) within A_s, 2λ value is subtracted for matches and mismatches and λ is reduced in the case of insertion or deletion.

2 Alignment quality measures

In the following subsections alignment quality measurements for pairwise alignments applied in this work are described in more detail.

For a pairwise local alignment of two random sequences we define the alignment length or expected length as the sum of the two aligned subsequences.

Sum-of-Pairs Score (SPS)

The *Sum-of-Pairs Score (SPS)* is a global sequence alignment quality measurement for multiple sequence alignments [7]. The alignment quality can be measured by comparing to a reference alignment, which is considered as ground truth. In this study we restrict our benchmark to pairwise sequence alignments. For the purpose of assessing pairwise alignments, the *SPS* computes the fraction of the reference alignment columns that are as well present in the evaluated alignment (i.e. the correctly predicted columns).

$$\text{SPS} = \frac{\#\text{ correctly predicted columns}}{|\text{reference}|} \tag{8}$$

Consequently, a perfect prediction has a SPS of one, whereas a SPS of zero is assigned to a completely mispredicted alignment.

Local alignment quality *maxSPS*

The SPS score can not thoroughly evaluate the quality of local alignments, because it only validates how many alignment edges (equivalent to alignment columns) are correctly predicted. A comprehensive local alignment metric needs to consider the undesired extension of the alignment into the context surrounding the local motif. To make the score applicable for local alignments we changed the divisor. The extension of the alignment into the context is not preferred and therefore is penalized by having a greater divisor. If the length of the predicted alignment is longer than the reference alignment the divisor will be the length of the predicted alignment.

The *maxSPS*, which measures the local alignment quality of a local alignment 'prediction' compared to a 'reference', is defined as

$$\text{maxSPS} = \frac{\#\text{ correctly predicted edges}}{\max(|\text{reference}|, |\text{prediction}|)}, \tag{9}$$

where |reference| and |prediction| denote the respective lengths of the reference and the predicted local alignment. Note that the alignment quality will be again between 0 and 1. The maxSPS can be as good as the SPS would be for a local alignment, or the value will be lower. For the same alignment the maxSPS will never be higher than the SPS value.

Supplementary Figure S1: **maxSPS example.** First a reference alignment is given. The two predicted alignment are based on different parameter settings. True alignment edges, edges within the reference, are highlighted in green. Edges that are predicted but not part of the reference alignment are highlighted in red. The refSPS divides the correctly predicted edges by the number of alignment edges of the reference alignment (6). The divisor of the maxSPS will be the maximum of either all alignment edges of the reference or of the predicted alignment.

Matthews correlation coefficient(MCC) for computing structure prediction quality

Matthews correlation coefficient(MCC) evaluates the correctness of the alignment structure prediction. The MCC compares each base pair of the predicted structure alignment P_{pred} with the structure of the reference alignment P_{ref}. If the length of the sequence is of size N, at most $N/2$ base pairs can exists. A base pair of the predicted alignment is denoted by $(i, j) \in P_{pred}$ and of the reference alignment by $(i_0, j_0) \in P_{ref}$. For the entire structure the numbers of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) can be calculated as following [1]:

- TP = the number of times $(i, j) \in P_{pred}$ and $(i_0, j_0) \in P_{ref}$
- TN = the number of times $(i, j) \notin P_{pred}$ and $(i_0, j_0) \notin P_{ref}$
- FN = the number of times $(i, j) \notin P_{pred}$ and $(i_0, j_0) \in P_{ref}$
- FP = the number of times $(i, j) \in P_{pred}$ and $(i_0, j_0) \notin P_{ref}$

The correlation coefficient is always between -1 and $+1$. A value of $+1$ indicates a total agreement of the data, meaning all the structural elements of the prediction and reference are the same. In contrast to this a value of -1 would indicate total disagreement (no base pair of P_{pred} was predicted correctly). If the MCC is close to 0 the predicted alignment by the tool is not better than a random alignment.

$$C(D, M) = \frac{TP * TN - FP * FN}{\sqrt{(TP + FN)(TP + FP)(TN + FP)(TN + FN)}},$$

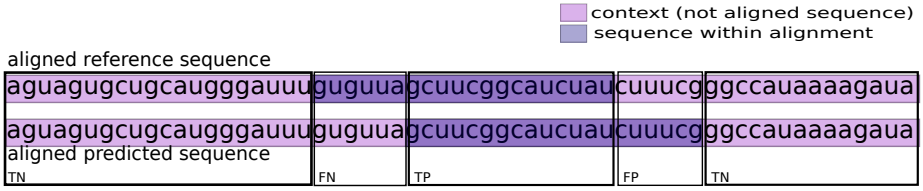
The correlation coefficient uses all four values (TP, TN, FP, FN) and therefore provides a better combined evaluation than just the sensitivity or specificity.

Quality of alignment boundaries

Validating the alignment quality can be based on measuring how many nucleotides of the local motif are predicted and how many nucleotides of the context are not part of the predicted alignment. This allows us to introduce *sensitivity*, to measure how well the structured RNA motif is found, and *specificity*, which assesses how well the alignment avoids extensions into the context.

True positive TP or true negative TN values are all nucleotides that are part or not part of the predicted and the reference alignment. False positive FP values are all nucleotides that are part of the predicted but not of the reference alignment and false negative FN values are all nucleotides which are not part of the predicted but part of the reference alignment.

Thus, we compute the sensitivity by dividing the amount of correctly predicted nucleotides (TP) by the length of the local motif (TP + FN). This will now measure how much of the local motif (or ncRNA) area is covered by the



Supplementary Figure S2: **Example of predictive and actual classes.** Each sequence input to a local alignment can be considered as two parts: the context (light violet) and the sequence local motif or predicted alignment (dark violet). Shown here is a sequence with its alignment information of a reference alignment and of the predicted alignment. The statistical measurements needed for computation of the sensitivity and specificity are calculated by combining the position information within the two rows. Therefore all nucleotides which are not aligned in the reference and predicted alignment are counted as true negatives (TN). The nucleotides which are part of the reference alignment but not of the predicted alignment are counted as false negatives (FN). All nucleotides aligned in the predicted and reference alignments are counted as true positives (TP). And all nucleotides not part of the reference but part of the predicted alignment of counted as false positives (FP).

alignment. A sensitivity of one would mean that the complete motif is covered by the alignment prediction. The sensitivity is a therefore a appropriate measure for how good the local motif (or ncRNA) motif is predicted by an algorithm.

$$\text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

To evaluate the specificity, we divide the number of nucleotides that are correctly not aligned (TN) by the size of the complete context (TN + FP)

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}$$

The specificity is therefore a appropriate measure to detect undesired alignment extension.

The F1 measure combines the sensitivity and precision and is therefore a combination on how well the reference alignment is found and how accurate the predicted area is.

$$F1 = TN/(1/sensitivity + 1/precision)$$

3 SMAC

Optimization of parameters is a long-standing research area in machine learning (ML). Simple approaches rely on grid-search, however, more recent ML-based

approaches use the program as a blackbox and iteratively run the program on different parameter settings. Sequential model-based optimization (SMBO) [6] uses the data from these runs as a training-set and try to learn the parameter-score relationship.

To optimize four crucial parameter of LocARNAs objective function Sequential Model-based Algorithm Configuration (SMAC)[4], a blackbox SMBO method, is used. This method can be applied to any algorithm, where a parameter configuration space, a set of instances and a cost metric or score can be defined.

SMAC, like most SMBO methods, is divided into 3 parts. (1) The *fit model* function, where already computed data is used to build a model. In our case the model is a random forest. (2) The *select configuration* function where the model is used to traverse the parameter configuration space and propose a new parameter settings for further investigation and validation. (3) At the *intensify* function the proposed or new parameter configurations are applied to the objective function of the algorithm (LocARNA) and its performance is compared against the best parameter configuration seen so far. This computations are used in the next round to build a more precise model. This iteration is repeated until a stop criterion is reached.

In the *intensify phase* the actual cost-function is computed, meaning here is the connection between the black-box learner and the algorithm, whose parameter should be optimized. The cost-function, to which LocARNA parameters are optimized in the global alignment setting is the geometric mean of SPS and MCC. Since we can give only one score to optimize the parameters of our alignment approach, we cannot use sensitivity and specificity to score the local alignments of our benchmark set. An optimization of the sensitivity and specificity would help to tweak the parameters to find the correct alignment boundary edges.

Instead, we used the maxSPS-value, a novel local alignment quality score, to score an resulting alignment. The maxSPS value implicitly scores with the correct alignment edges also the sensitivity and specificity of the found regions for local alignment. The maxSPS value implicitly scores the correct alignment edges, and is therefore similarly to the sensitivity and specificity information of their found regions for local alignment.

4 Calling LocARNA and reproducing results

In this publication we use LocARNA v2.0.0RC6 which can be found here: https://github.com/s-will/LocARNA/releases/tag/v2.0.0RC6. Since the parameter optimization was performed with LocARNA v1.9., we set the default parameter of LocARNA v2.0.0RC6 to the one of LocARNA v1.9. For runtime purposes we set the following parameters: `--max-diff-am -1 --min-prob 0.0005`. Further more to get a more comprehensive output from LocARNA the following parameters were set for each call: `--moreverbose -v`

We performed several parameter optimizations which resulted in different

parameter sets. They are summarized in Table 1 of the main text. We describe the resulting parameter sets as follows:

- Default parameters of LocARNA version 1.9. (Default values). Can be found in line 1 of Table 1.
- Optimized parameters of the LocARNAs global alignment using position-wise penalty of 0 (Global optimized). Can be found in line 2 of Table 1.
- Optimized parameters of LocARNAs local alignment using position-wise penalty of 0 (Local optimized with no penalty ($\lambda = 0$))Can be found in line 3 of Table 1.
- Optimized parameters of LocARNAs local alignment using position-wise penalty of 15 (Local optimized with penalty ($\lambda = 15$)) Can be found in line 3 of Table 1.

The two local alignment optimized parameter sets are not used in any analysis displayed in the figures of the main text. However, they proved that the position-wise penalty can compensate the structure contribution of the score and enable the use of the same parameter setting in local and global mode. In the following we will described different LocARNA calls used in this publication:

global mode LocARNA call using Default values

```
mlocarna --max-diff-am -1 --min-prob 0.0005
  --indel -350 --indel-opening -500 --struct-weight 200 --tau 0
  --write-structure 200 --pw-aligner-options "--pos-output
  --penalized=15 --sequ-local=off" input.fa --moreverbose -v
```

local mode LocARNA call using Default values

```
mlocarna --max-diff-am -1 --min-prob 0.0005
  --indel -350 --indel-opening -500 --struct-weight 200 --tau 0
  --pw-aligner-options "--sequ-local on --penalize 0
  --pos-output" --local-progressive input.fa --moreverbose -v
```

globel mode LocARNA call using Global optimized parameters

```
mlocarna --max-diff-am -1 --min-prob 0.0005
  --indel -68 --indel-opening -807 --struct-weight 200 --tau 72
  --write-structure 200 --pw-aligner-options "--pos-output
  --penalized=0 --sequ-local=off" input.fa --moreverbose -v
```

local mode LocARNA call using Local optimized with no penalty ($\lambda = 0$) parameters

```
mlocarna --max-diff-am -1 --min-prob 0.0005
--indel -136 --indel-opening -975 --struct-weight 115 --tau 38
```

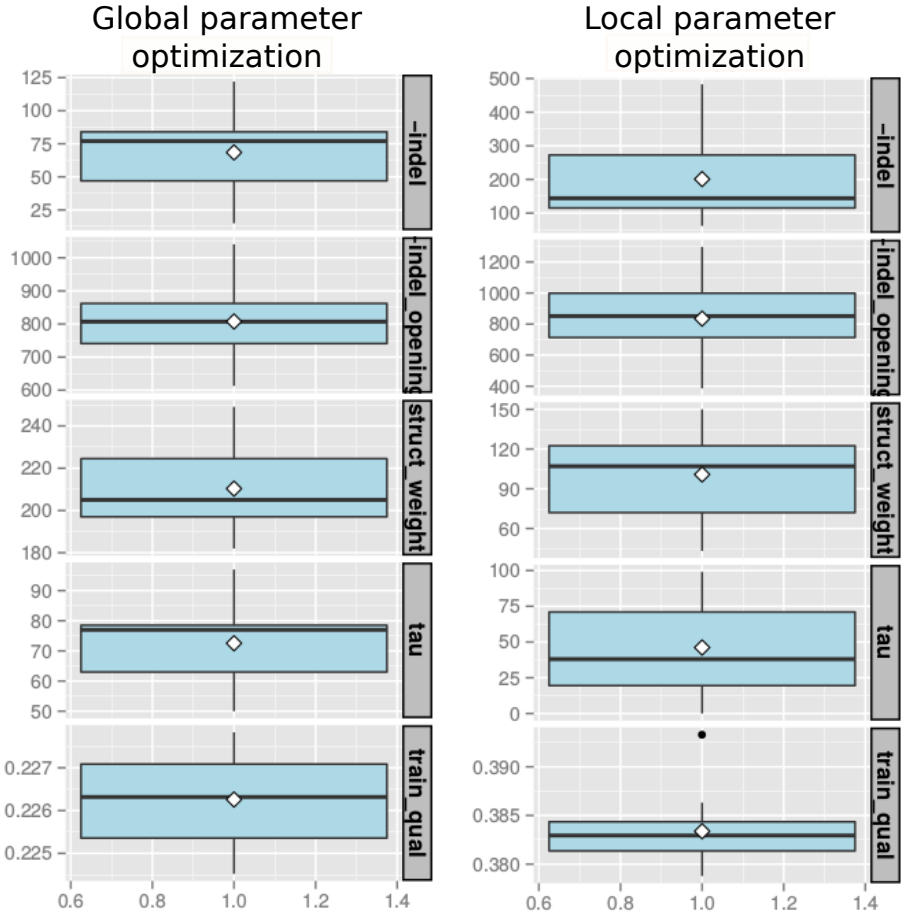
```
  --pw-aligner-options "--sequ-local on --penalize 0
  --pos-output" --local-progressive input.fa --moreverbose -v
```

local mode LocARNA call using Global optimized parameters and position-wise penalty 15

```
mlocarna --max-diff-am -1 --min-prob 0.0005
  --indel -68 --indel-opening -807 --struct-weight 200 --tau 72
  --pw-aligner-options "--sequ-local on --penalize 15
  --pos-output" --local-progressive input.fa --moreverbose -v
```

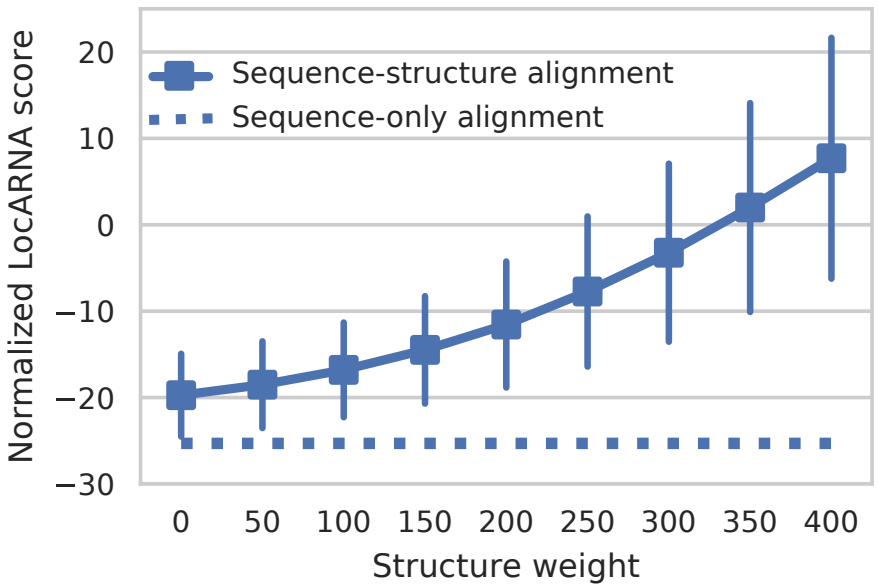
5 Supplementary Figures

5.1 Parameter optimization results

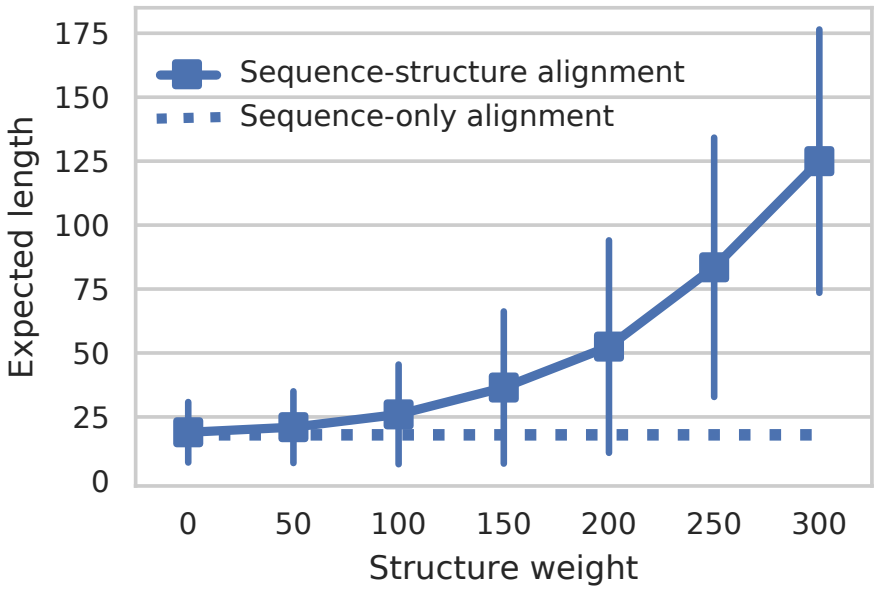


Supplementary Figure S3: **Global and local optimization results.** Parameter distribution for 15 different optimization runs. The optimized parameters are indel or gap-scoring, indel opening or gap opening, structure weight and the tau factor. The reported training quality for the global alignment is base on the square root of SPS times MCC and for the local alignment the quality function is based on the maxSPS. Since the parameters are interacting the effect of there changes can vary. However, compared to the default parameters the optimized parameter show clear trends how parameters should be set to reach more accurate results.

5.2 Score and alignment length results

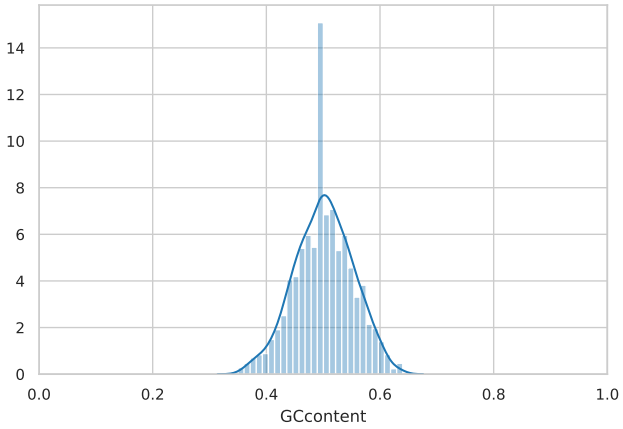


Supplementary Figure S4: **Normalized alignment growth of artificial data.** Normalized alignment growth with increasing structure weight. Using a artificially generated data set (box) indicates that compared to a sequence only score (dashed line) the normalized alignment score increases with increasing structure weight.

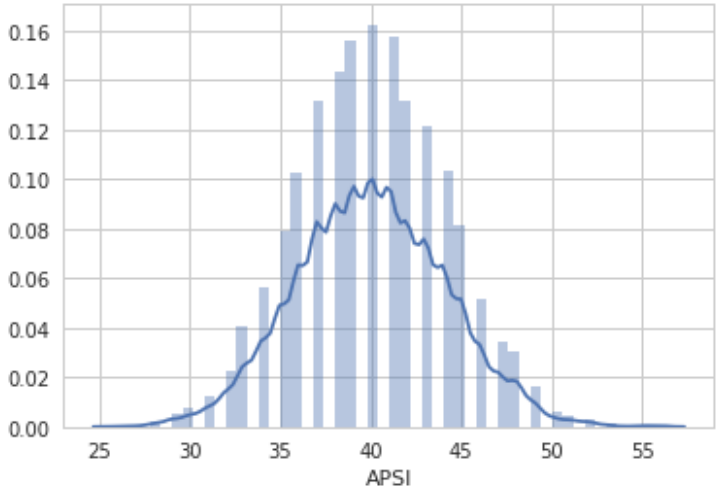


Supplementary Figure S5: **Expected alignment length of random ncRNAs.** The expected length of shuffled ncRNAs (box) for Sequence-structure and sequence-only alignments (dashed line). The shuffled BRAliBase data set also indicates that with increasing structure weight the expected alignment length grows.

5.3 Artificial data set distributions

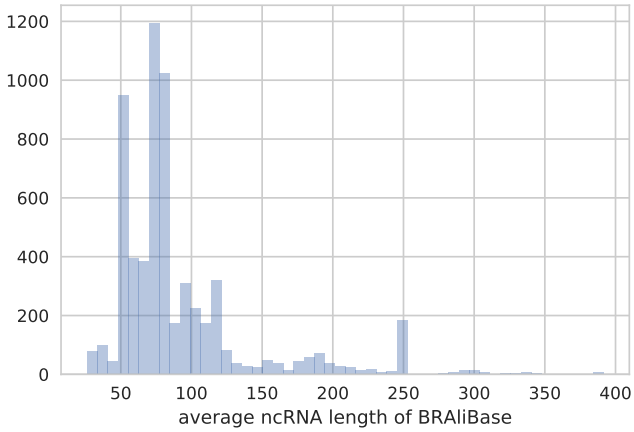


Supplementary Figure S6: **GC-content of artificial data set.** The GC-content distribution of the artificially generated data set (methods). Over the complete artificial data set the GC-content is equally distributed.



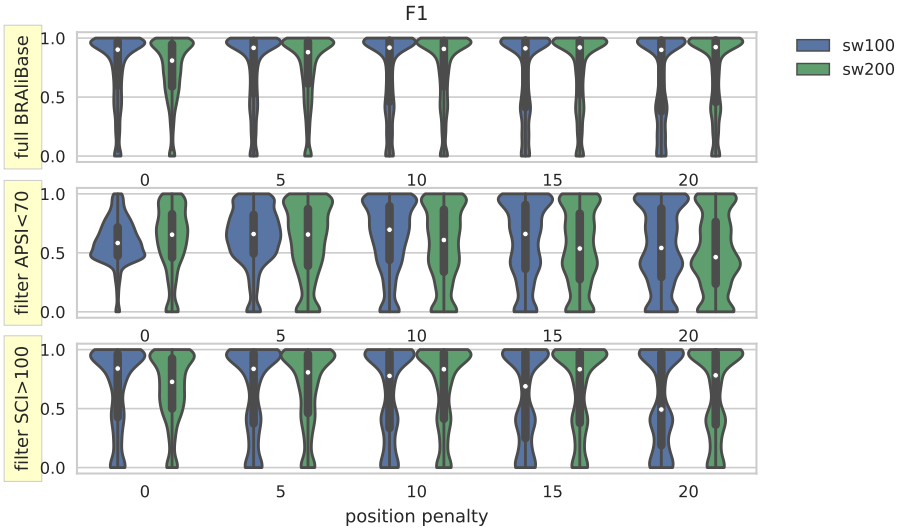
Supplementary Figure S7: **APSI of artificial data set.** The APSI distribution of the artificially generated data set (methods). The APSI is calculated using Alistat. The average APSI distribution is around 40%. Since the data set is artificially curated, it is expected to have a rather low APSI.

5.4 BRAliBase length distributions



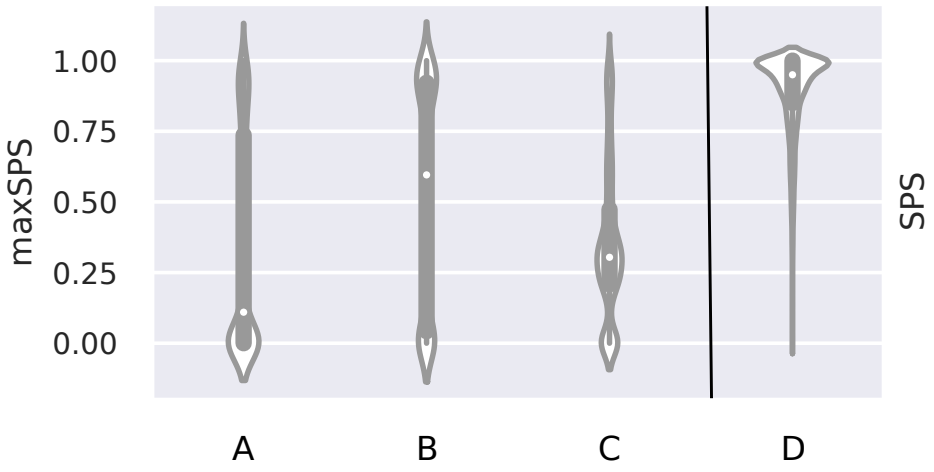
Supplementary Figure S8: **BRAliBase length distribution.** The average sequence length distribution of the two ncRNAs input sequences of all BRAliBase-K2 instances. Length normalization is done by dividing the summed up length by two. Most of the sequences are not longer then 100. There are just a view sequences that are longer than 250 nt.

5.5 Position-wise penalty F1 comparison



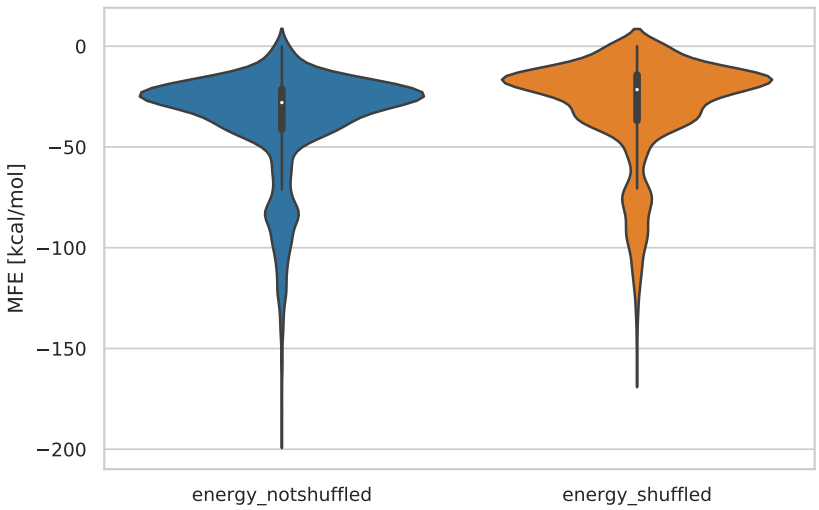
Supplementary Figure S9: **F1 measure comparison.** Alignment quality measured by the F1 value for different penalties and structure weights 100 and 200. The LocalBRAliBase is filtered once for alignments with a APSI lower than 70 and second for alignments with a SCI higher then 100 (methods). k2 LocalBRAliBase: comparing results of structure weight (SW) 100 to a combination of SW 200 and position-wise penalty 10 to 15 achieves similar results. Filtered APSI smaller 70: a low position-wise penalty helps. In contrast high penalties make the results worse. Filtered SCI higher 100: for highly structured sequences the combination of SW 200 and position-wise penalty helps.

5.6 Default and optimized maxSPS

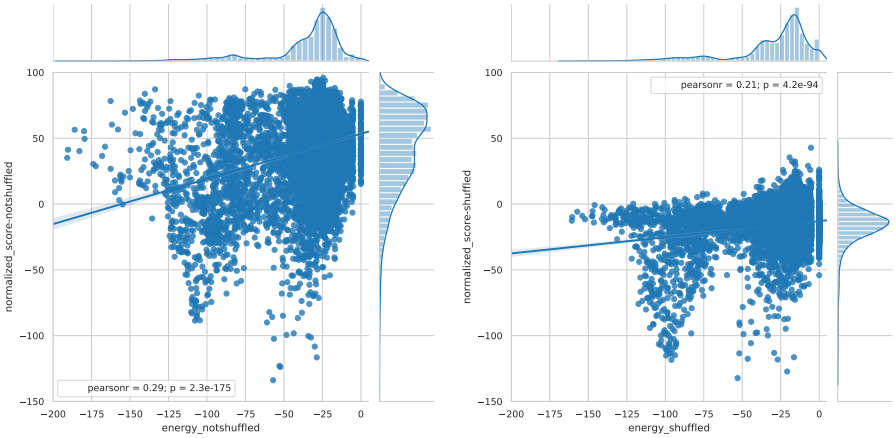


Supplementary Figure S10: **Alignment quality and upper maxSPS boundary.** Alignment quality comparison for different LocARNA settings: (A) local alignment calculations using the default parameters; (B) local alignment calculations using the optimized parameters; (D) global alignment calculations using the optimized parameters. The optimized parameters are displayed in table 1 of the main text. The setting used to generate (D) displays an upper bound for the alignment score that could be reached once the correct alignment boundaries are found. (C) Foldalign (version 2.1.0) [2] results. Foldalign is a another Sankoff-like alignment method applying mechanisms to improve the local alignment. The comparison between the two methods shows that Foldalign (C) performs better than LocARNA with default parameters (A), but it still faces the addressed issues of local Sankoff-like alignment. Therefore, LocARNA using the suggested parameter setting (B) has an improved alignment quality over Foldalign.

5.7 Minimum Free Energy (MFE) results

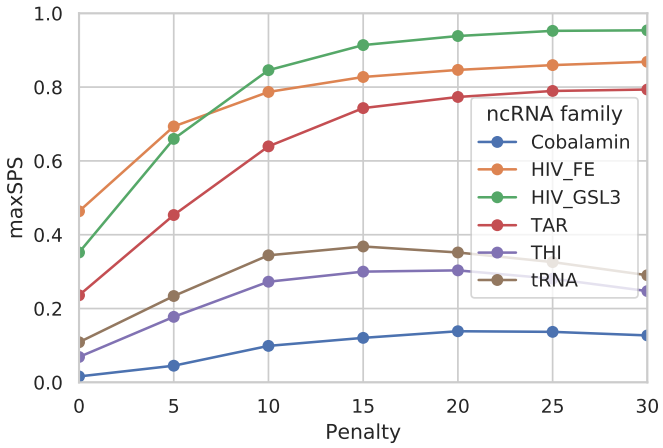


Supplementary Figure S11: **MFE difference of ncRNAs vs. random sequences.** The Minimum Free Energy (MFE) distribution of ncRNAs in BRAliBase data set versus the di-nucleotide shuffled sequences. ncRNAs have lower MFEs than the shuffled sequences.



Supplementary Figure S12: **Normalized score vs. MFE correlation.** The distribution of LocARNA normalized pairwise alignment scores versus the average Minimum Free Energy (MFE) of the two input sequences. (Left: ncRNA sequences from BRAliBase, Right: the shuffled sequences)

5.8 ncRNA family alignment quality



Supplementary Figure S13: **ncRNA family alignment quality.** The distribution of LocARNA alignment quality (average of maxSPS values) of sixth ncRNA families for different position-wise penalties. From the LocalBRAliBase k2 ncRNA family are selected by the number of instances within the LocalBRAliBase. The displayed families are the once with the most instances: Cobalamin (188 instances); HIV_FE (704 instances); HIV_GSL3 (754 instances); TAR (276 instances); THI (318 instances); tRNA (1573 instances). For the maxSPS calculation the global optimized parameters including structure weight 200 where used. The full LocalBRAliBase k2 analysis is shown in figure 5 of the main text.

5.9 Comparison of default and suggested parameter settings by an example

Supplementary Figure S14: **Example of the local alignment performance for detecting structure motif boundaries** This is shown for one instance of our benchmark set LocalBRAliBase from HIV_FE of sequence AJ405950.1_1-52 and AY519071.1_1-252, where the ncRNAs (red boxes) are embedded in their genomic context. We show the local alignment using our optimized parameters (upper alignment) in comparison to the local alignment with the non-optimized default parameters (lower alignment). Using the default parameters, the identified regions in the local alignment extend to the end of the input sequence. Only the optimized parameters allow to identify the regions of the RNAs. Note that for visualization purposes, we don't show the complete benchmark sequences.

References

[1] Jan Gorodkin, Shawn L Stricklin, and Gary D Stormo. Discovering common stem–loop motifs in unaligned rna sequences. *Nucleic Acids Research*, 29(10):2135–2144, 2001.

[2] Jakob H Havgaard, Elfar Torarinsson, and Jan Gorodkin. Fast pairwise structural rna alignments by pruning of the dynamical programming matrix. *PLOS computational biology*, 3(10), 2007.

[3] Jakob Hull Havgaard, Rune B Lyngsø, Gary D Stormo, and Jan Gorodkin. Pairwise local structural alignment of rna sequences with sequence similarity less than 40%. *Bioinformatics*, 21(9):1815–1824, 2005.

[4] Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pp. 507–523. Springer, 2011.

[5] David Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.

[6] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[7] Julie D Thompson, Frédéric Plewniak, and Olivier Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucleic acids research*, 27(13):2682–2690, 1999.