Takizawa *et al. BMC Bioinformatics* (2020) 21:210
https://doi.org/10.1186/s12859-020-3535-5

BMC Bioinformatics

METHODOLOGY ARTICLE Open Access

RintC: fast and accuracy-aware decomposition of distributions of RNA secondary structures with extended logsumexp

Hiroki Takizawa[1], Junichi Iwakiri[1] and Kiyoshi Asai[1,2*]

*Correspondence:
asai@k.u-tokyo.ac.jp
[1]Graduate School of Frontier Sciences, The University of Tokyo Chiba, Japan
[2]Artificial Intelligence Research Center (AIRC), National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan

Abstract

Background: Analysis of secondary structures is essential for understanding the functions of RNAs. Because RNA molecules thermally fluctuate, it is necessary to analyze the probability distributions of their secondary structures. Existing methods, however, are not applicable to long RNAs owing to their high computational complexity. Additionally, previous research has suffered from two numerical difficulties: overflow and significant numerical errors.

Result: In this research, we reduced the computational complexity of calculating the landscape of the probability distribution of secondary structures by introducing a maximum-span constraint. In addition, we resolved numerical computation problems through two techniques: extended logsumexp and accuracy-guaranteed numerical computation. We analyzed the stability of the secondary structures of 16S ribosomal RNAs at various temperatures without overflow. The results obtained are consistent with previous research on thermophilic bacteria, suggesting that our method is applicable in thermal stability analysis. Furthermore, we quantitatively assessed numerical stability using our method..

Conclusion: These results demonstrate that the proposed method is applicable to long RNAs. .

Keywords: RNA secondary structure, Dynamic programming, Accuracy-guaranteed numerical computation, Interval arithmetic, Ribosomal RNA.

Background

Functional non-coding RNAs (ncRNAs) play essential roles in a wide range of biological phenomena. Secondary structures are often crucial to the functions of RNAs. A number of studies and software tools can predict a single secondary structure for a given RNA sequence [1]. According to detailed analyses of free energy, however, some RNAs do not always form a single stable structure. Therefore, quantitative evaluations of the fluctuation of RNA secondary structures have recently attracted attention. Recent studies have provided methods to analyze the distribution of RNA secondary structures in more detail using the marginal probability of Hamming Distance, in which each RNA structure is located in a discrete metric space [2–4].

Structures of long ncRNAs (*e.g.*, $>$ 1000 bases, including ribosomal RNAs) are important for understanding their functions, but analyzing the probability landscape of the structure remains a challenging task. Fourier transform has been utilized to reduce computational complexity [5–7], but the computational costs of previous methods are still too high to apply to long ncRNAs. Furthermore, the Fourier transform distributes numerical errors uniformly across large and small marginal probabilities, which makes small marginal probabilities unreliable.

Small marginal probabilities, however, are also of interest occasionally. In kinetic analyses, for example, meta-stable structures may have considerably higher free energy compared to the minimum free energy structure [8]. In such a case, the Boltzmann probability of the meta-stable regions can be very small. For reliable evaluation, quantitative assessment of numerical errors is necessary. Previous studies have described this type of numerical instability, but they have not shown detailed analyses [9].

To provide quantitative evaluation of numerical instability, we have implemented accuracy-guaranteed numerical computation based on interval arithmetic and evaluated the numerical errors associated with the Fourier transform. Interval arithmetic is a method in which arithmetic operations are defined along intervals expressing numerical values between the upper/lower edges. The approximate calculation of pi by Archimedes in the 3rd century BC is known as the oldest example of interval arithmetic. Around the 1950s, interval arithmetic came to be used for estimating the upper bounds on the numerical error caused by floating-point arithmetic in computers. For example, Sunaga [10] published one of the first studies in English on comprehensive algorithms for interval arithmetic for computers. Interval arithmetic for accuracy-guaranteed numerical computation has been established as a research field and detailed in a textbook [11].

To reduce computational costs, we have introduced the maximum-span constraint, which forbids long-range base-pairs. Such a constraint has been used for the prediction of secondary structures [12], but it has not been used to estimate marginal probabilities of discrete metrics (*e.g.*, Hamming distances). It may seem inappropriate to ignore long-range interactions in secondary structures because there are long-range interacting base-pairs in the 3D structures of long RNAs (*e.g.*, 16S rRNA). The predictions of long-range interactions, however, are known to be unreliable even if long-range base-pairs are allowed [13], while the widely used nearest neighbor energy model is not compatible with long RNAs and its parameters have been determined by experiments using short RNAs [14]. Accordingly, our method lost little by excluding long-range base-pairs. At the same time, we show that estimated stability based on our tool with maximum-span constraint was consistent with the previous research.

Maximum-span constraint has enabled the calculation of marginal probabilities on a discrete metric for long RNA sequences, but we had to cope with numerical overflow for calculations with long sequences. In stochastic models such as hidden Markov models and stochastic context free grammars, which are common for modeling and analyzing RNA structures, logsumexp (logarithm of the sum of exponentials) is the standard solution for preventing overflow or underflow in numerical calculation [15, 16]. There is a limitation, however, in that it cannot handle zero or negative values. This limitation is a problem when processing complex numbers with rectangular coordinates in the Fourier transform. One solution is to apply logsumexp only to radii using polar coordinates, but simple application of polar coordinates causes problems when combined with interval arithmetic for accuracy-guaranteed numerical computation. Complicated conditions occur when the angular interval crosses zero or the radius interval contains zero. In this paper, in addition to a radius of polar coordinates, normalized orthogonal coordinates, rather than angles, are combined for interval arithmetic of logsumexp. Consequently, we have realized the advantages of logsumexp and interval arithmetic while preserving the simplicity of implementation.

Results

Computation time

To demonstrate computational efficiency, the computation time of the proposed method using the S151 Rfam Dataset [15] was measured. In the proposed method, the reference structures were obtained by CentroidFold [17]($\gamma = 1.0$). All cores of the Intel Core i7 4770 CPU were used as a computational resource in this measurement.

We measured the computation time in the case where the maximum-span constraint $W = 100$ is introduced and in the case where no restriction is applied (equivalent to RintW [7]) (Fig. 1).

We also examined how the calculation time changes when the value of the maximum-span constraint W is changed (Fig. 2). In this experiment, 32 Intel Xeon Gold 6130 cores were used for computation.

Thermal stability of ribosomal RNA

As an application of the proposed method, we analyzed the thermal stability of the secondary structures of 16S rRNAs derived from *E. coli* and *T. thermophilus* using Credibility Limit [18] as the metric.

The Credibility Limit (0.5) of a given secondary structure was obtained with temperatures ranging from 37 to 55 degrees Celsius (Fig. 3). As the origin of the Hamming distance, three types of reference structure were prepared. (i) The "initial reference" was

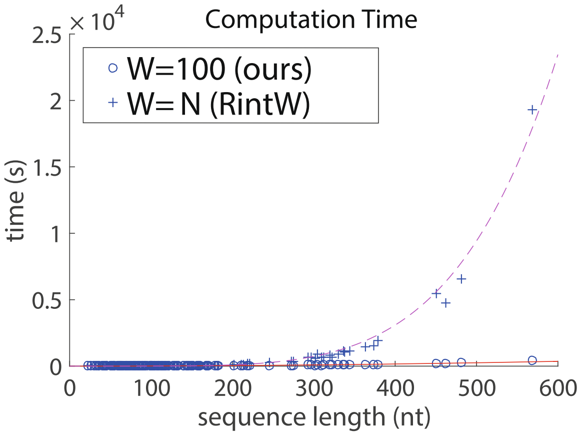

Fig. 1 Calculation time of the proposed method. The red solid line represents $y = 0.0010101x^2$. The purple dashed line represents $y = 3.0163e - 10x^5$. Both lines were fit to the result using the least squares method

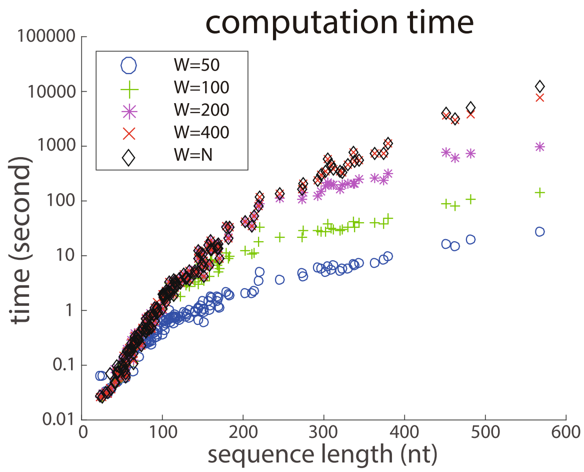

Fig. 2 Calculation time of the proposed method. Each data point is the calculation time of a single sequence from the S151Rfam dataset. The *y*-axis represents the logarithm of calculation time, and the *x*-axis represents the length of RNA

obtained using CentroidFold. (ii) The "refined reference" was obtained by the following steps: RintC was performed with the initial reference; a Hamming distance interval in which the probability is locally high was chosen; the BPPM of the interval was calculated; and a "refined reference" was obtained from the BPPM by posterior decoding with CentroidFold. (iii) The "natural reference" was the reference structure derived from the three-dimensional structure in NDB.

Numerical error evaluation

For a quantitative evaluation of RintC numerical error, accuracy-guaranteed numerical computations with interval arithmetic were applied to the calculation process of RintC

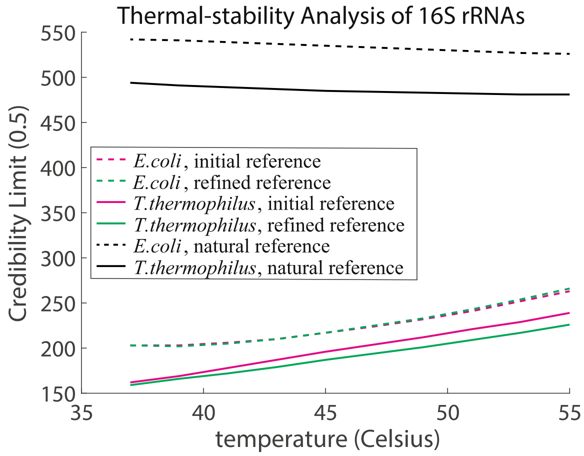

Fig. 3 Thermal-stability analysis for secondary structures of *E. coli* and *T. thermophilus* 16S rRNAs. The "initial reference" is the reference structure obtained using CentroidFold. The "refined reference" is the reference structure obtained using RintC and the base-pairing probability matrix (BPPM) (see the "Thermal stability of ribosomal RNA" section for the details). The "natural reference" is the reference structure derived from the three-dimensional structure. The "Experimental procedure" section provides a detailed description of the "natural reference"

with the RF00008B sequence in the S151 Rfam dataset [15]. The length of the RF00008B sequence was short enough for the evaluation of time-consuming calculation without any type of Fourier transform. The numerical errors of three types of calculation (DFT, FFT, and non-FFT) are shown in Fig. 4.

Numerical error evaluation was also conducted for all sequences in the S151 Rfam dataset and *E. coli* 16S rRNA (Fig. 5). Each data point corresponds to an individual sequence in the S151 Rfam dataset or *E. coli* 16S rRNA. In comparisons of the numerical errors between DFT and FFT versions (Fig. 5a), DFT is always more accurate than FFT. This result is consistent with that shown in Fig. 4. In addition, a relationship between the numerical error and sequence length in the DFT results was also investigated (Fig. 5b).

Discussion

Computational efficiency

Figure 1 shows that the computation time when using the constraint $W = 100$ follows the theoretical complexities of the square of the length of the sequence, while computational time scales with the fifth power of the length of the sequence when no constraint is used. This confirms that computational complexity is drastically reduced by introducing the maximum-span constraint into the proposed method.

As Fig. 2 shows, when $W < N$, the calculation time is reduced by the effect of the maximum-span. When $N \leq W$, the same calculation time is required regardless of the value of W. Since many of the data points were short RNAs, the differences for large W values were unclear. Nevertheless, the relationship between W and computation time was consistent with the theory. In the next subsection, we demonstrate that the proposed method works for long RNAs.

Thermal stability of ribosomal RNA

The credibility limits of natural references are much higher than those of others, because natural references include long-range base-pairs. Maximum-span constraint $W = 100$ was introduced because most of the actual spans of base-pairs are less than or equal to 100 bases (Supplementary Table S1), but long-range interactions may play an important role in structures. Thermophilic bacteria have reduced dynamics of intracellular macromolecules (mainly proteins) compared with mesophilic bacteria [19]. Several thermophilic RNAs exhibit higher thermal stability than the mesophilic homologous [20, 21]. In addition, thermal adaptation of the thermophilic ribosomal subunit including 16S rRNA has been suggested by structural and evolutionary analysis [22]. Our result using

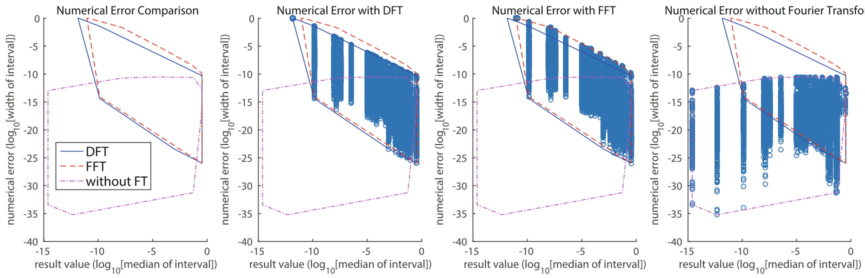

Fig. 4 The result of the numerical error experiment with RF00008B (54 nucleotides). The leftmost plot explains the convex hulls for the result values and their errors under each experimental condition. The three plots to the right are scatter plots of the raw data for the result values and errors under one experimental condition and the convex hulls for each. In this evaluation, the reference structure was obtained by CentroidFold

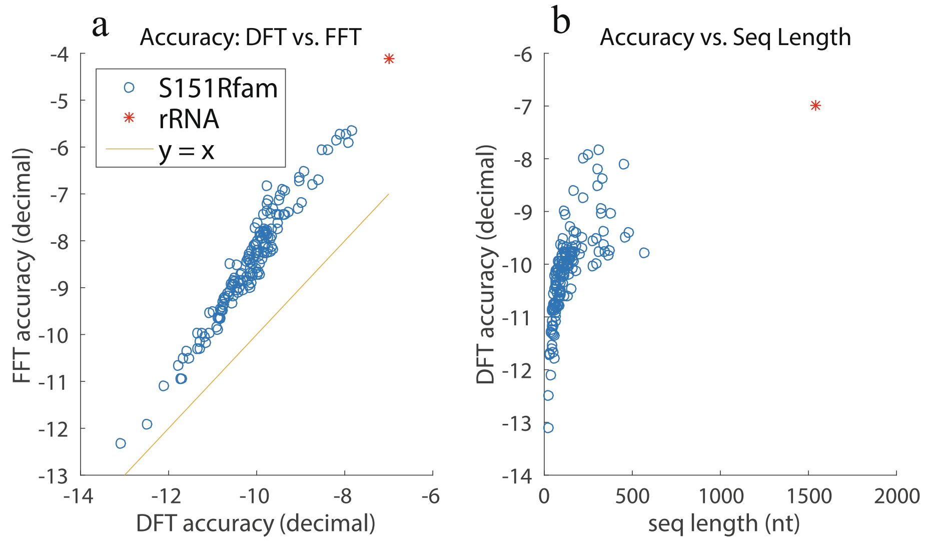

Fig. 5 The results of the numerical error experiment. **a** Numerical error comparison. **b** Relationship between sequence length and numerical error. The *x*- and *y*-axes have minima of $(log_{10}(median) + log_{10}(width))$ under the DFT and FFT methods. The reference structure was obtained using CentroidFold [17]. ($\gamma = 1.0$)

maximum-span constraint $W = 100$ shows that the 16S rRNA of *T. thermophilus* had a lower Credibility Limit than that of E. coli, which also implies not only the protein components but also the rRNA playing a role in the thermal stability of thermophilic ribosomes (Fig. 3).

The use of the natural secondary structure as a representative structure exhibited a relatively higher Credibility Limit, compared with the "initial" and "refined" references. This implies the DP calculation with the Turner model is compatible for the representative structure derived from the Turner model, such as the "initial" and "refined" references. Note that this result would not indicate the advantage of "initial" and "refined" structures over the "natural" structures.

Numerical error evaluation

As Fig. 4 shows, for the DFT and FFT methods, the numerical error (i.e., interval width) is almost equal to 1, when the calculated existence probability is quite small. Interval width = 1 indicates that the probability is within $[0, 1]$, thus providing no meaningful information owing to the numerical error. In contrast, the numerical error remains low when the existence probability is moderate or high. DFT-based results are slightly more accurate than FFT-based results. In further numerical error comparisons between non-Fourier transform results and DFT or FFT results, the numerical error of the non-Fourier result is smaller than those of the DFT or FFT results. This implies that the problematic numerical error is indeed caused by Fourier transform.

Figure 5b demonstrates that the numerical error in the marginal probability of the structures for long RNA sequences ($\geq$ 1000 nt) is sufficiently small (about 10^{-7} for 16S rRNA) for the structures with a moderate or high probability of existence. This accuracy is sufficient for thermal stability analysis because an accurate evaluation of large clusters is only required for their analysis.

Conclusions

Since RNA secondary structures have large thermal fluctuations, prediction of the most stable secondary structure is insufficient for representing native structural behavior of an RNA molecule. Marginal probabilities on Hamming distances from reference structures, which represent the landscape of all the possible RNA secondary structures, can be efficiently computed by combining Fourier transform with dynamic programming, but the computational costs are still too high for long RNAs.

In this research, we have implemented a maximum-span constraint of base-pairs to reduce computational complexity. For long RNAs, however, there remains another problem: numerical overflow. As the standard method for avoiding overflow in stochastic models, logsumexp (logarithm of the sum of exponentials) is not directly applicable to Fourier transforms, we have developed an extended logsumexp method for whole complex numbers. We have shown that reduced computational time enables us to analyze the thermal stability of long RNAs, such as 16S ribosomal RNAs, while the predicted structures using the same maximum-span constraint tend to be inaccurate.

We have also adopted accuracy-guaranteed numerical computation with interval arithmetic to evaluate numerical errors. We have shown that numerical errors for small probabilities are substantial when FFT or DFT is used. Quantitative assessment of the observed numerical instabilities, however, revealed that our method achieves sufficient numerical accuracy for thermodynamic stability analysis of RNA secondary structures. These results demonstrate that our method is a powerful tool for understanding long RNAs.

Methods

RintW + maximum-span

Initially, we introduced a maximum-span constraint in base-pairs to the baseline algorithm of RintW [7]. Detailed descriptions of RintW and the proposed method are described below. The inputs of the algorithm are an RNA sequence and a reference secondary structure, and the outputs are the existence probability and the base-pairing probability matrix (BPPM) for each Hamming distance from the reference secondary structure.

RNA secondary structure representation

As a computationally efficient expression, the RNA secondary structure was represented by a binary upper triangular matrix σ where each element is $\{0, 1\}$. Each element of σ is decided as follows.

$$\sigma_{i,j} = \begin{cases} 1 \ (i < j \ and \ (i,j) \ forms \ a \ base \ pair) \\ 0 \ (otherwise) \end{cases}$$

The distance between two RNA secondary structures σ_1, σ_2 are determined by the number of elements with different values, namely, Hamming distance values.

We used Hamming distance as the discrete metric of our implementation, as was used in previous studies [2, 5, 7]. The Hamming distance corresponds to the number of unit changes of the secondary structure over time, that is, the forming or breaking a base-pair. Natural distance satisfying axioms can be used to track the trajectory of the structural changes of RNAs. Hamming distance is compatible with efficient dynamic programming algorithms that can be constructed. We know that there are more important base-pairs and less important base-pairs for the function of RNAs, but Hamming distance is at least a convenient metric to observe the landscape of the probability distribution of the secondary structures.

Only secondary structures that satisfy the following constraints were considered ($N =$ sequence length).

1 Only Watson–Crick base-pairs (A-T, C-G) or wobble base-pairs(G-U) exist.
2 Prohibition of pseudo-knots: For all $1 \leq i < j < k < l \leq N$, (i, k) and (j, l) do not form base-pairs at the same time.
3 Max loop constraint: For all $1 \leq i < j < k < l \leq N$, if (i, l) and (j, k) form base-pairs and no paired base exists between $i + 1$ and $j - 1$ nor between $k + 1$ and $l - 1$, then $j - i + l - k \leq C + 2$. C is a max-loop parameter, and we used $C = 30$ following many previous studies.
4 Max span constraint: For all $1 \leq i < j \leq N$, if (i, j) forms a base-pair, then $j - i \leq W$.

Constraints 1 and 2 are the standard constraints used in the previous methods [2, 5, 7]. Constraint 3 is called the max loop constraint. This constraint was adopted by many RNA secondary structure analysis methods using the energy model [14] described in the next section. This constraint reduces time complexity. It is empirically known that this constraint has little effect on the calculation result. Constraint 4 is a constraint studied in previous work [12, 13, 23–25], but it was not used in RintW[7] until we introduced it. This constraint is considered to be suitable for examining local structural motifs [13].

Energy model

The nearest neighbor energy model [14], which can be analyzed by dynamic programming, was adopted. The energy of the secondary structure was expressed as the sum of the following five functions in this model.

1 $f_h(i, j) =$ the energy of base-pair (i, j) forming a hairpin loop.
2 $f_l(i, j, k, l) =$ the energy of base-pairs, (i, l) and (j, k), making a 2-Loop when $i < j < k < l$.
3 $f_{mc} =$ the energy of having one multi-loop.
4 $f_{mi} =$ the energy of having one internal multi-loop branch.
5 $f_d(i, j) =$ the energy of a base-pair (i, j) forming a multi-loop or being an outermost base-pair.

Polynomial approach

In previous research [5–7], the polynomial approach was used as a method to reduce the time complexity of dynamic programming. A naïve dynamic programming method requires a convolution operation. This operation is regarded as computation in the spatial domain and is expressed by calculation in the frequency domain. A convolution operation can be converted to an inner product, thus reducing computational complexity. After completing the dynamic programming computation, a shift to the spatial domain is achieved by performing the Fourier transform. The same method was used in this study.

Preprocessing

As in the RintW algorithm, we calculate the following $g_0^Z(i, j)$ functions in $O(N^2)$ time as preprocessing, to obtain the gains of the Hamming distances g_1^Z to g_8^Z and g_1^W to g_5^W:

for $(1 \le i \le N)$

$$g_0^Z(i,i) = g_0^Z(i,i-1) = 0$$

for $(1 \le i < j \le N)$

$$\begin{aligned} g_0^Z(i,j) &= \sum_{p=i}^{j-1} \sum_{q=p+1}^{j} \sigma_{pq} \\ &= \sigma_{ij} + g_0^Z(i+1,j) + g_0^Z(i,j-1) - g_0^Z(i+1,j-1). \end{aligned}$$

Here, σ is a binary matrix representation of the reference secondary structure. The maximum Hamming distance of the secondary structure from the representative secondary structure (H_{max}) is also computed at this time [6].

Definitions of function gs

Prior to the description of the main algorithm, auxiliary functions for calculating the distance between substructures are defined as follows. These functions are the same as those used in previous studies [6, 7].

$$\begin{aligned} b(i,j) &= 1 - 2\sigma_{ij} \\ g_1^Z(i,j) &= g_0^Z(i,j) \\ g_2^Z(i,j,k) &= g_0^Z(i,j) - g_0^Z(i,k) - g_0^Z(k+1,j) \\ g_3^Z(i,j,k) &= g_0^Z(i,j) - g_0^Z(i,k) \\ g_4^Z(i,j) &= g_0^Z(i,j) + b(i,j) \\ g_5^Z(i,j,k,l) &= g_0^Z(i,j) - g_0^Z(k,l) + b(i,j) \\ g_6^Z(i,j,k) &= g_0^Z(i,j) - g_0^Z(i+1,j-1) - g_0^Z(k,j-1) + b(i,j) \\ g_7^Z(i,j,k) &= g_0^Z(i,j) - g_0^Z(k,j) \\ g_8^Z(i,j,k) &= g_0^Z(i,j) - g_0^Z(i,k-1) - g_0^Z(k,j) \\ g_9^Z(i,j,k) &= g_3^Z(i,j,k) \\ g_1^W(i,j) &= g_0^Z(1,N) - g_0^Z(i,j) - g_0^Z(1,i-1) - g_0^Z(j+1,N) \\ g_2^W(i,j,h,l) &= g_0^Z(h,l) - g_0^Z(i,j) + b(h,l) \\ g_3^W(i,j,h,l) &= g_2^W(i,j,h,l) - g_0^Z(h+1,i-1) \\ g_4^W(i,j,h,l) &= g_2^W(i,j,h,l) - g_0^Z(j+1,l-1) \\ g_5^W(i,j,h,l) &= g_2^W(i,j,h,l) - g_0^Z(h+1,i-1) - g_0^Z(j+1,l-1). \end{aligned}$$

These functions calculate the Hamming distance of a substructure from the reference substructure. More specifically, in the binary matrix representation of the structure, each g_*^* accumulates differences in rectangular regions of the matrix. According to Mori et al. [6], by changing this function, one can decompose the structures by another distance metric (i.e., other than the Hamming distance), which indicates further potential of this concept, but this was outside of the scope of the study.

Dynamic programming of the partition function

In the following equations, x is the $(H_{max}+1)$-th root of unity. If Cooley–Tukey fast Fourier transform (FFT) is used instead of discrete Fourier transform (DFT) in post-processing, x is the smallest power of 2 that is equal to or greater than $(H_{max}+1)$. There are $(H_{max}+1)$ kinds of $(H_{max}+1)$-th roots of unity calculated independently. Therefore, parallel computation is possible.

In order to avoid overflow, the proposed extended logsumexp computation is used. In the following equations, g_*^* and $-\frac{f_*}{kT}$ are real exact numbers (i.e., logsumexp was not applied), but $x^{g_*^*}$ and $e^{-\frac{f_*}{kT}}$ are converted into a complex logsumexp type (i.e., only exponents were recorded). Consequently, All DP-variable $Z_{*,*}^*$, $W_{*,*}^*$, and $Q_{*,*}^*$ values are also of complex logsumexp type.

Initialization:

$$\text{for } (1 \le i \le N)$$

$$Z_{i,i} = Z_{i,i-1} = 1$$

$$Z_{i,i}^1 = Z_{i,i}^b = Z_{i,i}^m = Z_{i,i-1}^m = Z_{i,i}^{m1} = 0$$

$$W_{1,N}^b = \begin{cases} 1 & ((1,N)\ \textit{forms a base pair}) \\ 0 & (\textit{otherwise}) \end{cases}$$

Recursion:

for $(1 \le i < j \le N)$ s.t. $(j - i \le W)$

$$Z_{1,j} = x^{g_1^Z(1,j)} + \sum_{h=1}^{j-1} Z_{1,h-1} Z_{h,j}^1 x^{g_2^Z(1,j,h)}$$

$$Z_{i,N} = Z_{i+1,N} x^{g_7^Z(i,N,i+1)} + \sum_{h=i+1}^{min(N,i+W)} Z_{i,h}^b e^{-\frac{f_d(i,h)}{kT}} Z_{h+1,N} x^{g_2^Z(i,N,h)}$$

$$Z_{i,j}^1 = \sum_{h=i+1}^{min(j,i+W)} Z_{i,h}^b e^{-\frac{f_d(i,h)}{kT}} x^{g_3^Z(i,j,h)}$$

$$Z_{i,j}^b = e^{-\frac{f_h(i,j)}{kT}} x^{g_4^Z(i,j)} + \sum_{h=i+1}^{min(i+C+1,j-2)} \sum_{l=max(h+1,j+h-i-C-2)}^{j-1} Z_{h,l}^b e^{-\frac{f_l(i,h,l,j)}{kT}} x^{g_5^Z(i,h,l,j)} + \sum_{h=i+2}^{j-1} Z_{i+1,h-1}^m Z_{h,j-1}^{m1} e^{-\frac{f_d(j,i)+f_{mc}}{kT}} x^{g_6^Z(i,j,h)}$$

$$Z_{i,j}^m = \sum_{h=i}^{j-1} (x^{g_7^Z(i,j,h)} + Z_{i,h-1}^m x^{g_8^Z(i,j,h)}) Z_{h,j}^{m1}$$

$$Z_{i,j}^{m1} = \sum_{h=i+1}^{j} Z_{i,h}^b e^{-\frac{f_d(i,h)+f_{mi}}{kT}} x^{g_3^Z(i,j,h)}$$

$$W_{i,j}^b = Z_{1,i-1} Z_{j+1,N} e^{-\frac{f_d(i,j)}{kT}} x^{g_1^W(i,j)}$$

$$+ \sum_{h=max(1,i-C-1,i-W)}^{i-1} \sum_{l=j+1}^{min(N,h+W,j+h-i+C+2)} W^b_{h,l} e^{-\frac{f_l(h,i,j,l)}{kT}} x^{g^W_2(h,i,j,l)}$$

$$+ \sum_{h=max(1,i-W)}^{i-1} \sum_{l=j+1}^{min(N,h+W)} W^b_{h,l} e^{-\frac{f_d(l,h)+f_{mc}+f_d(i,j)+f_{mi}}{kT}} ($$

$$Z^m_{h+1,i-1} x^{g^W_3(h,i,j,l)}$$

$$+ Z^m_{j+1,l-1} x^{g^W_4(h,i,j,l)}$$

$$+ Z_{h+1,i-1} Z_{j+1,l-1} x^{g^W_5(h,i,j,l)})$$

$$Q^b_{i,j} = Z^b_{i,j} W^b_{i,j}$$

The $Z_{*,*}$ functions are the *inside* partition functions, which represent the sums of all the Boltzmann factors in the corresponding sub-sequences. $Z^1_{*,*}$, $Z^b_{*,*}$, $Z^m_{*,*}$, and $Z^{m1}_{*,*}$ are the specified partition functions defined in the McCaskill algorithm [1]. $W^b_{i,j}$ is the *outside* partition function, which represents the outside of the base-pair (i,j). The $Q^b_{i,j}$ is the conditional partition function, the sum of all the Boltzmann factors when (i,j) forms a base-pair.

The intuitive meanings of $Z^*_{*,*}$ and $W^b_{*,*}$ are as follows. $Z^b_{i,j}$ ($W^b_{i,j}$) is the inside (outside) partition function of partial structure between the i-th base and j-th base when the i-th base and the j-th base form a base-pair. $Z^*_{i,j}$ is the inside partition function for different conditions. For example, $Z^1_{i,j}$ accumulates the cases in which only one outmost base-pair exists, whose 5' base is the i-th base, while $Z^m_{i,j}$ and $Z^{m1}_{i,j}$ are considered only for multi-loops.

The values g^Z_1 to g^Z_8 and g^W_1 to g^W_5, which are computed using the pre-computed function $g^Z_0(i,j)$, are the gains of the Hamming distance for the transitions represented by the recursions of partition functions. The significant difference from RintW is that the recursions of $Z_{1,n}$, $Z^1_{i,j}$, and $W^b_{i,j}$ include the maximum-span constraint W of base-pairs in their range of the sum. A small improvement in this approach is that only the required edges, namely, $Z_{1,j}$ and $Z_{i,N}$, are calculated instead of calculating all $Z_{*,*}$ values. Regarding the maximum-span constraint of base-pairs, the algorithmic concept is equivalent to the calculation of dynamic programming (DP) variables α_{Outer} and β_{Outer} in Rfold [12] and ParasoR [13], but the notation of RintW is followed in the above recursions.

Fourier transform and post-processing

The conditional partition function on each Hamming distance, $Q^b_{i,j}$, is efficiently obtained by Fourier transformation. For all (i,j), such that $(1 \leq i \leq j \leq N)$ and $(j - i \leq W)$, a complex number sequence of $(H_{max} + 1)$ elements are calculated. Let $Z(d)_{1,N}$ and $Q(d)^b_{i,j}$ be the conditional partition functions for a Hamming distance d of $Z_{1,n}$ and $Q^b_{i,j}$, respectively. Then, the existence probability of Hamming distance d is written as

$$\frac{Z(d)_{1,N}}{\sum_{d=0}^{H_{max}} Z(d)_{1,N}},$$

and the BPPM for Hamming distance d is written as

$$\frac{Q(d)^b_{i,j}}{Z(d)_{1,N}}.$$

The obtained partition functions and probabilities mutually differ by several tens of digits. However, since all variables are convoluted during post-processing, all numerical errors propagate to all variables. This makes marginal probabilities of small values unreliable.

Computational complexity

In the following description, N is the length of the sequence, H_{max} is the maximum Hamming distance from the reference structure, and U is the degree of parallelism. Here $U \leq H_{max} + 1$ is assumed. In the original RintW algorithm, the computational complexity of pre-processing is $O(N^2)$ in both time and space. In the partition function calculation, the time complexity is $O(N^4 H_{max}/U)$ and the space complexity is $O(N^2 H_{max} U)$. The original RintW uses DFT for post-processing; the time complexity of the post-processing part is $O(N^2 H_{max}^2/U)$, and its space complexity is $O(H_{max} U)$. Since $H_{max} \leq N$ holds, the computational complexity of the post-processing can be ignored in total complexity in both time and space. Finally, the time and space complexity of the original RintW algorithm as a whole are $O(N^4 H_{max}/U)$ and $O(N^2 H_{max} U)$, respectively.

When the maximum-span constraint is introduced, the computational complexity of pre-processing remains $O(N^2)$ in both time and space. In the distribution function calculation, the time complexity is $O(NW^3 H_{max}/U)$, and the space complexity is $O(NWH_{max}U)$ for the maximum-span of base-pair W. When DFT is used for post-processing, the complexity of the post-processing is $O(NWH_{max}^2/U)$ in time and $O(H_{max}U)$ in space. Because the H_{max} may be close to N, the computational complexity of the post-processing cannot be ignored. By using FFT instead of DFT, we can reduce the time complexity of the post-processing component to $O(NWH_{max} log(H_{max})/U)$. Then, the total computational complexity is $O(N(N + WH_{max}(W^2 + log(H_{max}))/U))$ in time and $O(N(N + WH_{max}U))$ in space.

The summary of computational complexities is shown, with the notation simplified by using $H_{max} \leq N$, in Table 1.

Table 1 Computational complexity of the existing and proposed methods are summarized

	RintW, time	RintC (proposed), time
preprocessing	$O(N^2)$	$O(N^2)$
main calculation	$O(N^5/U)$	$O(N^2W^3/U)$
postprocessing (DFT)	$O(N^4/U)$	$O(N^3W/U)$
postprocessing (FFT)	$O(N^3 logN/U)$	$O(N^2 WlogN/U)$
total (DFT)	$O(N^5/U)$	$O(N^2W(W^2 + N)/U)$
total (FFT)	$O(N^5/U)$	$O(N^2W(W^2 + logN)/U)$
	RintW, space	RintC (proposed), space
preprocessing	$O(N^2)$	$O(N^2)$
main calculation	$O(N^3U)$	$O(N^2WU)$
postprocessing (DFT)	$O(NU)$	$O(NU)$
postprocessing (FFT)	$O(NU)$	$O(NU)$
total (DFT)	$O(N^3U)$	$O(N^2WU)$
total (FFT)	$O(N^3U)$	$O(N^2WU)$

N = sequence length. W = maximum-span. Note that $H_{max} \leq N$ and $W \leq N$ always holds. U = degree of parallelism

Interval arithmetic and accuracy assurance

In this subsection, we briefly explain the rounding mode control function of IEEE 754 and the accuracy assurance arithmetic. Representing real numbers by floating-point numbers can cause deviations from actual values. Therefore, numerical values can conceivably be held as an interval including the actual value. We define arithmetic operations between intervals to obtain an interval necessarily containing the results of arithmetic operations on actual values. Then, the upper bound of the numerical error is obtained as the width of the interval of the calculation result. Most modern computers use the IEEE 754 method for floating-point arithmetic. This method has a rounding mode control function, and we can specify truncation and rounding-up. By using this function, the accuracy assurance calculation described above can be executed efficiently. Our accuracy assurance calculation used the kv library [26]) implemented in C++. The kv library is open source software and requires only C++ Boost for its backend.

Logsumexp on complex numbers with interval arithmetic

A method to perform logsumexp computation on whole complex numbers has been developed. Details of the calculation algorithm are provided in the following subsections. There are different parts of algorithms for scalar and interval types, but those for scalar types are described in the supplementary file. In this subsection, only methods for interval types are described. If only the scalar type is considered, the complex number defined in polar coordinates and logsumexp defined only in terms of a radius are sufficient. Extensions to interval arithmetic, however, are complicated.

The Vienna RNA Package [27] prevents overflow by scaling. Their scaling factor construction is sophisticated, and under some assumptions, the scaling is equivalent to a kind of logsumexp. The original RintW [7] also utilized the same scaling technique as Vienna. However, with Vienna's method, the deviation between the scaling factor and the value of the actual distribution function can increase exponentially, so overflowing cannot be completely avoided. Unlike them, logsumexp does not need scaling factors, and overflows are completely avoided.

Notation and representation

In this subsection, a bracketing character like $[\,x]$ indicates an interval type variable. A pair of values in a bracket (e.g., $[\,0,1]$) indicates a closed interval. When two variables are enclosed (e.g., $[\,x,y]$), each variable x and y is a scalar type (or floating-point type), not an interval type. It is possible to convert one scalar x into an interval type while guaranteeing accuracy. Such an interval variable is expressed as $[\,x,x]$ (i.e., $[\,x,x]$ is an interval that includes the real value x). Finally, a function $f_{upper}([\,x]\,)\,=\,u$ for obtaining the maximum value of the interval type variable $[\,x]=[\,l,u]$, a function $f_{lower}([\,x]\,)\,=\,l$ for obtaining the minimum value, and a function $f_{mid}([\,x]\,)\,=\,\frac{l+u}{2}$ for obtaining the median value are used. However, it is assumed that they are not necessarily accuracy-guaranteed functions.

To represent the complex number $[\,a]+[\,b]\,i$, $([\,r]\,,[\,c]\,,[\,d]\,)$ is held for

$$[\,a]+[\,b]\,i\subseteq e^{[r]}([\,c]+[\,d]\,i)$$

.

However, as a normalization condition,

$$f_{upper}([\,c]^2+[\,d]^2\,)=\begin{cases}0 & ([\,a]=[\,0,0]\quad and\quad [\,b]=[\,0,0]\,)\\ 1 & (otherwise)\end{cases}$$

must be satisfied. It is assumed that 1 is numerically almost 1. The difference from 1 accumulates by multiplication, but it is reset by addition. For convenience, $[\,r]=[\,0,0]$ must be satisfied when $([\,a]=[\,0,0] \quad and \quad [\,b]=[\,0,0]\,)$.

The conversion protocol between this and the usual representation is described in the supplementary file. Normalization, multiplication, and addition protocols are described below.

Normalization

When a number $([\,r'],[\,c'],[\,d']\,)$ that is not normalized is given, a method of obtaining the normalized number with accuracy assurance $([\,r],[\,c],[\,d]\,) \supseteq ([\,r'],[\,c'],[\,d']\,)$ is as follows.

Algorithm 1 Normalize

```
Input: $([\,r'],[\,c'],[\,d']\,)$::(Interval,Interval,Interval)
Output: $([\,r],[\,c],[\,d]\,)$::(Interval,Interval,Interval) where
    $e^{[r]}([\,c]+[\,d]\,i) \supseteq e^{[r']}([\,c']+[\,d']\,i)$ and
    IsNormalized($[\,r],[\,c],[\,d]$) = True
s ⇐ $f_{upper}([\,c']^2+[\,d']^2\,)$
if $s = 0$ then
    return $([\,0,0],[\,0,0],[\,0,0]\,)$
end if
$t \Leftarrow \frac{1}{sqrt(s)}$
return $([\,r']-log([\,t,t]\,),[\,t,t]\,[\,c'],[\,t,t]\,[\,d']\,)$
```

Description:

First, compute

$$s=f_{upper}([\,c']^2+[\,d']^2\,)$$
$$t=\frac{1}{sqrt(s)}$$

,

where t is the reciprocal of the maximum value of the absolute value of the input. At this time

$$([\,r],[\,c],[\,d]\,)=\begin{cases}([\,0,0],[\,0,0],[\,0,0]\,) & (s=0)\\ ([\,r']-log([\,t,t]\,),[\,t,t]\,[\,c'],[\,t,t]\,[\,d']\,) & (otherwise)\end{cases}$$

is a normalized solution.

Multiplication

The multiplication of the two values $([\,r_1],[\,c_1],[\,d_1]\,)$ and $([\,r_2],[\,c_2],[\,d_2]\,)$ can be described as

$$\begin{aligned}&([\,r_1],[\,c_1],[\,d_1]\,)([\,r_2],[\,c_2],[\,d_2]\,)\\ &=e^{[r_1]}([\,c_1]+[\,d_1]\,i)e^{[r_2]}([\,c_2]+[\,d_2]\,i)\\ &=e^{[r_1]+[r_2]}([\,c_1]+[\,d_1]\,i)([\,c_2]+[\,d_2]\,i)\\ &=e^{[r_1]+[r_2]}(([\,c_1]\,[\,c_2]-[\,d_1]\,[\,d_2]\,)+([\,c_1]\,[\,d_2]+[\,d_1]\,[\,c_2]\,)i),\end{aligned}$$

and $([r_1]+[r_2], [c_1][c_2]-[d_1][d_2], [c_1][d_2]+[d_1][c_2])$ is obtained as a solution. In normalization post-processing, if $[c_1][c_2]-[d_1][d_2]=[c_1][d_2]+[d_1][c_2]=[0,0]$, $[r]=[0,0]$ is substituted. Otherwise, because the product of the complex numbers with absolute value 1 is absolute value 1, it is naturally normalized.

Algorithm 2 Multiplication

```
Input: $(([r_1], [c_1], [d_1]), ([r_2], [c_2], [d_2]))$::
    ((Interval,Interval,Interval),(Interval,Interval,Interval)) where
        IsNormalized($[r_1], [c_1], [d_1]$) = True and
        IsNormalized($[r_2], [c_2], [d_2]$) = True
Output: $([r], [c], [d])$::(Interval,Interval,Interval) where
    $e^{[r]}([c]+[d]i) \supseteq e^{[r_1]}([c_1]+[d_1]i)e^{[r_2]}([c_2]+[d_2]i)$ and
    IsNormalized($[r], [c], [d]$) = True
$[r] \Leftarrow [r_1]+[r_2]$
$[c] \Leftarrow [c_1][c_2]-[d_1][d_2]$
$[d] \Leftarrow [c_1][d_2]+[d_1][c_2]$
if $([c], [d]) = ([0,0], [0,0])$ then
    $[r] \Leftarrow [0,0]$
end if
return $([r], [c], [d])$
```

Addition

Consider the sum of the two values $([r_1], [c_1], [d_1])$ and $([r_2], [c_2], [d_2])$. As addition is commutative, assuming $f_{mid}([r_1]) \geq f_{mid}([r_2])$ does not decrease generality. Then, it can be formulated as

$$\begin{aligned} p = & f_{upper}([r_1]) - f_{mid}([r_1]) \\ & + f_{upper}([r_2]) - f_{mid}([r_2]) \quad (p \geq 0) \end{aligned}$$

and

$$\begin{aligned} & ([r_1], [c_1], [d_1]) + ([r_2], [c_2], [d_2]) \\ & = e^{[r_1]}([c_1]+[d_1]i) + e^{[r_2]}([c_2]+[d_2]i) \\ & = e^{[r_1]}([c_1]+[d_1]i) + e^{[r_1]}(e^{[r_2]-[r_1]}[c_2]+e^{[r_2]-[r_1]}[d_2]i) \\ & = e^{[r_1]}(([c_1]+e^{[r_2]-[r_1]}[c_2]) + ([d_1]+e^{[r_2]-[r_1]}[d_2])i) \\ & = e^{[r_1]+[p,p]}((e^{-[p,p]}[c_1]+e^{[r_2]-[r_1]-[p,p]}[c_2]) \\ & \quad + (e^{-[p,p]}[d_1]+e^{[r_2]-[r_1]-[p,p]}[d_2])i). \end{aligned}$$

Thus, $f_{upper}(e^{-[p,p]}) \leq 1$ follows from the assumption of $p \geq 0$. Additionally, $f_{upper}(e^{[r_2]-[r_1]-[p,p]}) \leq 1$ follows from the assumption that $f_{mid}([r_1]) \geq f_{mid}([r_2])$ (the proof is provided in the supplementary file). Therefore, $e^{-[p,p]}$ and $e^{[r_2]-[r_1]-[p,p]}$ can be directly calculated without overflow occurring. Therefore,

$$\begin{aligned} & = [r_1]+[p,p] \\ [c'] & = (e^{-[p,p]}[c_1]+e^{[r_2]-[r_1]-[p,p]}[c_2]) \\ [d'] & = (e^{-[p,p]}[d_1]+e^{[r_2]-[r_1]-[p,p]}[d_2]) \end{aligned}$$

can be calculated, and $([r'],[c'],[d'])$ satisfies

$$([r_1],[c_1],[d_1]) + ([r_2],[c_2],[d_2]) = ([r'],[c'],[d'])$$

as the summation. Finally, since this is not normalized, normalization processing is required.

Algorithm 3 Addition

```
Input: $(([r_1],[c_1],[d_1]),([r_2],[c_2],[d_2]))$::
    ((Interval,Interval,Interval),(Interval,Interval,Interval)) where
        IsNormalized($[r_1],[c_1],[d_1]$) = True and
        IsNormalized($[r_2],[c_2],[d_2]$) = True
Output: $([r],[c],[d])$::(Interval,Interval,Interval) where
    $e^{[r]}([c]+[d]i) \supseteq e^{[r_1]}([c_1]+[d_1]i) + e^{[r_2]}([c_2]+[d_2]i)$ and
    IsNormalized($[r],[c],[d]$) = True
if $f_{mid}([r_1]) < f_{mid}([r_2])$ then
    return Addition($([r_2],[c_2],[d_2]),([r_1],[c_1],[d_1])$)
end if
$p \Leftarrow f_{upper}([r_1]) - f_{mid}([r_1]) + f_{upper}([r_2]) - f_{mid}([r_2])$
$[k_1] \Leftarrow e^{-[p,p]}$
$[k_2] \Leftarrow e^{[r_2]-[r_1]-[p,p]}$
$[r] \Leftarrow [r_1] + [p,p]$
$[c] \Leftarrow [k_1][c_1] + [k_2][c_2]$
$[d] \Leftarrow [k_1][d_1] + [k_2][d_2]$
return Normalize($[r],[c],[d]$)
```

In the classic logsumexp, numerical errors of summation are reduced by using a summation-specific technique rather than recursively using the two-operand addition function. For the summation-specific technique, in three or more operands, one can use the maximum number as the scaling factor and scale the others. On the other hand, we developed only the normal two-operand addition function. The following experiment shows that our method brings sufficient numerical accuracy. Nevertheless, further improvement may still be possible.

Requirements for an accuracy assurance calculation library

The functions that the accuracy assurance calculation library must perform in this method are as follows:

1 The conversion from scalar to interval type guarantees accuracy.
2 Four arithmetic operations, log, and exp, with accuracy assurance for the interval type.
3 The previously described $f_{upper}([x])$ and $f_{mid}([x])$.

Credibility limit

In order to evaluate the magnitude of thermal fluctuation, we used Credibility Limit [18] as the metric. Credibility Limit is the minimum distance in which a certain percentage of structures is distributed. More specifically, given a representative structure σ and a distance d, consider p_d, the sum of Boltzmann probabilities of structures whose distances from σ are less than or equal to d. Then, given a probability value p, CL(p) is the smallest d such that $p_d \geq p$. The larger the Credibility Limit value, the more intense the thermal fluctuation of the molecule.

Experimental procedure

The S151 Rfam Dataset 'with all pseudoknots removed' [15] was used for evaluation of time complexity and numerical accuracy in interval operation.

For the application of our proposed method to RNA molecules longer than those in the S151 Rfam dataset [15], the primary sequences and the corresponding native secondary structures of 16S rRNAs were obtained from three-dimensional structures of *E. coli* and *T. thermophilus*, while those of 70S ribosomes were from the Nucleic Acid Database (NDB) [28, 29]. The NDB IDs of the *E. coli* and *T. thermophilus* ribosome structures were 4V9D (chainID: AA) [30] and 4V51 (chainID: AA) [31], respectively. As the secondary structures of these 16S rRNAs, base-pairs were selected according to the "base-pair hydrogen bonding classification" provided by NDB. Specifically, base-pairs were classified as 1 in the Leontis–Westhof classification [32] and either 19, 20, or 28 in the Saenger classification [33]. Base-to-base correspondence between the primary sequence and its secondary structure (derived from the three-dimensional structure in which several residues are missing) was estimated using Needleman–Wunsch alignment [34].

The energy parameter rna_turner2004.par included in the Vienna RNA package [27] version 2.4.9 was used. However, the source code itself of Vienna was not used. The algorithms were implemented by the authors, except for parameter file reading, which is based on ParasoR's implementation [13].

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1186/s12859-020-3535-5.

Additional file 1: Supplementary PDF file.

Abbreviations

BPPM: Base-pairing probability matrix; DFT: Discrete fourier transform; DP: Dynamic programming; FFT: Fast fourier transform; NDB: Nucleic acid database; ncRNA: Non-coding RNA; rRNA: Ribosomal RNA

Acknowledgements

The authors deeply thank the developers of the RintW, ParasoR, and ViennaRNA Package implementations, which were necessary for the developed software tool. The authors also deeply thank Dr. Shun Sakuraba, who suggested the possible application of maximum-span constraint and numerical problems in previous research. The authors also thank members of the Artificial Intelligence Research Center, AIST, and Hisanori Kiryu's laboratory for useful discussions. Computations were partially performed on the NIG supercomputer at the ROIS National Institute of Genetics.

Authors' contributions

H.T. and K.A. conceived of the idea and designed the model. H.T. developed RintC. H.T. and J.I. performed experiments. All authors wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work was supported by MEXT/JSPS KAKENHI Grant Numbers JP16H02484 and JP16H06279 to K.A. for computational experiment and JP16K16143 to J.I. for survey of related research, and JST CREST Grant Number JPMJCR18S1 to K.A. for place fee. Those funding bodies only provided financial support.They did not played any roles in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

The source code for RintC is available on the following website. https://github.com/eukaryo/rintc The S151Rfam dataset is available on the following website. http://contra.stanford.edu/contrafold/download.html

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 7 June 2019 Accepted: 5 May 2020
Published online: 24 May 2020

References

1. McCaskill JS. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers. 1990;29(6-7):1105–19. https://doi.org/10.1002/bip.360290621.
2. Freyhult E, Moulton V, Clote P. RNAbor: a web server for RNA structural neighbors. Nucleic Acids Res. 2007;35(Web Server):305–9. https://doi.org/10.1093/nar/gkm255.
3. Lorenz R, Flamm C, Hofacker IL. 2D projections of RNA folding landscapes; 2009. p. 11–20.
4. Newberg LA, Lawrence CE. Exact Calculation of Distributions on Integers, with Application to Sequence Alignment. J Comput Biol. 2009;16(1):1–18. https://doi.org/10.1089/cmb.2008.0137.
5. Senter E, Sheikh S, Dotu I, Ponty Y, Clote P. Using the Fast Fourier Transform to Accelerate the Computational Search for RNA Conformational Switches. PLoS ONE. 2012;7(12):50506. https://doi.org/10.1371/journal.pone.0050506.
6. Mori R, Hamada M, Asai K. Efficient calculation of exact probability distributions of integer features on RNA secondary structures. BMC Genomics. 2014;15(Suppl 10):6. https://doi.org/10.1186/1471-2164-15-S10-S6.
7. Hagio T, Sakuraba S, Iwakiri J, Mori R, Asai K. Capturing alternative secondary structures of RNA by decomposition of base-pairing probabilities. BMC Bioinformatics. 2018;19(S1):38. https://doi.org/10.1186/s12859-018-2018-4.
8. Michálik J, Touzet H, Ponty Y. Efficient approximations of RNA kinetics landscape using non-redundant sampling. Bioinformatics. 2017;33(14):283–92. https://doi.org/10.1093/bioinformatics/btx269.
9. Senter E, Dotu I, Clote P. RNA folding pathways and kinetics using 2D energy landscapes. J Math Biol. 2015;70(1-2): 173–96. https://doi.org/10.1007/s00285-014-0760-4.
10. Sunaga T. Theory of an interval algebra and its application to numerical analysis. Jpn J Ind Appl Math. 1958;26(2-3): 125–43. https://doi.org/10.1007/BF03186528.
11. Petkovic M, Petković M, Petkovic MS, Petkovic LD. Complex Interval Arithmetic and Its Applications. Mathematical Research: Wiley; 1998. https://books.google.co.jp/books?id=Vtqk6WgttzcC.
12. Kiryu H, Kin T, Asai K. Rfold: an exact algorithm for computing local base pairing probabilities. Bioinformatics. 2008;24(3):367–73. https://doi.org/10.1093/bioinformatics/btm591.
13. Kawaguchi R, Kiryu H. Parallel computation of genome-scale RNA secondary structure to detect structural constraints on human genome. BMC Bioinformatics. 2016;17(1):203. https://doi.org/10.1186/s12859-016-1067-9.
14. Mathews DH, Disney MD, Childs JL, Schroeder SJ, Zuker M, Turner DH. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc Natl Acad Sci U S A. 2004;101(19):7287–92. https://doi.org/10.1073/pnas.0401799101.
15. Do CB, Woods DA, Batzoglou S. CONTRAfold: RNA secondary structure prediction without physics-based models. Bioinformatics. 2006;22(14):90–8. https://doi.org/10.1093/bioinformatics/btl246.
16. Durbin R, Eddy S, Krogh A, Mitchison G. Biological Sequence Analysis. Cambridge University Press; 1998. https://doi.org/10.1017/CBO9780511790492.
17. Hamada M, Kiryu H, Sato K, Mituyama T, Asai K. Prediction of RNA secondary structure using generalized centroid estimators. Bioinformatics. 2009;25(4):465–73. https://doi.org/10.1093/bioinformatics/btn601.
18. Webb-Robertson B-JM, McCue LA, Lawrence CE. Measuring Global Credibility with Application to Local Sequence Alignment. PLoS Comput Biol. 2008;4(5):1000077. https://doi.org/10.1371/journal.pcbi.1000077.
19. Tehei M, Franzetti B, Madern D, Ginzburg M, Ginzburg BZ, Giudici-Orticoni M-T, Bruschi M, Zaccai G. Adaptation to extreme environments: macromolecular dynamics in bacteria compared in vivo by neutron scattering,. EMBO Rep. 2004;5(1):66–70. https://doi.org/10.1038/sj.embor.7400049.
20. Baird N, Srividya N, Krasilnikov AS, Mondragon A, Sosnick TR, Pan T. Structural basis for altering the stability of homologous RNAs from a mesophilic and a thermophilic bacterium. RNA. 2006;12(4):598–606. https://doi.org/10.1261/rna.2186506.
21. Jegousse C, Yang Y, Zhan J, Wang J, Zhou Y. Structural signatures of thermal adaptation of bacterial ribosomal RNA, transfer RNA, and messenger RNA. PLoS ONE. 2017;12(9):0184722. https://doi.org/10.1371/journal.pone.0184722.
22. Mallik S, Kundu S. A comparison of structural and evolutionary attributes of escherichia coli and thermus thermophilus small ribosomal subunits: Signatures of thermal adaptation. PLoS One. 2013;8(8):69898. https://doi.org/10.1371/journal.pone.0069898.
23. Hofacker IL, Priwitzer B, Stadler PF. Prediction of locally stable RNA secondary structures for genome-wide surveys. Bioinformatics. 2004;20(2):186–90. https://doi.org/10.1093/bioinformatics/btg388.
24. Bernhart SH, Hofacker IL, Stadler PF. Local RNA base pairing probabilities in large sequences. Bioinformatics. 2006;22(5):614–5. https://doi.org/10.1093/bioinformatics/btk014.
25. Lange SJ, Maticzka D, Möhl M, Gagnon JN, Brown CM, Backofen R. Global or local? Predicting secondary structure and accessibility in mRNAs. Nucleic Acids Res. 2012;40(12):5215–26. https://doi.org/10.1093/nar/gks181.
26. Kashiwagi M. kv - a C++ Library for Verified Numerical Computation. 2018. http://verifiedby.me/kv/index-e.html. Accessed 10 Oct 2018.
27. Lorenz R, Bernhart SH, Höner zu Siederdissen C, Tafer H, Flamm C, Stadler PF, Hofacker IL. ViennaRNA Package 2.0. Algoritm Mol Biol. 2011;6(1):26. https://doi.org/10.1186/1748-7188-6-26.

28. Coimbatore Narayanan B, Westbrook J, Ghosh S, Petrov AI, Sweeney B, Zirbel CL, Leontis NB, Berman HM. The Nucleic Acid Database: new features and capabilities. Nucleic Acids Res. 2014;42(D1):114–22. https://doi.org/10.1093/nar/gkt980.
29. Berman HM, Olson WK, Beveridge DL, Westbrook J, Gelbin A, Demeny T, Hsieh SH, Srinivasan AR, Schneider B. The nucleic acid database. A comprehensive relational database of three-dimensional structures of nucleic acids. Biophys J. 1992;63(3):751–9. https://doi.org/10.1016/S0006-3495(92)81649-1.
30. Dunkle JA, Wang L, Feldman MB, Pulk A, Chen VB, Kapral GJ, Noeske J, Richardson JS, Blanchard SC, Cate JHD. Structures of the bacterial ribosome in classical and hybrid states of tRNA binding. Sci N Y. 2011;332(6032):981–4. https://doi.org/10.1126/science.1202692.
31. Selmer M, Dunham CM, Murphy FV, Weixlbaumer A, Petry S, Kelley AC, Weir JR, Ramakrishnan V. Structure of the 70S ribosome complexed with mRNA and tRNA,. Sci N Y. 2006;313(5795):1935–42. https://doi.org/10.1126/science.1131127.
32. Leontis NB, Westhof E. Geometric nomenclature and classification of RNA base pairs,. RNA N Y. 2001;7(4):499–512.
33. Saenger W. Principles of Nucleic Acid Structure. Springer Advanced Texts in Chemistry. New York, NY: Springer; 1984. https://doi.org/10.1007/978-1-4612-5190-3. http://link.springer.com/10.1007/978-1-4612-5190-3.
34. Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol. 1970;48(3):443–53. https://doi.org/10.1016/0022-2836(70)90057-4.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.