博士論文

An Efficient Algorithm for Distributions of Various Feature Values in Bioinformatics Problems

(情報生命科学の諸問題における特徴量分布の効率的計算方法)

森 遼太

Thesis for Doctoral Degree Overview Year 2016

An Efficient Algorithm for Distributions of Various Feature Values in Bioinformatics Problems

Many problems in bioinformatics can be classified as prediction or estimation problems, such as sequence alignment[1], RNA or protein high-dimensional structure estimation[2][3], gene coding region prediction[4], phylogenetic tree estimation[5], RNA/protein - RNA/protein interaction prediction[6][7][8], or chromatin state estimation[9].

Most of these algorithms attempt to select one specific solution from the pool of all possible candidates, a process called point estimation. However, all of these point estimation strategies have latent problems, in common, including unclear reliability, omission of information on thermal fluctuations, or omission of significant sub-optimal solutions[10]. These problems rise from the near zero probability of each solution because of the massive amount of candidates[11] and are unavoidable irrespective of the estimator or algorithm chosen.

In this study, we propose an efficient method for calculating the probability distributions of a feature value which is assigned to each candidate. If the feature value belongs to the group of bounded integers, we can construct an algorithm to obtain the exact distribution. In contrast, if the feature value belongs to the bounded real numbers, an approximation must be introduced. Calculating the distribution of feature values enables us to undertake biological interpretation without identifying a specific uncertain solution.

The structure of this paper is as follows. First, we introduce some point estimating problems with our previous studies. Second, we explain the proposed method used to calculate the distributions of the integer, real number, and vector features respectively, and give some applications to the feature values of RNA secondary structure to concretely illustrate the procedure. Third, we examine the performance of our algorithms and analyze RNA sequences using them. In this section, a "credibility limit" is defined and used to evaluate the reliability of the estimated structure. Finally, we give our conclusions and plan for future work.

Keywords: ambiguity of point estimation, feature value distribution

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

In this section, we briefly introduce our proposed technique and set the background to the study.

First, we explain the basic concept and give the definition of point estimation, illustrated by examples from our previous work. Second, we point out the unavoidable uncertainty which attaches to traditional point estimating strategies and summarize current techniques used to address these uncertainties. Finally, the positioning of this study and the definition of the problem are described.

1.0.1 Point Estimation and Basic Assumptions

Point estimation is the main strategy for estimation problems used in bioinformatics to identify a single solution from the ensemble of candidate solutions, under some specific criteria. Two representative estimators are used as criteria: the maximum likelihood estimator (ML estimator) and the maximum expected gain estimator (MEG estimator)[13][14].

This study assumes that problems conform to two conditions, i.e., that the number of candidate solutions is finite even if it is very large, and that there is some (pseudo-)energy function E, which provides the distribution of candidate ensemble $p(\theta)$, given by the following canonical distribution:

$$p(\theta) = \frac{1}{Z} e^{\frac{-E(\theta)}{k_B T}} \tag{1.1}$$

$$Z = \sum_{\theta \in \Theta} e^{\frac{-E(\theta)}{k_B T}}, \tag{1.2}$$

where Θ is a set of all solution candidates, k_B is the Boltzmann constant, T is a temperature constant, and Z is the partition function, which is a summation of the Boltzmann factor $e^{-E(\theta)/k_B T}$ among all possible solutions.

The ML estimator $\hat{\theta}_{ML}$ chooses a solution whose likelihood function is maximized by:

$$\hat{\theta}_{ML} = \arg\max_{\theta\in\Theta} p(\theta|D), \tag{1.3}$$

where D is a given data set. The ML estimator can be identified with the minimum free energy estimator, since $\arg\max_{\theta\in\Theta} p(\theta|D)$ is equivalent to $\arg\min_{\theta\in\Theta} E(\theta|D)$.

In contrast, the MEG estimator $\hat{\theta}_{MEG}$ chooses a solution by maximizing the expectation of gain function G, as follows:

$$\hat{\theta}_{MEG} = \arg\max_{\theta\in\Theta} \sum_{\theta'\in\Theta} G(\theta,\theta')p(\theta'|D). \tag{1.4}$$

If the gain function is an accuracy measure such as sensitivity, specificity, PPV, or F-score, the MEG estimator is also called a maximum expected accuracy estimator (MEA estimator). We can assume that the ML estimator is a special case of the MEG estimator if the Kronecker delta is used as the gain function G_{ML}:

$$\begin{aligned}
\hat{\theta}_{ML} &= \arg\max_{\theta\in\Theta} p(\theta|D) && (1.5)\\
&= \arg\max_{\theta\in\Theta} \sum_{\theta'\in\Theta} \delta_{\theta\theta'} p(\theta'|D) && (1.6)\\
&= \arg\max_{\theta\in\Theta} \sum_{\theta'\in\Theta} G_{ML}(\theta,\theta')p(\theta'|D) && (1.7)
\end{aligned}$$

$$\delta_{ab} = \begin{cases} 1 & (a=b) \\ 0 & (otherwise). \end{cases} \tag{1.8}$$

ML and MEG estimators are common criteria used for point estimation, though a range of strategies are available for constructing estimators, such as consistency with experimental data. The choice of estimator depends on the particular situation, and this is not further discussed. In the next three sections, we introduce concrete bioinformatic point estimation problems from our previous work which are relevant to point estimation.

(1) Sequence Alignment

Sequence alignment is a problem that arises when aligning DNA, RNA, or protein sequences based on similarity. Sequence similarity is closely linked to functional relationships between sequences as well as to structural relationships, and is therefore utilized in a range of biological applications such as gene function and evolutionary predictions. Although there are many objectives and formularizations, from the point of the view of point estimation the sequence alignment problem is a task which requires choosing one specific alignment between given sequences from the space of all possible alignments.

We previously developed an alignment algorithm for bisulfite-converted DNA sequences[15]. Bisulfite conversion is a treatment which identifies the position of methylated cytosines in DNA fragments. Unmethylated cytosines in the bisulfite-treated sequences are converted to thymines. Accordingly, we can find the methylated positions by aligning bisulfite-treated reads with the reference genome; C-C match positions are assumed to be methylated. We accommodated a traditional alignment technique to the model which considers bisulfite conversion.

A scoring matrix S_{xd} (x is a nucleotide in the reference and d is a nucleotide in a read) can be derived in the following way:

$$\begin{aligned}
S_{xd} &= \lambda \ln(R_{xd}) && (1.9)\\
R_{xd} &= \frac{P(x, d|A)}{P(x)P(d)} && (1.10)\\
&= \frac{\sum_{y\in\{a,c,g,t\}} P(x, y|A)P(d|y)}{P(x)P(d)} && (1.11)\\
&= \frac{1}{P(x)P(d)} \cdot \sum_{y\in\{a,c,g,t\}} \frac{P(x, y|A)P(y|d)P(d)}{P(y)} && (1.12)\\
&= \sum_{y\in\{a,c,g,t\}} \frac{P(x, y|A)P(y|d)P(d)}{P(x)P(d)P(y)} && (1.13)\\
&= \sum_{y\in\{a,c,g,t\}} \frac{P(x, y|A)P(y|d)}{P(x)P(y)} && (1.14)\\
&= \sum_{y\in\{a,c,g,t\}} \left\{R_{xy}P(y|d)\right\}, && (1.15)
\end{aligned}$$

where $P(x, y|A)$ is the probability of observing x aligned to y, $P(x)$ and $P(y)$ are the probabilities of observing x and y respectively, and $P(y|d)$ is the probability that the observed d was y before sequencing and bisulfite-treatment as follows:

$$P(y|d) = \sum_{b\in\{a,c,g,t\}} P(y|b)P(b|d). \qquad (1.16)$$

Here, $P(y|b)$ is the probability that a base is y before bisulfite treatment, given a bisulfite-treated base b, and $P(b|d)$ is the probability that a base is b under base caller, calling it d.

$P(y|b)$ and $P(b|d)$ are constructed as follows:

First, $P(y|b)$ is given by:

$$\begin{aligned} P(y|b) &= \frac{P(y)P(b|y)}{P(b)} && (1.17) \\ &= \frac{P(y)P(b|y)}{\sum_{y \in \{a,c,g,t\}} P(b|y)P(y)}, && (1.18) \end{aligned}$$

where:

$$P(b|y) = A_{yb} \tag{1.19}$$

$$A = \begin{array}{c} \\ a \\ c \\ g \\ t \end{array} \begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \alpha & 0 & 1-\alpha \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \end{array}. \tag{1.20}$$

From (1.18), (1.19), and (1.20),

$$\begin{aligned} P(y|b) &= \frac{P(y)A_{yb}}{\sum_{y \in \{a,c,g,t\}} A_{yb}P(y)} && (1.21) \\ &= B_{by} && (1.22) \end{aligned}$$

$$B = \begin{array}{c} \\ a \\ c \\ g \\ t \end{array} \begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{(1-\alpha)P(c)}{(1-\alpha)P(c)+P(t)} & 0 & \frac{P(t)}{(1-\alpha)P(c)+P(t)} \end{pmatrix} \end{array}. \tag{1.23}$$

Next, $P(b|d)$ is given by:

$$P(b|d) \quad = \quad C_{db} \tag{1.24}$$

$$C = \begin{array}{c} \\ a \\ c \\ g \\ t \end{array} \begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{pmatrix} (1-\beta)+\beta P(a)' & \beta P(c)' & \beta P(g)' & \beta P(t)' \\ \beta P(a)' & (1-\beta)+\beta P(c)' & \beta P(g)' & \beta P(t)' \\ \beta P(a)' & \beta P(c)' & (1-\beta)+\beta P(g)' & \beta P(t)' \\ \beta P(a)' & \beta P(c)' & \beta P(g)' & (1-\beta)+\beta P(t)' \end{pmatrix} \end{array} \tag{1.25}$$

Here:

$$\begin{aligned} \begin{pmatrix} P(a)' \\ P(c)' \\ P(g)' \\ P(t)' \end{pmatrix} &= A^{\mathrm{T}} \begin{pmatrix} P(a) \\ P(c) \\ P(g) \\ P(t) \end{pmatrix} \\ &= \begin{pmatrix} P(a) \\ \alpha P(c) \\ P(g) \\ (1-\alpha)P(c)+P(t) \end{pmatrix}. \end{aligned} \tag{1.26}$$

Therefore, $P(y|d)$ is as follows:

$$P(y|d) = M_{dy} \tag{1.27}$$

$$\begin{aligned} M &= CB \\ &= \begin{array}{c} \\ a \\ c \\ g \\ t \end{array} \begin{array}{c} \begin{array}{cccc} a & c & g & t \end{array} \\ \begin{pmatrix} \beta P(a)+(1-\beta) & \beta P(c) & \beta P(g) & \beta P(t) \\ \beta P(a) & \beta P(c)+(1-\beta) & \beta P(g) & \beta P(t) \\ \beta P(a) & \beta P(c) & \beta P(g)+(1-\beta) & \beta P(t) \\ \beta P(a) & \beta P(c)+\frac{(1-\alpha)(1-\beta)P(c)}{(1-\alpha)P(c)+P(t)} & \beta P(g) & \beta P(t)+\frac{(1-\beta)P(t)}{(1-\alpha)P(c)+P(t)} \end{pmatrix} \end{array}. \end{aligned} \tag{1.28}$$

Parameters α and β can be obtained by:

$$\alpha \quad = \quad \frac{\text{frequency of C in reads}}{\text{frequency of C in reference}} \tag{1.29}$$

$$\beta \quad = \quad 10^{-q/10} \quad (q : \textit{phred score}). \tag{1.30}$$

This is the basic scheme for adapting bisulfite conversion to the traditional model.

However, the simplification made in (1.29) is too naive, because it is unreasonable to assign the same methylation rate α to all cytosines in the genome. Accordingly, we suggest identifying cytosines by the profile of the query sequence. For example, in a mammalian genome, cytosine methylation occurs approximately three times as often in CpG as CpH[16]. In contrast, plants have methylate cytosines in CpG and those methylate cytosines are also in CpHpG[17]. If we adopt a range of different values of α as α_1, α_2, and α_3 to the Cs in CpGpN, CpHpG, and CpHpH respectively, our model can estimate $\tilde{\alpha}$ as follows:

$$\begin{aligned}
\tilde{\alpha} &= \mathbb{E}\left[\alpha|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}\right] && (1.31)\\
&= \sum_{i=1}^{3} \alpha_i P(m_i|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}) && (1.32)\\
&= \sum_{i=1}^{3} \alpha_i \frac{P(\mathrm{XpN_1pN_2}|m_i, \mathrm{C} \to \mathrm{X})P(m_i|\mathrm{C} \to \mathrm{X})}{P(\mathrm{XpN_1pN_2}|\mathrm{C} \to \mathrm{X})} && (1.33)\\
&= \sum_{i=1}^{3} \alpha_i \frac{P(\mathrm{XpN_1pN_2}|m_i, \mathrm{C} \to \mathrm{X})P(m_i|\mathrm{C} \to \mathrm{X})}{\sum_{j=1}^{3} P(\mathrm{XpN_1pN_2}|m_j, \mathrm{C} \to \mathrm{X})P(m_j|\mathrm{C} \to \mathrm{X})}. && (1.34)
\end{aligned}$$

Here, $\mathbb{E}\left[\alpha|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}\right]$ is the expected probability of methylation α at position X under the base caller calling $\mathrm{N_1}$ and $\mathrm{N_2}$ and given that X was C before bisulfite treatment. For each $i = 1\ to\ 3$, $P(m_i|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X})$ is the probability that $\mathrm{XpN_1pN_2}$ was converted to CpGpN, CpHpG, and CpHpH respectively by bisulfite treatment under the same conditions. For each $i = 1\ to\ 3$, $P(m_i|\mathrm{C} \to \mathrm{X})$ is the probability of the occurrence of sequence type CpGpN, CpHpG, and CpHpH before bisulfite treatment. In other words, it represents the proportion of CpGpN, CpHpG, and CpHpH in the genome sequence. $P(m_i|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X})$ can be calculated by:

$$\begin{aligned}
P(m_1|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}) &= M'_{1\mathrm{N_1G}} && (1.35)\\
P(m_2|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}) &= M'_{1\mathrm{N_1H}} \cdot M'_{2\mathrm{N_2G}} && (1.36)\\
P(m_3|\mathrm{XpN_1pN_2}, \mathrm{C} \to \mathrm{X}) &= M'_{1\mathrm{N_1H}} \cdot M'_{2\mathrm{N_2H}}. && (1.37)
\end{aligned}$$

From (1.28),

$$\begin{aligned}
M'_i &= \begin{array}{c c} & \begin{array}{cc} H & G \end{array} \\ \begin{array}{c} H \\ \\ G \end{array} & \begin{pmatrix} 1 - \beta_i P(g) & \beta_i P(g) \\ & \\ \beta_i \overline{P(g)} & 1 - \beta_i \overline{P(g)} \end{pmatrix} \end{array} && (1.38)\\
\overline{P(g)} &= 1 - P(g) && (1.39)\\
\beta_i &= the\ corresponding\ \beta\ for\ \mathrm{N_i}. && (1.40)
\end{aligned}$$

Calculation of $\tilde{\alpha}$ is required if and only if X = T (see (1.28)).

Detailed results and discussions are given in [15] but are secondary to the themes of this study and are therefore omitted here.

(2) RNA Secondary Structure Estimation

RNA secondary structure estimation is a problem which arises when estimating internal hydrogen bonds in RNA without considering the steric effect[18].

RNA conformation is a significant issue in biological functions. Ideally, a tertiary structure should be the best from analyzing the behavior of RNA. However, the analysis of tertiary structure for anything other than very short RNAs has high costs and presents technical difficulties[19]. The secondary structure is also an important feature when identifying the functions of RNA, and high-throughput methods for determining the secondary structures are becoming more widespread, which is also increasing the importance of computational analyses of RNA sequences.

Estimation strategies can be classified into two categories: "*in silico*" and "composite". *In silico* strategies only require the use of a computer. More than 50 tools are available to predict a structure[20], and most algorithms base the prediction on the free energy of the structures. The composite strategy, in contrast, uses both experimental data and computation. For example, in the SHAPE method, experiments are used to extract flexible regions as a SHAPE profile, and the structure is predicted by reference to the profile[21][22]. Note that even if when using experimental approaches, the secondary structure cannot be directly observed.

Regardless of the strategy used, from the point of the view of point estimation the RNA secondary structure problem requires one specific secondary structure of the given RNA sequence to be chosen from the space of all possible secondary structures.

A previously developed algorithm allowed point-estimates to be made of an RNA secondary structure based on nuclear magnetic resonance (NMR) spectroscopy (manuscript in preparation). NMR spectroscopy is a widely used technology for exploring the intermolecular interactions and structure of chemicals[23]. Two-dimensional NMR spectroscopy such as correlation spectroscopy (COSY)[24], heteronuclear single-quantum correlation spectroscopy (HSQC)[25], or nuclear Overhauser effect spectroscopy (NOESY)[26] is used to solve the three-dimensional structure of molecules, including short RNAs. Even with two-dimensional NMR spectroscopy however, calculating three-dimensional structure is challenging, and the exact number of C-G, A-U, and G-U hydrogen bonds can be obtained with comparative ease. Our algorithm point-estimates the most stable minimum free energy RNA secondary structure in the set of structures which is consistent with the number of hydrogen bonds derived by NMR spectroscopy. We modified IPknot[27], which is a fast and accurate software for estimating RNA secondary structure without excluding unusual hydrogen bonds called pseudo-knots. IPknot utilizes integer programming to narrow the candidate structure space. By adding the following restriction to the original IPknot model, we were able to obtain the secondary structure which is consistent with the NMR spectroscopy:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \mathbf{B}_{ij} = \mathbf{B}^{NMR}, \tag{1.41}$$

where:

$$\mathbf{B}_{ij} = \begin{pmatrix} \delta_{x_i C}\delta_{x_j G} + \delta_{x_i G}\delta_{x_j C} \\ \delta_{x_i A}\delta_{x_j U} + \delta_{x_i U}\delta_{x_j A} \\ \delta_{x_i G}\delta_{x_j U} + \delta_{x_i U}\delta_{x_j G} \end{pmatrix} \tag{1.42}$$

$$\mathbf{B}^{NMR} = \begin{pmatrix} \text{the number of C-G base pairs estimated by NMR} \\ \text{the number of A-U base pairs estimated by NMR} \\ \text{the number of G-U base pairs estimated by NMR} \end{pmatrix} \tag{1.43}$$

$$x_i: \text{ i-th base of the RNA sequence (A, C, G, or U).} \tag{1.44}$$

(3) Other Problems

Many other estimation problems arise in bioinformatics, including gene coding region estimation from an input genome sequence, phylogenetic tree estimation from genomes of multiple species, estimation of RNA/protein - RNA/protein interactions (presence/absence or contact position, and chromatin state estimation. These have the common characteristic that they can be considered as problems of point estimation from some discrete solution space.

1.0.2 Unavoidable Uncertainty of Traditional Point Estimating Strategies

The point estimation problems described above have unavoidable issues when applied to biological investigation. The most serious problem is that the reliability is unclear because of the enormous number of candidate solutions, which can be approximated as follows[28]:

1. sequence alignment between length n DNAs:

$$\frac{(1+\sqrt{2})^{2n+1}}{\sqrt{n}} \tag{1.45}$$

2. length n RNA secondary structure:

$$\sqrt{\frac{15+7\sqrt{5}}{8\pi n^3}}\left(\frac{3+\sqrt{5}}{2}\right)^n \tag{1.46}$$

The number of candidates typically increases exponentially or hyper-exponentially as the problem size increases, and the probability that a predicted solution is exactly true tends to be extremely low, despite this being the only indicator of reliability for methods based on ML estimation. Even if the best solution in the solution space is found, all other solutions might be concentrated far away from the ML solution. This may cause misunderstanding of the biological interpretation if only a single unlikely solution is considered (Figure 1.1).

In contrast, an MEA solution is a centroid-like solution weighted by an accuracy measure, so that a well-designed MEA estimator is effective when the distribution of the solution is mono-modal and relatively dense. However, a problem with the MEA-based approach is that we cannot assess the credibility of the MEA solution. The probability assigned to an MEA solution is generally lower than that assigned to an ML solution. We cannot directly observe the level of concentration of probability around the MEA solution. In the extreme case, there might be almost no probability density when the candidate ensemble has a multi-modal distribution (Figure 1.2)[29]. Here the blue and purple points are sampled points in the ensemble, and the red point may be returned by some prediction algorithms as an MEA solution. Uncertainty is unavoidable when using point estimation.

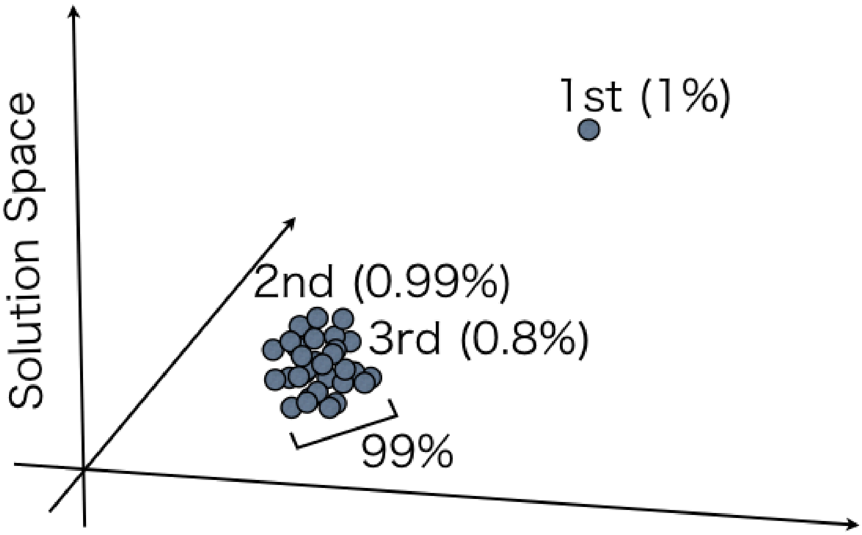
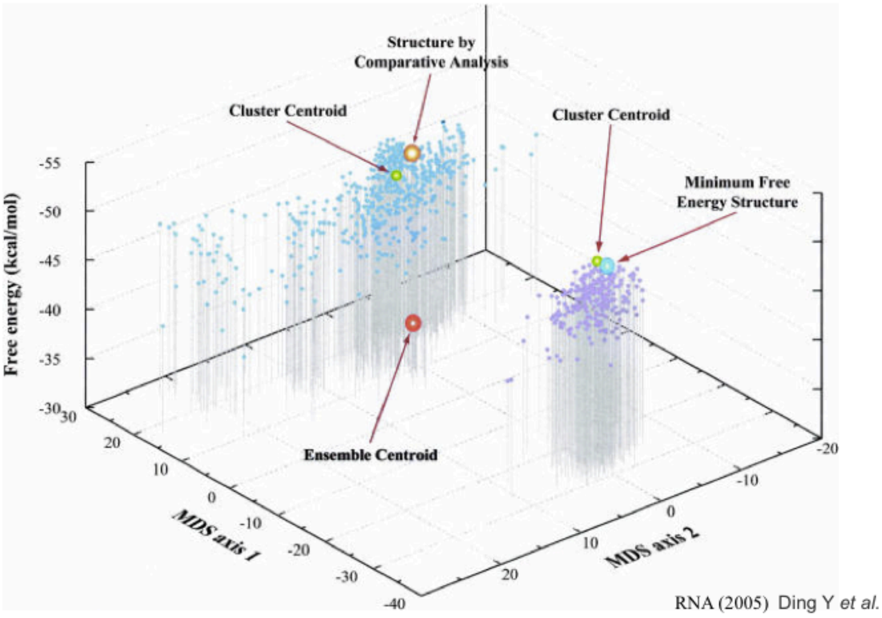


Figure 1.1: A potential case in which the best solution is far from all other solutions in the solution space.



RNA (2005) Ding Y *et al.*

Figure 1.2: A concrete case in which existing algorithms return solutions which seem unsatisfactory.

1.0.3 Current Techniques Which Treat the Whole Distribution and Their Problems

To address the limitations of point estimation, we need alternative methods which consider the whole distribution. The limitations would disappear if we were able to calculate the existence probability for all candidate solutions and then interpret the distribution, but this is unrealistic given the high dimensionality of the discrete space and the computational complexity. For example, if we adopt the Hamming distance of the base-pairing positions to define the distance between the length n RNA secondary structures, the dimension of the structure space becomes $n(n-1)/2$.

Alternatively, probabilistic sampling techniques can be used to calculate sub-optimal solutions. Those methods, however, require sample sizes that increase hugely as the problem size increases. This makes it almost impossible to guarantee that the output will properly reflect the true values when using probabilistic sampling-based techniques.

1.0.4 Positioning of This Study

In this study, we propose an alternative method for interpreting the deep features of the solution by considering the whole distribution. Our method calculates the distribution of some feature value or score S which is assigned to each solution. This approach allows the feature to be addressed directly without fixing a single uncertain solution. Here S has restrictions. S is a bounded integer or real variable function which can be obtained by a DP algorithm and which includes a structure of calculations for the partition function. A concrete example is given in Figure 1.3. Here, we selected the Hamming distance from a specific RNA secondary structure as S. S can be calculated by the same DP format with its partition function. The distribution of S provides information on how much probability is concentrated around the selected structure. The definition of the Hamming distance is given in a later section.

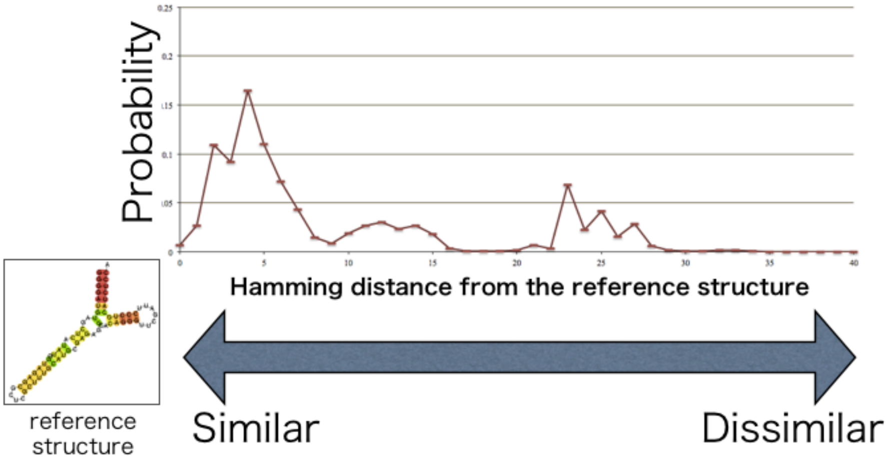


Figure 1.3: An example of feature score S.

Our algorithm can be roughly classified into two categories: when S is restricted to a bounded integer, and when it is not. If S is restricted to a bounded integer, a fast and exact algorithm by adopting Discrete Fourier Transform (DFT). Although a similar idea has been suggested for acceleration by distributed processing in the field of sequence alignment [30], our approach offers order level improvement as well as acceleration by distributed processing.

If S is not restricted to integer values, scores are allocated to finite bins, and the probability is calculated for each bin. Our proposed score-distributing technique achieves fast and accurate approximation.

The detailed algorithm and concrete applications to the distribution of the features of RNA secondary structure are discussed in the methods section. Analyses and performance by the above implementation are in the results section. These approaches provide deeper understandings about distributions of solution, and profit our biological discussion based on bioinformatics analyses.

Chapter 2

METHODS

In this section, we show the general procedure used to construct score-calculating algorithms. First, basic definitions are introduced. Next, we describe two strategies for obtaining an integer feature distribution: the polynomial and complex number strategies. The polynomial strategy is a naive implementation for this task, while the complex number strategy achieves time and space reduction. Third, we describe a real feature value case, in which it is difficult to obtain an exact distribution of the feature. Bins and a windows function which distributes values to bins are used to calculate an approximate distribution. Finally, we extend the approach to a higher dimensional distribution, the probability distribution of a feature value vector.

2.1 Definitions and Preliminaries

We first introduce some definitions. In this study, RNA secondary structure estimation problems are used to exemplify concrete algorithms, and we therefore also show here the basic assumptions used for estimating RNA secondary structure.

2.1.1 Definition of the Feature Score Probability Distribution

Let us assume that s represents a mapping from $x \in U$ to an integer score $s(x) \in \mathbb{Z}$:

$$\begin{array}{rccc} s: & U & \longrightarrow & \mathbb{Z} \\ & \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ & x & \longmapsto & s(x) \end{array}. \tag{2.1}$$

The integer score distribution is defined as the probability distribution $p(s)$ of $s(x)$ derived from the probability distribution $p(x)$ of x:

$$p(s) = \sum_{\{x|s=s(x)\}} p(x). \tag{2.2}$$

On the other hand, if s represents a mapping from $x \in U$ to a real score $s(x) \in \mathbb{R}$, then:

$$\begin{array}{rccc} s: & U & \longrightarrow & \mathbb{R} \\ & \rotatebox{90}{$\in$} & & \rotatebox{90}{$\in$} \\ & x & \longmapsto & s(x) \end{array}. \tag{2.3}$$

The real score distribution is defined as the probability density distribution $p(s)$. We assume that x is a discrete element, and thus the probability $p(s_l \leq s \leq s_u)$ is

$$p(s_l \leq s \leq s_u) = \int_{s_l}^{s_u} p(s)ds \tag{2.4}$$

$$= \sum_{\{x|s_l \leq s(x) \leq s_u\}} p(x). \tag{2.5}$$

In the case of RNA secondary structures, U is the space of all possible secondary structures for a given RNA sequence, and an integer or real score $s(x)$ represents a feature or a property assigned to each structure x. In this study, we discuss the efficient computation of integer and real feature value/score distributions both in general and in the specific case of RNA secondary structures. Our proposed method efficiently computes the exact distribution when $p(s)$ and the partition function of $p(x)$ can be calculated by dynamic programming algorithms sharing the same form.

2.1.2 Dynamic Programming

Dynamic programming is a computing technique used to arrive at a total calculation by dividing the problem up and using sequential calculations. Two strategies can be used to implement dynamic programming: bottom up and top down strategies.

We use algorithms for calculating the Fibonacci sequence to exemplify these strategies. The Fibonacci sequence $F[n]$ is defined by the following recursion:

$$F[1] = F[2] = 1 \tag{2.6}$$

$$F[n] = F[n-1] + F[n-2] \quad (3 \leq n). \tag{2.7}$$

(1) Naive Implementation for the Fibonacci Sequence

The simplest implantation for $F[n]$ is Algorithm 1.

Algorithm 1 Naive Implementation for Fibonacci Sequence F[n]

```
if ($n == 1 || n == 2$) then
    return 1
else
    return $F[n] + F[n-1]$
end if
```

This pseudo-code looks intuitive; however, this procedure imposes a large calculation burden. For example, $F[10]$ is expanded into the following:

$$\begin{aligned}
F[10] =& F[9] + F[8] && (2.8)\\
=& (F[8] + F[7]) + (F[7] + F[6]) && (2.9)\\
=& ((F[7] + F[6]) + (F[6] + F[5])) + ((F[6] + F[5]) + (F[5] + F[4])) && (2.10)\\
=& ((F[6] + F[5]) + (F[5] + F[4])) + ((F[5] + F[4]) + (F[4] + F[3])) && (2.11)\\
&+ ((F[5] + F[4]) + (F[4] + F[3])) + ((F[4] + F[3]) + (F[3] + F[2])) && (2.12)\\
&\vdots\\
&\vdots\\
&\vdots\\
=& F[2] + F[1] + \cdots\cdots\cdots + F[2] + F[1] + F[2] && (2.13)\\
=& 55. && (2.14)
\end{aligned}$$

It is known that this naive strategy requires $O(1.619^n)$ calculations, so that 30 billion calculations are needed for $F[50]$. This is clearly impractical.

(2) Adopting Dynamic Programming Technique for the Fibonacci Sequence

In contrast, a dynamic programming technique reduces the number of calculations required by reusing calculations. Algorithm 2 uses a bottom up strategy. In this strategy, $F[10]$ is calculated as follows:

$$
\begin{aligned}
F[1] =& 1 \quad (2.15)\\
F[2] =& 1 \quad (2.16)\\
F[3] =& F[2] + F[1] = 2 \quad (2.17)\\
F[4] =& F[3] + F[2] = 3 \quad (2.18)\\
& \vdots\\
F[10] =& F[9] + F[8] = 55. \quad (2.19)
\end{aligned}
$$

Algorithm 2 Bottom up Implementation for Fibonacci Sequence F[n]

```
$F[1] = F[2] = 1$
**for** $i = 3; i \leq n; ++i$ **do**
    $F[i] = F[i-1] + F[i-2]$
**end for**
return $F[n]$
```

Algorithm 3 uses a top down, or memorization strategy, which obviates the need to repeat the same calculations:

$$
\begin{aligned}
F[10] =& F[9] + F[8] \quad (2.20)\\
=& ((F[7] + F[6]) + F[7]) + F[8] \quad (2.21)\\
=& (((F[6] + F[5]) + F[6]) + F[7]) + F[8] \quad (2.22)\\
& \vdots\\
=& (((8 + 5) + 8) + F[7]) + F[8] \quad (2.23)\\
=& ((13 + 8) + 13) + F[8] \quad (2.24)\\
=& (21 + 13) + 21 \quad (2.25)\\
=& 55. \quad (2.26)
\end{aligned}
$$

Both strategies require only $O(n)$ calculations. We adopt the bottom up strategy in the algorithms presented in this paper.

Algorithm 3 Top down Implementation for Fibonacci Sequence F[n]

```
(prepare $memo[] = \{0, 1, 1, -1, -1, \cdots, -1\}$)
if $memo[n]! = -1$ then
    $memo[n] = F[n-1] + F[n-2]$
end if
return $memo[n]$
```

2.1.3 Abstract Forms of Dynamic Programming Structure in Bioinformatics

For a certain class of problems in bioinformatics, including sequence alignment, the partition function of the objective distribution can be calculated abstractly by Algorithm 4. Here, Z is the partition function given in equation (1.2). $Z[]$ is a scalar array of length N representing the partition function of the problem size N, whose components are aligned in the computing order required for dynamic programming, while $t(k|i)$ is a quantity proportional to the probability of the transition from state i to state k, which is normally quite sparse in values.

Algorithm 4 An Abstract Form of Calculating the Partition Function

```
$Z[0] = 1$
for $k = 1$ to $N$ do
    $Z[k] = \sum_{i=0}^{k-1} Z[i]t(k|i)$
end for
$Z = Z[N]$
```

In the case of the partition function for the distribution of RNA secondary structure, however, Algorithm 4 is not sufficient, and Algorithm 5 is used istead. The difference between Algorithm 4 and Algorithm 5 is the addition of a term $\sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z[i]Z[j]t(k|i,j)$, where $t(k|i,j)$ is proportional to the probability of a transition from state (i, j) to k. This term represents the effect of combining $Z[]$s. The concrete dynamic programming structure for the partition function of the RNA secondary structure distribution is given in the next section.

Algorithm 5 An Abstract Form of Calculating the Partition Function (RNA secondary structure)

```
$Z[0] = 1$
for $k = 1$ to $N$ do
    $Z[k] = \sum_{i=0}^{k-1} Z[i]t(k|i) + \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z[i]Z[j]t(k|i,j)$
end for
$Z = Z[N]$
```

2.1.4 Basic Policy for Estimating RNA Secondary Structures *in silico*

To exemplify our theoretical approach we use the concrete calculation of RNA secondary structure. We therefore first show some basic assumptions used for estimating RNA secondary structure *in silico*. First, we introduce a distribution in which folding RNA follows thermodynamic laws, and we show an effective conventional algorithm for RNA folding as an archetype of our models.

(1) Canonical Distribution and Partition Function

The probability of each RNA secondary structure is calculated by the following canonical distribution:

$$p_i = \frac{1}{Z} e^{-E_i/(k_B T)} \tag{2.27}$$

$$Z = \sum_i e^{-E_i/(k_B T)}, \tag{2.28}$$

where p_i is a probability that a certain structure i exists among the whole ensemble of RNA structures, E_i is a free energy for the structure i, k_B is the Boltzmann constant, T is a temperature constant, and Z is the partition function which is a summation of Boltzmann factor $e^{-E_i/(k_B T)}$ among all possible structures. As can be seen, the free energy of each structure corresponds to its probability, so the probability can be calculated from the partition function and the free energy. Although we cannot compute the probabilities of all structures as the candidates are too numerous, we can obtain the partition function efficiently by applying a dynamic programming algorithm.

(2) McCaskill Model

The McCaskill model is a well-known application of dynamic programming to the partition function based on energy parameters [31]. The McCaskill model includes Inside and Outside algorithms for computing the base pairing probability, that the i-th and j-th bases of the RNA sequence make a base pair, but we do not employ these in our algorithm. In this model, the partition function is computed by a recursive scheme of polynomial order:

Initialization $(1 \le i \le n)$ **:**

$$Z(i,i) = 1.0 \tag{2.29}$$

$$Z^1(i,i) = Z^b(i,i) = Z^m(i,i) = Z^m(i,i-1) = Z^{m1}(i,i) = 0 \tag{2.30}$$

Recursion $(1 \le i < j \le n)$ **:**

$$Z(i,j) = 1.0 + \sum_{k=i}^{j-1} Z(i,k) Z^1(k+1,j) \tag{2.31}$$

$$Z^1(i,j) = \sum_{k=i+1}^{j} Z^b(i,k) \tag{2.32}$$

$$\begin{aligned} Z^b(i,j) &= e^{-\left[f_1(i,j)/k_BT\right]} + \sum_{k=i+1}^{j-2} \sum_{l=k+1}^{j-1} Z^b(k,l) e^{-\left[f_2(i,j,k,l)/k_BT\right]} \\ &\quad + \sum_{k=i+2}^{j-1} Z^m(i+1,k-1) Z^{m1}(k,j-1) e^{-\left[f_3(i,j)/k_BT\right]} \end{aligned} \tag{2.33}$$

$$Z^m(i,j) = \sum_{k=i}^{j-1} \left(e^{-\left[f_4(k-i)/k_BT\right]} + Z^m(i,k-1)\right) Z^{m1}(k,j) \tag{2.34}$$

$$Z^{m1}(i,j) = \sum_{k=i+1}^{j} Z^b(i,k) e^{-\left[f_4(j-k)/k_BT\right]}, \tag{2.35}$$

where $f_k(\cdot)$ $(k = 1 \cdots 4)$ are functions corresponding to the energy contribution to each state respectively, whose parameters are determined experimentally [32][33]. A more detailed explanation is given in Appendix B. The partition function Z is finally obtained as $Z(1,n)$. Although the second factor on the right side of equation (2.33) shows the procedure requiring $O(n^4)$ time, we are able to reduce this to $O(n^3)$ by taking a reasonable maximum value of the internal loop length as the threshold. This model forms the basis of our concrete implementations.

2.2 Calculating the Feature Distribution

In the next three sections, we describe the algorithms for calculating the objective feature distributions of the integer, real, and vector features, respectively.

2.3 Calculation of the Integer Feature Distribution

2.3.1 General Theory

If the feature value is restricted to integer values, a fast and exact calculating procedure is available. The basic idea is to adopt polynomials which include information on the feature value gain when calculating the partition function. We show that using complex numbers and Discrete Fourier Transform instead of polynomials reduces the calculation cost. We call the former approach the polynomial approach and the latter approach the complex number approach.

2.3.2 Polynomial Approach

In the field of sequence alignment, a method which calculates the distribution of the arbitrary integer score given to each alignment has already been proposed [30], but this scheme is also applicable to general dynamic programming applications. We show the general form given in [30] as Algorithm 6, where Z is an array of length N dynamic programming components aligned by computing order, $t(k|i)$ is proportional to the probability of a transition from state i to state k, $s(k)$ is the integer score or cost of a visit to k, and $s(i,k)$ is the integer score or cost of a transition from i to k.

Algorithm 6 General Polynomial Approach to Integer Score Distributions

```
$Z[0] = 1$
for $k = 1$ to $N$ do
    $Z[k] = x^{s(k)} \sum_{i=0}^{k-1} Z[i]t(k|i)x^{s(i,k)}$
end for
```

This is a natural expansion of Algorithm 4. In this algorithm, $Z[N]$ represents a polynomial in x whose factor of x^i is proportional to the probability of obtaining score i among all paths. We derive p_k, the probability of obtaining score k, by the following equation:

$$p_k = \frac{a_k}{\sum_{i=0}^{i=n} a_i}, \tag{2.36}$$

where:

$$Z[N] \equiv \sum_{i=0}^{n} a_i x^i. \tag{2.37}$$

Applied to the case of RNA secondary structure, Algorithm 5 is expanded in the same manner (Algorithm 7), where $s(i,j,k)$ is the integer score for a transition from state (i,j) to k, and $t(k|i,j)$ is proportional to the probability of a transition from state (i,j) to k. Of course both $t(k|i)$ and $t(k|i,j)$ are exceedingly sparse arrays.

Algorithm 7 General Polynomial Approach to Integer Score Distributions (for RNA secondary structure)

```
1: $Z[0] = 1$
2: **for** $k = 1$ to $N$ **do**
3:     $Z[k] = \sum_{i=0}^{k-1} Z[i]t(k|i)x^{s(i,k)} + \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z[i]Z[j]t(k|i,j)x^{s(i,j,k)}$
4: **end for**
```

2.3.3 Complex Number Approach

Following [30], distributed processing is available for Algorithm 6 if we apply DFT. Algorithm 7 can be given a reduced calculation order as well as decentralized. Our polynomial formulation includes the product of the n-th order of two polynomials, which requires $O(n^2)$. Applying DFT allows it to be calculated by $O(1)$ since we employ complex numbers on the unit circle instead of polynomials as Z.

We now explain how DFT can be used for the calculation. In the following discussion, feature value s is satisfied by $0 \le s \le S_{max}$ for brevity. The following equations for p_k, which give the probability of obtaining score k, are applicable when $s - k$ is an integer:

$$p_k = \sum_{\theta \in C_k} p(\theta|D) \tag{2.38}$$

$$= \frac{1}{Z} \sum_{s=0}^{S_{max}} Z_s \delta_{sk} \tag{2.39}$$

$$= \frac{1}{Z} \sum_{s=0}^{S_{max}} Z_s \sum_{r=0}^{S_{max}} \exp\left[2\pi i \frac{r(s-k)}{S_{max}+1}\right] / (S_{max}+1) \tag{2.40}$$

$$= \frac{1}{Z} \sum_{r=0}^{S_{max}} \sum_{s=0}^{S_{max}} Z_s \exp\left[2\pi i \frac{rs}{S_{max}+1}\right] \exp\left[2\pi i \frac{-rk}{S_{max}+1}\right] / (S_{max}+1), \tag{2.41}$$

where:

$$\begin{aligned} C_k &: \text{a set of candidate solutions whose feature values are } k \\ D &: \text{input data} \\ Z &: \text{partition function} \\ Z_s &: \text{subtotal of Boltzmann factors whose feature values are } s \\ \delta &: \text{Kronecker delta} \\ i &: \text{imaginary unit} \end{aligned}$$

DFT is a Fourier transform on a discrete sampling interval and is employed to improve the efficiency of a range of computational problems as well being used in frequency analysis.

DFT $\mathcal{F}$ satisfies the following equation:

$$\mathbf{z} = \mathcal{F}(\zeta), \tag{2.42}$$

where:

$$\begin{aligned} \mathbf{z} &= (z_0, z_1, \ldots, z_{S_{max}}) && (2.43) \\ \boldsymbol{\zeta} &= (\zeta_0, \zeta_1, \ldots, \zeta_{S_{max}}) && (2.44) \\ \zeta_k &= \sum_{r=0}^{S_{max}} z_r \left(\exp\left[2\pi i \frac{-rk}{S_{max}+1} \right] \right) / (S_{max}+1). && (2.45) \end{aligned}$$

Comparing equations (2.41) and (2.45), in the DFT approach we calculate:

$$z_r = \sum_{s=0}^{S_{max}} Z_s \exp\left[2\pi i \frac{r}{S_{max}+1} \right]^s \tag{2.46}$$

for each r instead of

$$\sum_{s=0}^{S_{max}} Z_s x^s. \tag{2.47}$$

We show Algorithm 7 modified by DFT approach as Algorithm 8. In this approach, we can reduce the calculation costs from $O(n^3 S_{max}^2)$ to $O(n^3 S_{max})$ for time and from $O(n^2 S_{max})$ to $O(n^2)$ for memory. In addition, each z_r can be calculated individually so we can reduce the cost to $O(n^3)$ time and $O(n^2 S_{max})$ memory by adopting maximum parallelization.

Algorithm 8 General Complex Number Approach to Integer Score Distributions (for RNA secondary structure)

```
for $r = 0$ to $S_{max}$ do
    $x = \exp\left(2\pi i \frac{r}{S_{max}+1}\right)$
    $Z_r[0] = 1$
    for $k = 1$ to $N$ do
        $Z_r[k] = \sum_{i=0}^{k-1} Z_r[i] t(k|i) x^{s(i,k)} + \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z_r[i] Z_r[j] t(k|i,j) x^{s(i,j,k)}$
    end for
end for
for $k = 0$ to $S_{max}$ do
    $p(score = k) = \left\{\sum_{r=0}^{S_{max}} Z_r[N] \exp\left(2\pi i \frac{-rk}{S_{max+1}}\right)\right\} / (S_{max} + 1)$ /*DFT*/
end for
```

2.3.4 Application to the Features of RNA Secondary Structure

We give two concrete applications of integer feature distribution. The first example is the Hamming distance from a certain structure, and the second example is the $5' - 3'$ distance.

2.3.5 Hamming Distance from a Certain Structure

The distribution of the Hamming distance from a certain reference structure provides information on how much probability is concentrated around the reference. If we choose a structure estimated by some point estimation algorithm as the reference, the calculated distribution provides new criteria for representing the credibility of the estimated structure.

(1) Definition of the Hamming Distance

We first introduce a vectorized representation of RNA secondary structure:

$$S[i][j] = \begin{cases} 1 & \text{(if i-th base and j-th base make a pair)} \\ 0 & \text{(otherwise)} \end{cases}. \tag{2.48}$$

Let us call this a structure vector. The dimension of a structure vector is $\binom{n}{2} = n(n-1)/2$ for RNA of length n.

We can then define the distance d between two structures by the Hamming distance of their structure vectors S_1 and S_2:

$$d = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_1[i][j] \oplus S_2[i][j]. \tag{2.49}$$

The Hamming distance d never exceeds its sequence length n in spite of its extremely high dimensions (See Appendix A for the proof).

(2) Application to the McCaskill Model

In this section, we show the concrete recursions, which are easily written down from equations (2.29) - (2.35) and Algorithm 7:

Initialization $(1 \le i \le n)$ **:**

$$\begin{aligned} Z(i,i) &= 1.0 && (2.50) \\ Z^1(i,i) &= Z^b(i,i) = Z^m(i,i) = Z^m(i,i-1) = Z^{m1}(i,i) = 0 && (2.51) \end{aligned}$$

Recursion $(1 \le i < j \le n)$ **:**

$$\begin{aligned} Z(i,j) &= x^{g_1(i,j,S)} + \sum_{k=i}^{j-1} Z(i,k) Z^1(k+1,j) x^{g_2(i,j,k,S)} && (2.52) \\ Z^1(i,j) &= \sum_{k=i+1}^{j} Z^b(i,k) x^{g_3(i,j,k,S)} && (2.53) \\ Z^b(i,j) &= e^{-\left[f_1(i,j)/k_B T\right]} x^{g_4(i,j,S)} + \sum_{k=i+1}^{j-2} \sum_{l=k+1}^{j-1} Z^b(k,l) e^{-\left[f_2(i,j,k,l)/k_B T\right]} x^{g_5(i,j,k,l,S)} \\ &\quad + \sum_{k=i+2}^{j-1} Z^m(i+1,k-1) Z^{m1}(k,j-1) e^{-\left[f_3(i,j)/k_B T\right]} x^{g_6(i,j,k,S)} && (2.54) \\ Z^m(i,j) &= \sum_{k=i}^{j-1} \left(e^{-\left[f_4(k-i)/k_B T\right]} x^{g_7(i,j,k,S)} + Z^m(i,k-1) x^{g_8(i,j,k,S)} \right) Z^{m1}(k,j) && (2.55) \\ Z^{m1}(i,j) &= \sum_{k=i+1}^{j} Z^b(i,k) e^{-\left[f_4(j-k)/k_B T\right]} x^{g_3(i,j,k,S)}, && (2.56) \end{aligned}$$

where S is the reference structure vector defined by equation (2.48), and each function $g_k(\cdot)$ $(k = 1 \cdots 8)$ returns an integer value as the Hamming distance from the reference structure, which accumulates with each transition. The full description of $g_k(\cdot)$ is as follows:

$$g_1(i,j,S) = \sum_{p=i}^{j-1}\sum_{q=p+1}^{j} S[p][q] \tag{2.57}$$

$$g_2(i,j,k,S) = \sum_{p=i}^{k}\sum_{q=k+1}^{j} S[p][q] \tag{2.58}$$

$$g_3(i,j,k,S) = \sum_{q=k+1}^{j}\sum_{p=i}^{q-1} S[p][q] \tag{2.59}$$

$$g_4(i,j,S) = \sum_{p=i}^{j-1}\sum_{q=p+1}^{j} S[p][q] + 1 - 2S[i][j] \tag{2.60}$$

$$\begin{aligned} g_5(i,j,k,l,S) &= \sum_{p=i}^{k-1}\sum_{q=p+1}^{j} S[p][q] + \sum_{p=k}^{l}\sum_{q=l+1}^{j} S[p][q] \\ &\quad + \sum_{p=l+1}^{j-1}\sum_{q=p+1}^{j} S[p][q] + 1 - 2S[i][j] \end{aligned} \tag{2.61}$$

$$\begin{aligned} g_6(i,j,k,S) &= \sum_{p=k}^{j-1} S[p][j] + \sum_{q=i+1}^{j} S[i][q] \\ &\quad + \sum_{p=i+1}^{k-1}\sum_{q=k}^{j} S[p][q] + 1 - 2S[i][j] \end{aligned} \tag{2.62}$$

$$g_7(i,j,k,S) = \sum_{p=i}^{k}\sum_{q=p+1}^{j} S[p][q] \tag{2.63}$$

$$g_8(i,j,k,S) = \sum_{p=i}^{k-1}\sum_{q=k}^{j} S[p][q]. \tag{2.64}$$

Calculation by this implementation requires $O(n^5)$ time and $O(n^3)$ memory. The following contrivances accelerate and decentralize the calculation, allowing them to be calculated with $O(n^3)$ time and $O(n^2 d_{max})$ memory under maximum decentralization.

(3) Pre-calculating for Comparing Distance between Structures

Equations (2.57) - (2.64) show that $O(n^2)$ calculations are needed for $g_k(\cdot)$, which derives the Hamming distance between two structures. This is one of the bottlenecks since these functions are embedded in recursive processes. If we pre-calculate a vector C before the recursive process, we obtain $g_k(\cdot)$ by $O(1)$ calculations. The definition of vector C_S corresponds to structure vector S as follows:

$$C_S[i][j] = \sum_{k=i}^{j-1} \sum_{l=k+1}^{j} S[k][l]. \tag{2.65}$$

Let us call this a cumulative structure vector.

C can be computed efficiently by the following dynamic programming technique:

Initialization:

$$C_S[i][i] = 0 \qquad (1 \leq i \leq n) \tag{2.66}$$

$$C_S[i][i+1] = S[i][i+1] \quad (1 \leq i \leq n-1) \tag{2.67}$$

Recursion $(1 \leq i \leq n-1, i+1 < j \leq n)$ **:**

$$C_S[i][j] = S[i][j] + C_S[i+1][j] + C_S[i][j-1] - C_S[i+1][j-1] \tag{2.68}$$

This pre-calculation requires $O(n^2)$ time.

We write down the $O(1)$ procedure of $g_k(\cdot)$ for reference.

$$\begin{aligned}
g_1(i,j,S) &= C_S[i][j] && (2.69)\\
g_2(i,j,k,S) &= C_S[i][j] - C_S[i][k] - C_S[k+1][j] && (2.70)\\
g_3(i,j,k,S) &= C_S[i][j] - C_S[i][k] && (2.71)\\
g_4(i,j,S) &= C_S[i][j] + 1 - 2S[i][j] && (2.72)\\
g_5(i,j,k,l,S) &= C_S[i][j] - C_S[k][l] + 1 - 2S[i][j] && (2.73)\\
g_6(i,j,k,S) &= C_S[i][j] - C_S[i+1][k-1] - C_S[k][j-1] + 1 - 2S[i][j] && (2.74)\\
g_7(i,j,k,S) &= C_S[i][j] - C_S[k][j] && (2.75)\\
g_8(i,j,k,S) &= C_S[i][j] - C_S[i][k-1] - C_S[k][j] && (2.76)
\end{aligned}$$

(4) Pre-calculating the Maximum of Distance

Equation (A.2) guarantees that the Hamming distance never exceeds the sequence length n. However, we can find the exact maximum value of the Hamming distance d_{max} as follows:

$$d_{max} = \max_{S_c \in \mathcal{S}} \left[\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_r[i][j] \oplus S_c[i][j] \right] \tag{2.77}$$

where $\mathcal{S}$ is a set of all possible candidate structure vectors and S_r is a reference structure vector.

The following $O(n^3)$ dynamic programming procedure is used to obtain d_{max}:

Initialization $(1 \le i \le n)$ **:**

$$D(i,i) = D^1(i,i) = D^b(i,i) = D^m(i,i) = D^m(i,i-1) = 0 \tag{2.78}$$

Recursion $(1 \le i \le n-1, i+1 < j \le n)$ **:**

$$D(i,j) = \max_k \left\{ \begin{array}{c} 0 \\ D(i,k) + D^1(k+1,j) - C_{S_r}[i][k] - C_{S_r}[k+1][j] \end{array} \right\} + C_{S_r}[i][j] \tag{2.79}$$

$$D^1(i,j) = \max_k \left\{ D^b_{(i,k)} - C_{S_r}[i][k] \right\} + C_{S_r}[i][j] \tag{2.80}$$

$$D^b(i,j) = \max_{k,l} \left\{ \begin{array}{c} 0 \\ D^b(k,l) - C_{S_r}[k][l] \\ D^m(i+1,k-1) + D^1(k,j-1) - C_{S_r}[i+1][k-1] - C_{S_r}[k][j-1] \end{array} \right\} + C_{S_r}[i][j] - 2S_r[i][j] + 1 \tag{2.81}$$

$$D^m(i,j) = \max_k \left\{ \begin{array}{c} D^m(i,k-1) + D^1(k,j) - C_{S_r}[i][k-1] - C_{S_r}[k][j] \\ D^1(k,j) - C_{S_r}[k][j] \end{array} \right\} + C_{S_r}[i][j] \tag{2.82}$$

Which finally gives d_{max} as $D(1,n)$.

2.3.6 $5' - 3'$ Distance of RNA Secondary Structure

Yoffe *et al.* reported that the distance between the 5′ and 3′ ends tends to be short and is largely independent of molecule length or sequence pattern [34]. They pointed out the relevance of these observations to biological interpretation, in particular of viral RNA evolution. A method for calculating the exact distribution of the 5′-3′ distances was proposed by Han *et al.* [35]. However, their method does not consider the existence probability of each structure nor the base pairing restrictions based on canonical base pairs. Instead, they assume that all structures occur at the same probability and that every base can make pairs with an arbitrary base, with the exception of pseudoknots. Although Clote *et al.* proposed a method for calculating an expected distance while considering the existence probability of each structure and the base pairing restrictions [36], no method is available for obtaining an exact distribution. Here we give an alternative algorithm for calculating this distribution.

(1) Definition of the $5' - 3'$ Distance

We follow the work of Yoffe and colleagues in defining the $5' - 3'$ distance $d_{5'-3'}$:

$$d_{5'-3'} = c_{ext} + h_{ext}, \tag{2.83}$$

where c_{ext} is the number of covalent bonds in the exterior loop and h_{ext} is the number of hydrogen bridges in the exterior loop. For example, in the secondary structure represented as Figure 2.1, counting the base pairs in the exterior loop (red arch), gives $c_{ext} = 9$, $h_{ext} = 2$, and accordingly $d_{5'-3'} = 9 + 2 = 11$.

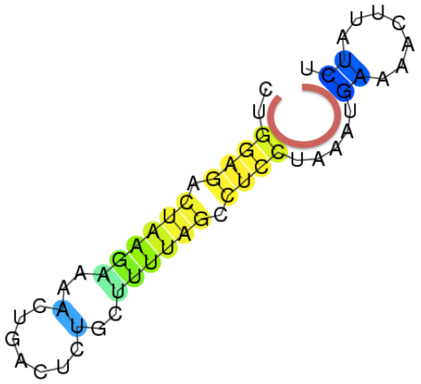


Figure 2.1: An example for introducing the definition of $d_{5'-3'}$.

(2) A Proposed Method for Calculating a $d_{5'-3'}$ Exact Distribution

We show a $O(n^4)$ time procedure for calculating a $d_{5'-3'}$ exact distribution.

Algorithm 9 Exact Calculation of a $d_{5'-3'}$ Distribution by DFT approach

```
$O(n^3)$ pre-calculation for $Z^b(i,j), (1 \leq i < j \leq n)$ by McCaskill model (equations (2.29) - (2.35))
for $k = 1$ to $n - 1$ do
    $x = \exp\left(2\pi i \frac{k-1}{n-1}\right)$
    $O(n^3)$ recursions for $Z_k(1,n)$ described on equations (2.84) - (2.87)
end for
for $k = 1$ to $n - 1$ do
    $p(d_{5'-3'} = k) = \left\{\sum_{r=1}^{n-1} Z_r(1,n)\left(\cos\left(\frac{2\pi k(r-1)}{n-1}\right) - i \sin\left(\frac{2\pi k(r-1)}{n-1}\right)\right)\right\} / (n-1)$ /*DFT*/
end for
```

The recursions implied above are as follows:

Initialization $(1 \leq i \leq n)$ **:**

$$Z_k(i,i) = 1.0 \tag{2.84}$$

$$Z_k^1(i,i) = 0 \tag{2.85}$$

Recursion $(1 \leq i < j \leq n)$ **:**

$$Z_k(i,j) = x^{j-i} + \sum_{k=i}^{j-1} Z_k(i,k) Z_k^1(k+1,j)x \tag{2.86}$$

$$Z_k^1(i,j) = \sum_{k=i+1}^{j} Z^b(i,k) x^{j-k+1} \tag{2.87}$$

Finally, we derive the probability $p(d_{5'-3'} = k)$.

2.4 Calculation of the Real Feature Distribution

If the feature score is not an integer but a real value, we cannot describe exact distributions; the probability distribution of the score is generally discrete and has almost unenumerable variations since the ensemble of solutions is also assumed to be discrete and very large. Therefore, we introduce an alternative method for calculating the objective feature distribution.

2.4.1 General Theory

First, we divide the feature value into bins and define a window function to distribute each probability to these bins. Next, we show the conditions required for a fine window function. Third, we introduce the window function adopted in this study. Finally, we construct an algorithm to calculate the objective distribution using this window.

2.4.2 Definition of Bins and Window Function

As described above, we cannot enumerate all instances if the feature value is a bounded real value. Instead, of calculating the distribution of proper value directly, therefore, we define equal interval bins which divide the feature value and allocate each probability to a bin according to the feature value. A windows function is an allocator which distributes the probabilities to the bins. In the following discussion, for brevity we assume that the minimum feature value is 0, that each bin size is 1, and that the number of bins is N (Figure 2.2).

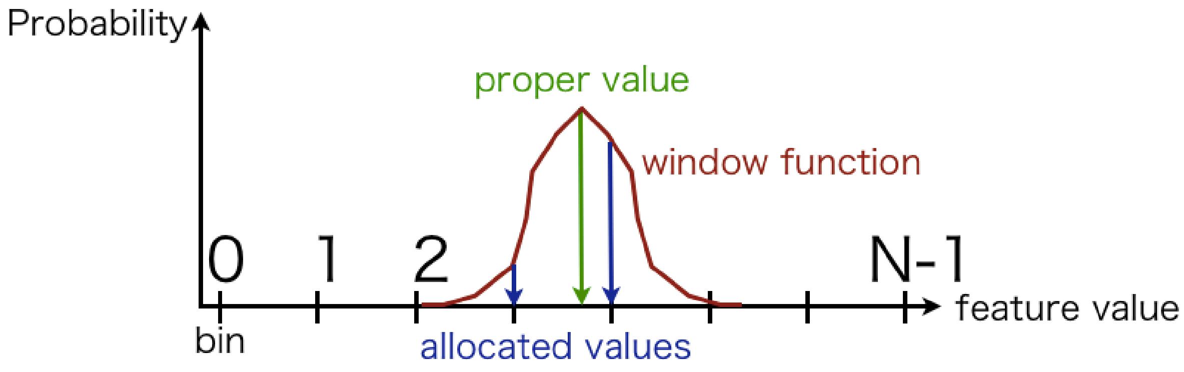


Figure 2.2: The definition of bins and a window function to distribute each probability into the bins.

Once the window function has been defined, we calculate the probability in the v-th bin by the following equation:

$$p_v = \sum_{\theta \in C} z_\theta f(s_\theta, v)/Z, \tag{2.88}$$

where:

$$\begin{aligned} C &: \text{a set of candidate solutions} \\ z_\theta &: \text{Boltzmann factor of } \theta \\ s_\theta &: \text{feature value of } \theta \\ f(s_\theta, v) &: \text{window function} \end{aligned}$$

The goal is to find a fine window function which enables p_v to be precisely and efficiently obtained.

2.4.3 Necessary Conditions for a Window Function

Three conditions are required for a good window function f.

First, the sum of the probabilities allocated to the bins must be equal to the original probability. This condition is equivalent to the following equation:

$$^\forall\theta, \sum_{v=0}^{N-1} f(s_\theta, v) = 1 \tag{2.89}$$

Second, it is necessary that some efficient algorithm exists for calculating equation (2.88).

Finally, the approximate distribution derived by applying the window function must be as precise as possible.

The most natural candidate for the window function might be the rectangular function (Figure 2.3) which allocates probability into the nearest bin:

$$f(s_\theta, v) = \begin{cases} 1 & (v - 1/2 \leq s_\theta < v + 1/2) \\ 0 & (otherwise) \end{cases}. \tag{2.90}$$

However, while the first and third conditions are satisfied, this function is unlikely to satisfy the second condition.

We briefly review some well-known window and bell-shaped functions. Window functions, including the rectangular, triangular, Parzen, Hann, Hamming,

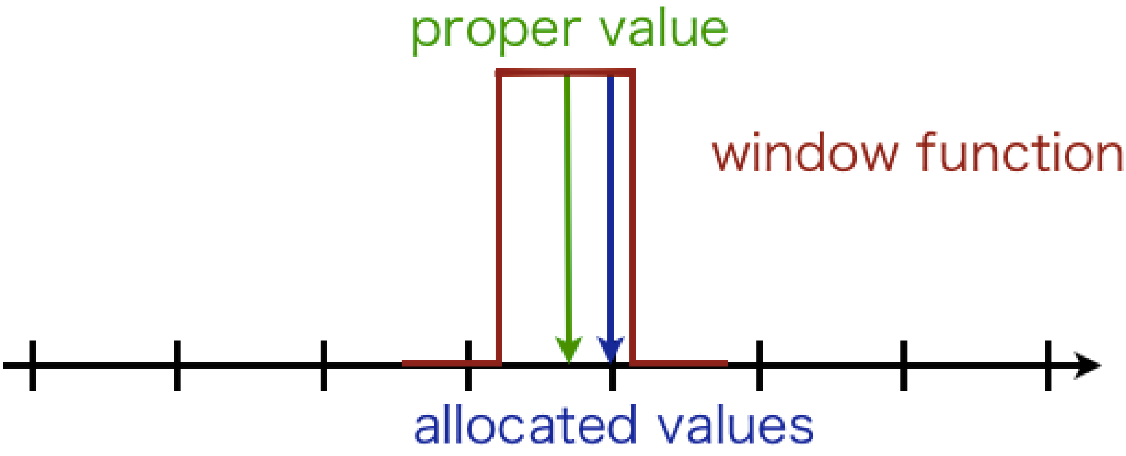


Figure 2.3: The illustration of adopting the rectangular function as the window function.

and Blackman windows, are often described in the following form:

$$f(s_\theta, v) = \begin{cases} \text{some function} & (|s_\theta - v| \le \alpha) \\ 0 & (otherwise) \end{cases}. \tag{2.91}$$

Some of these windows satisfy the first and the third conditions, but it is probably impossible for them to satisfy the second condition, as dividing the case according to $|s_\theta - v|$ complicates the calculation. Thus, being globally smooth is considered a necessary condition for a window function.

The following functions are well-known smooth bell-shaped window functions:

$$f(s_\theta, v) = \exp\left[-\frac{(s_\theta - v)^2}{\sigma^2}\right] \tag{2.92}$$

$$f(s_\theta, v) = \frac{\alpha}{1 + (s_\theta - v)^2} \tag{2.93}$$

$$f(s_\theta, v) = \alpha \frac{\sin^2(s_\theta - v)}{(s_\theta - v)^2} \tag{2.94}$$

$$f(s_\theta, v) = \frac{\alpha \exp\left[-|s_\theta - v|\right]}{(1 + \exp\left[-|s_\theta - v|\right])^2}. \tag{2.95}$$

However, these conventional functions only approximately satisfy the first condition and probably cannot satisfy the second condition, though some of the functions satisfy it approximately. The second condition is satisfied if $f(s_\theta, v)$ is polynomial. Therefore, a Taylor expansion of the above $f(s_\theta, v)$ seems applicable if its convergence radius is infinite. However, this is not realistic in practice because it requires superhigh-order terms.

As described above, the selection of the proper window function is an inflexible process.

2.4.4 The Window Function We Adopted

In this study, we propose the following original window function:

$$f(s_\theta, v) = \frac{1}{N}\int_{v-\frac{1}{2}}^{v+\frac{1}{2}} \cos\left[\pi\frac{(s_\theta - t)(N-1)}{N}\right]\frac{\sin\left[\pi(s_\theta - t)\right]}{\sin\left[\pi\frac{s_\theta - t}{N}\right]}dt. \tag{2.96}$$

The three conditions given above are satisfied by function (2.96).

(1) A Proof that Equation (2.89) is Satisfied

We now show a proof of the following equation:

$$^\forall\theta, \sum_{v=0}^{N-1}\frac{1}{N}\int_{v-\frac{1}{2}}^{v+\frac{1}{2}} \cos\left[\pi\frac{(s_\theta - t)(N-1)}{N}\right]\frac{\sin\left[\pi(s_\theta - t)\right]}{\sin\left[\pi\frac{s_\theta - t}{N}\right]}dt = 1. \tag{2.97}$$

First, the left side can be modified as follows:

$$\sum_{v=0}^{N-1}\frac{1}{N}\int_{-\frac{1}{2}}^{\frac{1}{2}} \cos\left[\pi\frac{((s_\theta - t) - v)(N-1)}{N}\right]\frac{\sin\left[\pi((s_\theta - t) - v)\right]}{\sin\left[\pi\frac{(s_\theta - t) - v}{N}\right]}dt. \tag{2.98}$$

Thus, (2.97) is satisfied if the following equation is satisfied:

$$-\frac{1}{2} \leq^\forall x \leq N - \frac{1}{2}, \sum_{v=0}^{N-1}\frac{1}{N}\cos\left[\pi\frac{(x-v)(N-1)}{N}\right]\frac{\sin\left[\pi(x-v)\right]}{\sin\left[\pi\frac{x-v}{N}\right]} = 1. \tag{2.99}$$

The left side of equation (2.99) can be modified as follows:

$$
\begin{aligned}
&\sum_{v=0}^{N-1}\frac{1}{N}\cos\left[\pi\frac{(x-v)(N-1)}{N}\right]\frac{\sin\left[\pi(x-v)\right]}{\sin\left[\pi\frac{x-v}{N}\right]} && (2.100)\\
=&\sum_{v=0}^{N-1}\frac{1}{N}\cos\left[\pi x-\pi\frac{x-v}{N}-\pi v\right]\frac{\sin\left[\pi x-\pi v\right]}{\sin\left[\pi\frac{x-v}{N}\right]} && (2.101)\\
=&\sum_{v=0}^{N-1}\frac{1}{N}\cos\left[\pi x-\pi\frac{x-v}{N}\right]\frac{\sin\left[\pi x\right]}{\sin\left[\pi\frac{x-v}{N}\right]} && (2.102)\\
=&\sum_{v=0}^{N-1}\frac{1}{N}\left(\sin\left[\pi x\right]\sin\left[\pi\frac{x-v}{N}\right]+\cos\left[\pi x\right]\cos\left[\pi\frac{x-v}{N}\right]\right)\frac{\sin\left[\pi x\right]}{\sin\left[\pi\frac{x-v}{N}\right]} && (2.103)\\
=&\sum_{v=0}^{N-1}\frac{1}{N}\left(\sin^2\left[\pi x\right]+\frac{\sin\left[\pi x\right]\cos\left[\pi x\right]}{\tan\left[\pi\frac{x-v}{N}\right]}\right) && (2.104)\\
=&\sin^2\left[\pi x\right]+\frac{1}{N}\sum_{v=0}^{N-1}\frac{\sin\left[\pi x\right]\cos\left[\pi x\right]}{\tan\left[\pi\frac{x-v}{N}\right]}. && (2.105)
\end{aligned}
$$

If $\cos\left[\pi x\right]=0$, expression (2.105) becomes 1 since $\sin^2\left[\pi x\right]=1$.

If $\sin\left[\pi x\right]=0$, that is $x\in\mathbb{Z}\wedge[-1/2,N-1/2]$, the above expression also becomes 1 because:

$$
\begin{aligned}
&\lim_{x-v\to 0}\frac{1}{N}\cos\left[\pi x-\pi\frac{x-v}{N}-\pi v\right]\frac{\sin\left[\pi x-\pi v\right]}{\sin\left[\pi\frac{x-v}{N}\right]} && (2.106)\\
=&\lim_{x-v\to 0}\frac{1}{N}\cos\left[\pi x-\pi\frac{x-v}{N}-\pi v\right]\frac{\sin\left[\pi x-\pi v\right]}{\pi x-\pi v}\frac{\pi\frac{x-v}{N}}{\sin\left[\pi\frac{x-v}{N}\right]}\frac{\pi x-\pi v}{\pi\frac{x-v}{N}} && (2.107)\\
=&\lim_{x-v\to 0}\frac{1}{N}N && (2.108)\\
=&1 && (2.109)
\end{aligned}
$$

and

$$
\frac{1}{N}\cos\left[\pi x-\pi\frac{x-v}{N}-\pi v\right]\frac{\sin\left[\pi x-\pi v\right]}{\sin\left[\pi\frac{x-v}{N}\right]}=0\quad(|x-v|\in\mathbb{Z}\wedge[1,N-1]). \qquad (2.110)
$$

In the following discussion we assume that $\cos\left[\pi x\right]\neq 0$ and $\sin\left[\pi x\right]\neq 0$.

Expression (2.105) becomes 1 if the following equation is satisfied:

$$\sum_{v=0}^{N-1} \frac{\sin[\pi x]\cos[\pi x]}{\tan\left[\pi\frac{x-v}{N}\right]} = N\cos^2[\pi x] \tag{2.111}$$

$$\Leftrightarrow \sum_{v=0}^{N-1} \cot\left[\pi\frac{x-v}{N}\right] = N\cot[\pi x]. \tag{2.112}$$

We prove equation (2.112) by the following description.

First, if we define $z = e^{2i\theta}$, $\cot\theta$ can be described by:

$$\cot\theta = i\frac{z+1}{z-1}. \tag{2.113}$$

Thus,

$$\cot N\theta = i\frac{z^N+1}{z^N-1}. \tag{2.114}$$

If we define $u = e^{2\pi i/N}$, u^v ($v = 0, \ldots, N-1$) are solutions for $z^N - 1 = 0$. Therefore,

$$z^N - 1 = \prod_{v=0}^{N-1}(z - u^v). \tag{2.115}$$

Thus,

$$\cot N\theta = i\frac{z^N+1}{\prod_{v=0}^{N-1}(z-u^v)}. \tag{2.116}$$

By partial fraction decomposition, the above equation can be modified as:

$$i\frac{z^N+1}{\prod_{v=0}^{N-1}(z-u^v)} = i\frac{z^N-1+2}{\prod_{v=0}^{N-1}(z-u^v)} \tag{2.117}$$

$$= i\left[1 + \sum_{v=0}^{N-1}\frac{2a_v}{z-u^v}\right]. \tag{2.118}$$

From the following identical equation,

$$f(z) = \sum_{v=0}^{N-1}\left(\prod_{v=0}^{N-1}(z-u^v)\right)\frac{a_v}{z-u^v} = 1. \tag{2.119}$$

By substituting $f(z) = f(u^v)$ $(v = 0, \ldots, N-1)$, we obtain:

$$a_v = \left(\prod_{j=0,1,\ldots,v-1,v+1,\ldots,N-1} (u^v - u^j) \right)^{-1}. \tag{2.120}$$

From equation (2.115),

$$\begin{aligned} a_0 =& \left(\prod_{j=1}^{N-1} (1 - u^j) \right)^{-1} && (2.121) \\ =& \lim_{z \to 1} \frac{z-1}{z^N - 1} && (2.122) \\ =& \lim_{z \to 1} \frac{z-1}{(z-1) \sum_{j=0}^{N-1} z^j} && (2.123) \\ =& \frac{1}{N}. && (2.124) \end{aligned}$$

In addition, because $u^N = e^{2\pi i} = 1$,

$$\begin{aligned} a_v =& \frac{a_v}{u^N} && (2.125) \\ =& \frac{1}{u} \left(\prod_{j=0,1,\ldots,v-1,v+1,\ldots,N-1} u(u^v - u^j) \right)^{-1} && (2.126) \\ =& \frac{1}{u} \left(\prod_{j=0,1,\ldots,v-1,v+1,\ldots,N-1} (u^{v+1} - u^{j+1}) \right)^{-1} && (2.127) \\ =& \frac{1}{u} \left(\prod_{j=1,2,\ldots,v,v+2,\ldots,N} (u^{v+1} - u^j) \right)^{-1} && (2.128) \\ =& \frac{1}{u} \left(\prod_{j=0,1,\ldots,v,v+2,\ldots,N-1} (u^{v+1} - u^j) \right)^{-1} && (2.129) \\ =& \frac{1}{u} a_{v+1} && (2.130) \\ & \Leftrightarrow a_{v+1} = u a_v. && (2.131) \end{aligned}$$

From equation (2.124) and equation (2.131):

$$a_v = \frac{u^v}{N}. \tag{2.132}$$

Thus,

$$\begin{aligned}
\cot N\theta =& i\left[1+\sum_{v=0}^{N-1}\frac{2a_v}{z-u^v}\right] && (2.133)\\
=& i\left[1+\frac{1}{N}\sum_{v=0}^{N-1}\frac{2u^v}{z-u^v}\right] && (2.134)\\
=& i\left[1+\frac{1}{N}\sum_{v=0}^{N-1}\frac{z+u^v-(z-u^v)}{z-u^v}\right] && (2.135)\\
=& i\left[1+\frac{1}{N}\sum_{v=0}^{N-1}\left(\frac{z+u^v}{z-u^v}-1\right)\right] && (2.136)\\
=& \frac{i}{N}\sum_{v=0}^{N-1}\frac{z+u^v}{z-u^v} && (2.137)\\
=& \frac{i}{N}\sum_{v=0}^{N-1}\frac{zu^{-v}+1}{zu^{-v}-1} && (2.138)\\
=& \frac{i}{N}\sum_{v=0}^{N-1}\frac{ze^{-2i\frac{\pi v}{N}}+1}{ze^{-2i\frac{\pi v}{N}}-1} && (2.139)\\
=& \frac{1}{N}\sum_{v=0}^{N-1}\cot\left[\theta-\frac{\pi v}{N}\right]. && (2.140)
\end{aligned}$$

By substituting $\theta = \pi x/N$ into equation (2.140), we obtain:

$$\cot[\pi x] = \frac{1}{N}\sum_{v=0}^{N-1}\cot\left[\frac{\pi x}{N}-\frac{\pi v}{N}\right] \tag{2.141}$$

$$\Leftrightarrow \sum_{v=0}^{N-1}\cot\left[\pi\frac{x-v}{N}\right] = N\cot[\pi x]. \tag{2.142}$$

This proves equation (2.112). As described above, this confirms that the first condition of the window function is satisfied.

(2) How to Calculate Equation (2.88) Efficiently

Next, we confirm that the second condition is also satisfied. Equation (2.88) can be expanded as follows:

$$
\begin{aligned}
p_v &= \sum_{\theta \in C} z_\theta f(s_\theta, v)/Z && (2.143)\\
&= \sum_{\theta \in C} z_\theta \frac{1}{N} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} \cos\left[\pi \frac{(s_\theta - t)(N-1)}{N}\right] \frac{\sin\left[\pi(s_\theta - t)\right]}{\sin\left[\pi \frac{s_\theta - t}{N}\right]} dt/Z && (2.144)\\
&= \sum_{\theta \in C} z_\theta \frac{1}{N} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\exp\left[\pi i \frac{(s_\theta - t)(N-1)}{N}\right] \frac{\sin\left[\pi(s_\theta - t)\right]}{\sin\left[\pi \frac{s_\theta - t}{N}\right]}\right] dt/Z && (2.145)\\
&= \sum_{\theta \in C} z_\theta \frac{1}{N} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\frac{\exp\left[2\pi i(s_\theta - t)\right] - 1}{\exp\left[2\pi i \frac{(s_\theta - t)}{N}\right] - 1}\right] dt/Z && (2.146)\\
&= \sum_{\theta \in C} z_\theta \frac{1}{N} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\sum_{p=0}^{N-1} \exp\left[2\pi i \frac{p(s_\theta - t)}{N}\right]\right] dt/Z && (2.147)\\
&= \frac{1}{NZ} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\sum_{\theta \in C} z_\theta \sum_{p=0}^{N-1} \exp\left[2\pi i \frac{p(s_\theta - t)}{N}\right]\right] dt && (2.148)\\
&= \frac{1}{NZ} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\sum_{\theta \in C} z_\theta \sum_{p=0}^{N-1} \exp\left[2\pi i \frac{p s_\theta}{N}\right] \exp\left[2\pi i \frac{-pt}{N}\right]\right] dt && (2.149)\\
&= \frac{1}{NZ} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} Re\left[\sum_{p=0}^{N-1} \sum_{\theta \in C} z_\theta \exp\left[2\pi i \frac{p s_\theta}{N}\right] \exp\left[2\pi i \frac{-pt}{N}\right]\right] dt && (2.150)\\
&\approx \frac{1}{NZM} Re\left[\sum_{q=0}^{M-1} \sum_{p=0}^{N-1} \sum_{\theta \in C} z_\theta \exp\left[2\pi i \frac{p s_\theta}{N}\right] \exp\left[2\pi i \frac{-p\left(v - \frac{1}{2} + \frac{q}{M}\right)}{N}\right]\right]. && (2.151)
\end{aligned}
$$

Thus, to obtain p_v, we calculate the following part first:

$$
\sum_{\theta \in C} z_\theta \exp\left[2\pi i \frac{p s_\theta}{N}\right], \tag{2.152}
$$

which is a similar procedure to that in the integer version. We then execute MN^2-time continuous calculations. MN-time calculations are distributed into $(N + M)$-time calculations. Therefore, the real feature value distribution can be calculated by the same order with integer feature algorithm.

(3) Precision as an Allocator

We show the shape of the proposed window function in Figure 2.4. We can observe that the shape converges to $N = \infty$ as the region is divided into smaller pieces, and that $N = 100$ and $N = \infty$ are approximately identical.

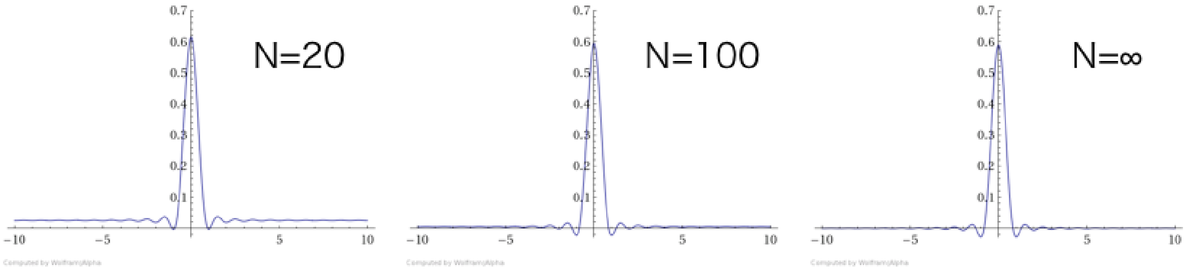


Figure 2.4: The shapes of the window functions in three cases: $N = 20, 100$, and ∞ (x-axis represents $s_\theta - v$).

When $N = \infty$, the window function can be written as below (applying sectional mensuration):

$$\lim_{N\to\infty} f(s_\theta, v) = \lim_{N\to\infty} \frac{1}{N} \int_{v-\frac{1}{2}}^{v+\frac{1}{2}} \cos\left[\pi \frac{(s_\theta - t)(N-1)}{N}\right] \frac{\sin\left[\pi(s_\theta - t)\right]}{\sin\left[\pi \frac{s_\theta - t}{N}\right]} dt \tag{2.153}$$

$$= \int_0^{\pi - 2\pi(s_\theta - v)} \frac{\sin t}{\pi t} dt + \int_0^{\pi + 2\pi(s_\theta - v)} \frac{\sin t}{\pi t} dt. \tag{2.154}$$

2.4.5 Canceling the Side Lobe Effect

We define the side lobe effect as the probability leak to distant bins by the window function (see Figure 2.5). The most naive correction method is to assume a uniform constant bias. This is then corrected by the following estimator:

$$\hat{p_v} = \frac{p_v - \alpha}{1 - \alpha}. \tag{2.155}$$

Actually, this assumption is optimistic, as the level of a probability leak depends on the positional relationship to the nearest bin. However, this positional dependency decreases as more distant bins are added to the summation (see Figure 2.6). The Y-axis in Figure 2.6 represents the total weight of the probability allocated to each range of bins. Thus, we calculate the probability at a finer resolution than required, then combine the bins to avoid the positional dependency.

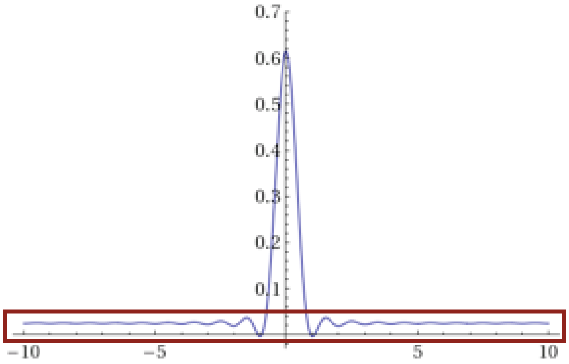


Figure 2.5: The definition of the side lobe (the red-colored part).

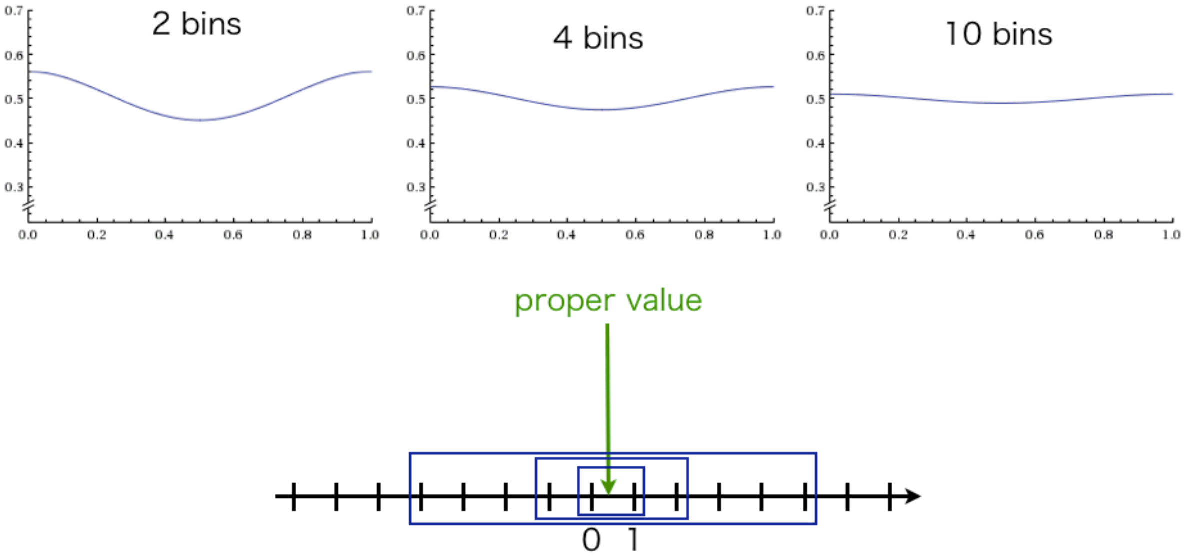


Figure 2.6: Positional dependency canceled by considering the summation of farther bins.

2.4.6 Application to the Features of RNA Secondary Structure

2.4.7 Free Energy of the RNA Secondary Structure

In this section we show a concrete algorithm. If we define the free energy as a real feature score, the procedure for calculating the distribution is given by Algorithm 10:

Algorithm 10 Calculating the distribution of free energy of RNA secondary structure

```
for $n = 0$ to $N - 1$ do
    $x = \exp\left(2\pi i \frac{n}{N}\right)$
    $Z_n(0) = 1$
    for $k = 1$ to $S$ do
        $Z_n(k) = \sum_{i=0}^{k-1} Z_n(i) \times (\text{Boltzmann factor gain by i to k}) \times x^{\text{Free energy gain by i to k}}$
    end for
end for
for $m = 0$ to $M - 1$ do
    $l = \frac{m}{M} - \frac{1}{2}$
    for $v = 0$ to $N - 1$ do
        $p_{mv} = Re\left[\sum_{n=0}^{N-1} Z_n(S)\left(\cos\left(\frac{2\pi v(n+l)}{N}\right) - i \sin\left(\frac{2\pi v(n+l)}{N}\right)\right)\right]/N$
    end for
end for
for $v = 0$ to $N - 1$ do
    $p_v = \sum_{m=0}^{M-1} p_{mv}/M$
end for
```

2.5 Calculation of the Feature Vector Distribution

2.5.1 General Theory and the 2D Expansion of the Integer Distribution

We next expand the above integer and real feature distribution to an integer and real vector feature distribution. In this study, we use the case of a 2D expansion of the algorithm for the Hamming distance from a certain structure.

Algorithm 11 gives the 2D expansion of Algorithm 7. Information on the distance from two reference structures is accumulated as exponents of x and y separately. Functions $s(k)$, $s(i,k)$, and $s(i,j,k)$ have captions x and y because their given score depends on the structure even when the transition is the same.

Algorithm 11 2D Expansion of Algorithm 7

```
$Z(0) = 1$
**for** $k = 1$ to $N$ **do**
    $Z(k) = \sum_{i=0}^{k-1} Z(i)t(k|i)x^{s_x(i,k)}y^{s_y(i,k)} + \sum_{i=0}^{k-2}\sum_{j=i+1}^{k-1} Z(i)Z(j)t(k|i,j)x^{s_x(i,j,k)}y^{s_y(i,j,k)}$
**end for**
```

Each polynomial factor of $x^k y^l$ is proportional to the existence probability of structure k from reference structure 1 and l from reference structure 2. The actual probability is obtained by the following equation:

$$p_{kl} = \frac{a_{kl}}{\sum_{i=0}^{i=n}\sum_{j=0}^{j=n} a_{ij}} \tag{2.156}$$

$$Z(N) \equiv \sum_{i=0}^{n}\sum_{j=0}^{n} a_{ij}x^i y^j. \tag{2.157}$$

We describe the concrete recursions after giving the definition of distance.

(1) Naive Implementation of 2D Algorithm

Next, we expand the 1D algorithm into two dimensions. This expansion enables us to compare two structures using $O(n^7)$ time and $O(n^4)$ memory. We employ polynomials in x and y instead of polynomials in x so as to accumulate the information on the Hamming distance from the two reference secondary structures.

Initialization $(1 \leq i \leq n)$ **:**

$$\begin{aligned} Z(i,i) &= 1.0 && (2.158) \\ Z^1(i,i) &= Z^b(i,i) = Z^m(i,i) = Z^m(i,i-1) = Z^{m1}(i,i) = 0 && (2.159) \end{aligned}$$

Recursion $(1 \leq i < j \leq n)$ **:**

$$Z(i,j) = x^{g_1(i,j,S_1)}y^{g_1(i,j,S_2)} + \sum_{k=i}^{j-1} Z(i,k)Z^1(k+1,j)x^{g_2(i,j,k,S_1)}y^{g_2(i,j,k,S_2)} \quad (2.160)$$

$$Z^1(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)x^{g_3(i,j,k,S_1)}y^{g_3(i,j,k,S_2)} \quad (2.161)$$

$$\begin{aligned} Z^b(i,j) &= e^{-\left[f_1(i,j)/k_BT\right]}x^{g_4(i,j,S_1)}y^{g_4(i,j,S_2)} + \sum_{k=i+1}^{j-2}\sum_{l=k+1}^{j-1} Z^b(k,l)e^{-\left[f_2(i,j,k,l)/k_BT\right]}x^{g_5(i,j,k,l,S_1)}y^{g_5(i,j,k,l,S_2)} \\ &+ \sum_{k=i+2}^{j-1} Z^m(i+1,k-1)Z^{m1}(k,j-1)e^{-\left[f_3(i,j)/k_BT\right]}x^{g_6(i,j,k,S_1)}y^{g_6(i,j,k,S_2)} \quad (2.162) \end{aligned}$$

$$\begin{aligned} Z^m(i,j) &= \sum_{k=i}^{j-1}\Bigg(e^{-\left[f_4(k-i)/k_BT\right]}x^{g_7(i,j,k,S_1)}y^{g_7(i,j,k,S_2)} \\ &\qquad + Z^m(i,k-1)x^{g_8(i,j,k,S_1)}y^{g_8(i,j,k,S_2)}\Bigg)Z^{m1}(k,j) \quad (2.163) \end{aligned}$$

$$Z^{m1}(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)e^{-\left[f_4(j-k)/k_BT\right]}x^{g_3(i,j,k,S_1)}y^{g_3(i,j,k,S_2)} \quad (2.164)$$

where S_1 and S_2 are the two arbitrary reference structure vectors.

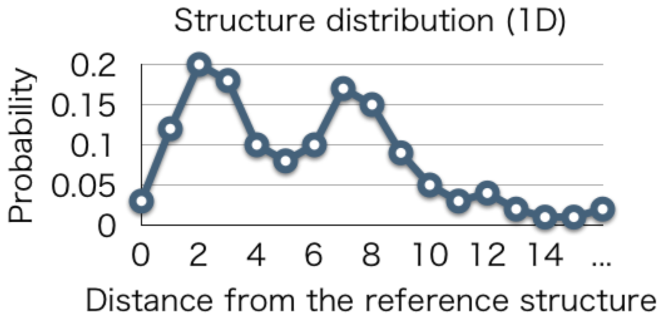


Figure 2.7: An example of structure distribution obtained by our 1D Algorithm.

The significance of the 2D expansion emerges when we observe the plural peaks from the 1D distribution. For example, let us assume a case in which we obtain a distribution like that in Figure 2.7 by the 1D Algorithm. This figure seems to suggest two structure clusters, but several different scenarios are possible. The leftmost image in Figure 2.8 shows the case in which the two structure clusters are clearly separate, while the central image shows the case in which the two structure clusters intercommunicate. The rightmost image in Figure 2.8 suggests the possibility of other structure clusters because large-scale structures are present, apart from the reference structure. Figure 2.7 cannot guarantee the existence of another significant cluster since the probability might be dispersed broadly among structures that are a similar distance from the reference structure.

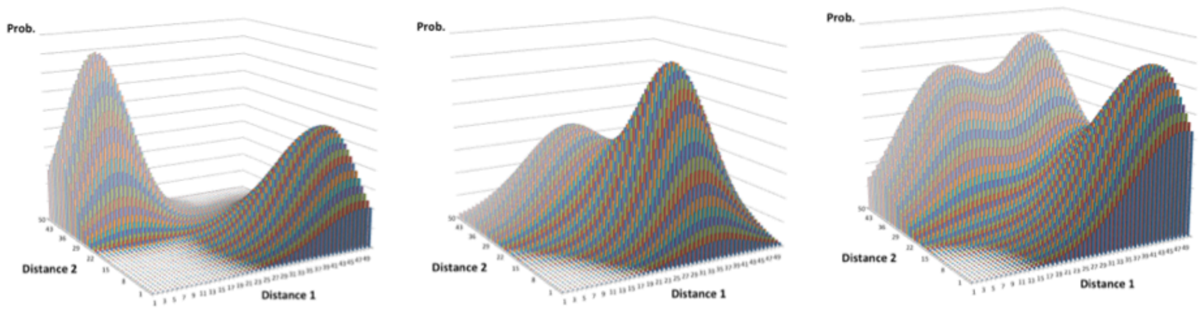


Figure 2.8: Examples of possible patterns of two structure clusters.

Algorithm 12 Naive 2D algorithm using the DFT approach

```
for $d_1 = 0$ to $d_{1max}$ do
    for $d_2 = 0$ to $d_{2max}$ do
        $x = \exp\left(2\pi i \frac{d_1}{d_{1max}+1}\right)$
        $y = \exp\left(2\pi i \frac{d_2}{d_{2max}+1}\right)$
        $Z_{(d_1,d_2)}(0) = 1$
        for $k = 1$ to $N$ do
            $Z_{(d_1,d_2)}(k) = \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z_{(d_1,d_2)}(i) Z_{(d_1,d_2)}(j) t(k|i,j) x^{s_x(i,j,k)} y^{s_y(i,j,k)} + \sum_{i=0}^{k-1} Z_{(d_1,d_2)}(i) t(k|i) x^{s_x(i,k)} y^{s_y(i,k)}$
        end for
    end for
end for
for $d_1 = 0$ to $d_{1max}$ do
    for $d_2 = 0$ to $d_{2max}$ do
        $Z'_{(d_1,d_2)} = \left\{\sum_{r=0}^{d_{2max}} Z_{(d_1,r)}(N) \left(\cos\left(\frac{2\pi r}{d_{2max}+1}\right) - i \sin\left(\frac{2\pi r}{d_{2max}+1}\right)\right)\right\} / (1 + d_{2max})$ /*DFT*/
    end for
end for
for $d_2 = 0$ to $d_{2max}$ do
    for $d_1 = 0$ to $d_{1max}$ do
        $p(d_1, d_2) = \left\{\sum_{r=0}^{d_{1max}} Z'_{(r,d_2)} \left(\cos\left(\frac{2\pi r}{d_{1max}+1}\right) - i \sin\left(\frac{2\pi r}{d_{1max}+1}\right)\right)\right\} / (1 + d_{1max})$ /*DFT*/
        /*where $p(d_1, d_2)$ is the probability that distance from structure 1 is $d_1$ and structure 2 is $d_2$*/
    end for
end for
```

(2) Modifying Recursions for 2D Algorithm

Actually, distributions computed by the 2D Algorithm are quite sparse. From constraints such as triangle inequality, the following expressions must be satisfied:

$$\forall S \in \mathcal{S} \quad , \quad |d(S_{R_1}, S) - d(S_{R_2}, S)| \leq d(S_{R_1}, S_{R_2}) \leq d(S_{R_1}, S) + d(S_{R_2}, S) \tag{2.165}$$

$$\forall S \in \mathcal{S} \quad , \quad d(S_{R_1}, S) \leq d_{1max} \tag{2.166}$$

$$\forall S \in \mathcal{S} \quad , \quad d(S_{R_2}, S) \leq d_{2max} \tag{2.167}$$

$$\forall S \in \mathcal{S} \quad , \quad \exists m \in \mathbb{N}, d(S_{R_1}, S) + d(S_{R_2}, S) + d(S_{R_1}, S_{R_2}) = 2m \tag{2.168}$$

where $\mathbb{N}$ is a set of natural numbers, $\mathcal{S}$ is a set of all possible secondary structure vectors, $S_{R_i}(i = 1, 2)$ is a structure vector of the i-th reference, and $d(S_1, S_2)$ is the Hamming distance between S_1 and S_2. Equation (2.165) is derived from triangle inequality, and equation (2.166) and (2.167) originate in definitions of d_{1max} and d_{2max}. The logic behind equation (2.168) is complex, but can be simplified as follows: a 1 bit transition of the structure vector of S invariably causes a change of 1 Hamming distance from any other structures, and every structure vector can visit every other one by repetition of 1 bit transitions.

To remove unnecessary calculations, we modified our 2D algorithm by converting the axes. The improved algorithm is shown as Algorithm 13.

Algorithm 13 Improved 2D algorithm by DFT approach

```
$\delta = d(S_{R_1}, S_{R_2})$
$d'_{1max} = \delta$
$d'_{2max} = \frac{d_{1max}+d_{2max}-\delta}{2}$
for $d_1 = 0$ to $d'_{1max}$ do
    for $d_2 = 0$ to $d'_{2max}$ do
        $x = \exp\left(2\pi i \frac{d_1}{d'_{1max}+1}\right)$
        $y = \exp\left(2\pi i \frac{d_2}{d'_{2max}+1}\right)$
        $Z_{(d_1,d_2)}(0) = 1$
        for $k = 1$ to $N$ do
            $Z_{(d_1,d_2)}(k) = \sum_{i=0}^{k-2} \sum_{j=i+1}^{k-1} Z_{(d_1,d_2)}(i) Z_{(d_1,d_2)}(j) t(k|i,j) x^{s_x(i,j,k)} y^{s_y(i,j,k)}$
            $+ \sum_{i=0}^{k-1} Z_{(d_1,d_2)}(i) t(k|i) x^{s_x(i,k)} y^{s_y(i,k)}$
        end for
    end for
end for
for $d_1 = 0$ to $d'_{1max}$ do
    for $d_2 = 0$ to $d'_{2max}$ do
        $Z'_{(d_1,d_2)} = \left\{\sum_{r=0}^{d'_{2max}} Z_{(d_1,r)}(N)\left(\cos\left(\frac{2\pi r}{d'_{2max}+1}\right) - i\sin\left(\frac{2\pi r}{d'_{2max}+1}\right)\right)\right\} /(1 + d'_{2max})$ /*DFT*/
    end for
end for
for $d_2 = 0$ to $d'_{2max}$ do
    for $d_1 = 0$ to $d'_{1max}$ do
        $Z''_{(d_1,d_2)} = \left\{\sum_{r=0}^{d'_{1max}} Z'_{(r,d_2)}\left(\cos\left(\frac{2\pi r}{d'_{1max}+1}\right) - i\sin\left(\frac{2\pi r}{d'_{1max}+1}\right)\right)\right\} /(1 + d'_{1max})$ /*DFT*/
    end for
end for
for $d_1 = 0$ to $d_{1max}$ do
    for $d_2 = 0$ to $d_{2max}$ do
        if ($|d_1 - d_2| \leq \delta \leq d_1 + d_2$) and ($d_1 + d_2 + \delta$ is even) then
            $p(d_1, d_2) = Z''_{(\frac{d_1-d_2+\delta}{2}, \frac{d_1+d_2-\delta}{2})}$
        else
            $p(d_1, d_2) = 0$
            /*where $p(d_1, d_2)$ is the probability that distance from structure 1 is $d_1$
            and structure 2 is $d_2$*/
        end if
    end for
end for
```

In this case, we need a minor modification to recursion (2.160) - (2.164) as follows:

$$\begin{aligned} Z(i,j) &= x^{\frac{g_1(i,j,S_1)-g_1(i,j,S_2)+\Delta\delta_1(i,j)}{2}} y^{\frac{g_1(i,j,S_1)+g_1(i,j,S_2)-\Delta\delta_1(i,j)}{2}} \\ &+ \sum_{k=i}^{j-1} Z(i,k)Z^1(k+1,j)x^{\frac{g_2(i,j,k,S_1)-g_2(i,j,k,S_2)+\Delta\delta_2(i,j,k)}{2}} y^{\frac{g_2(i,j,k,S_1)+g_2(i,j,k,S_2)-\Delta\delta_2(i,j,k)}{2}} \end{aligned} \tag{2.169}$$

$$Z^1(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)x^{\frac{g_3(i,j,k,S_1)-g_3(i,j,k,S_2)+\Delta\delta_3(i,j,k)}{2}} y^{\frac{g_3(i,j,k,S_1)+g_3(i,j,k,S_2)-\Delta\delta_3(i,j,k)}{2}} \tag{2.170}$$

$$\begin{aligned} Z^b(i,j) &= \sum_{k=i+2}^{j-1} Z^m(i+1,k-1)Z^{m1}(k,j-1)e^{-\left[f_3(i,j)/k_BT\right]}x^{\frac{g_6(i,j,k,S_1)-g_6(i,j,k,S_2)+\Delta\delta_5(i,j,k)}{2}} y^{\frac{g_6(i,j,k,S_1)+g_6(i,j,k,S_2)-\Delta\delta_5(i,j,k)}{2}} \\ &+ \sum_{k=i+1}^{j-2}\sum_{l=k+1}^{j-1} Z^b(k,l)e^{-\left[f_2(i,j,k,l)/k_BT\right]}x^{\frac{g_5(i,j,k,l,S_1)-g_5(i,j,k,l,S_2)+\Delta\delta_4(i,j,k,l)}{2}} y^{\frac{g_5(i,j,k,l,S_1)+g_5(i,j,k,l,S_2)-\Delta\delta_4(i,j,k,l)}{2}} \\ &+ e^{-\left[f_1(i,j)/k_BT\right]}x^{\frac{g_4(i,j,S_1)-g_4(i,j,S_2)+\Delta\delta_1(i,j)}{2}} y^{\frac{g_4(i,j,S_1)+g_4(i,j,S_2)-\Delta\delta_1(i,j)}{2}} \end{aligned} \tag{2.171}$$

$$\begin{aligned} Z^m(i,j) &= \sum_{k=i}^{j-1}\Bigg\{ e^{-\left[f_4(k-i)/k_BT\right]}x^{\frac{g_7(i,j,k,S_1)-g_7(i,j,k,S_2)+\Delta\delta_6(i,j,k)}{2}} y^{\frac{g_7(i,j,k,S_1)+g_7(i,j,k,S_2)-\Delta\delta_6(i,j,k)}{2}} \\ &+ Z^m(i,k-1)x^{\frac{g_8(i,j,k,S_1)-g_8(i,j,k,S_2)+\Delta\delta_7(i,j,k)}{2}} y^{\frac{g_8(i,j,k,S_1)+g_8(i,j,k,S_2)-\Delta\delta_7(i,j,k)}{2}}\Bigg\} Z^{m1}(k,j) \end{aligned} \tag{2.172}$$

$$Z^{m1}(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)e^{-\left[f_4(j-k)/k_BT\right]}x^{\frac{g_3(i,j,k,S_1)-g_3(i,j,k,S_2)+\Delta\delta_3(i,j,k)}{2}} y^{\frac{g_3(i,j,k,S_1)+g_3(i,j,k,S_2)-\Delta\delta_3(i,j,k)}{2}} \tag{2.173}$$

where $\Delta\delta_k(\cdot)$ are:

$$\Delta\delta_1(i,j) = E_{S_{R_1},S_{R_2}}[i][j] \tag{2.174}$$

$$\Delta\delta_2(i,j,k) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[i][k] - E_{S_{R_1},S_{R_2}}[k+1][j] \tag{2.175}$$

$$\Delta\delta_3(i,j,k) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[i][k] \tag{2.176}$$

$$\Delta\delta_4(i,j,k,l) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[k][l] \tag{2.177}$$

$$\Delta\delta_5(i,j,k) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[i+1][k-1] - E_{S_{R_1},S_{R_2}}[k][j-1] \tag{2.178}$$

$$\Delta\delta_6(i,j,k) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[k][j] \tag{2.179}$$

$$\Delta\delta_7(i,j,k) = E_{S_{R_1},S_{R_2}}[i][j] - E_{S_{R_1},S_{R_2}}[i][k-1] - E_{S_{R_1},S_{R_2}}[k][j] \tag{2.180}$$

$E_{S_{R_1},S_{R_2}}[i][j]$ is defined by the Hamming distance of the partial structure vector:

$$E_{S_{R_1},S_{R_2}}[i][j] = \sum_{p=i}^{j-1} \sum_{q=p+1}^{j} S_{R_1}[p][q] \oplus S_{R_2}[p][q] \tag{2.181}$$

This is derived effectively by the following recursions:

Initialization:

$$E_{S_{R_1},S_{R_2}}[i][i] = 0 \qquad (1 \leq i \leq n) \tag{2.182}$$

$$E_{S_{R_1},S_{R_2}}[i][i+1] = S_{R_1}[i][i+1] \oplus S_{R_2}[i][i+1] \quad (1 \leq i \leq n-1) \tag{2.183}$$

Recursion $(1 \leq i \leq n-1, i+1 < j \leq n)$ **:**

$$\begin{aligned} E_{S_{R_1},S_{R_2}}[i][j] = S_{R_1}[i][j] \oplus S_{R_2}[i][j] + E_{S_{R_1},S_{R_2}}[i+1][j] \\ +E_{S_{R_1},S_{R_2}}[i][j-1] - E_{S_{R_1},S_{R_2}}[i+1][j-1] \end{aligned} \tag{2.184}$$

$\Delta\delta_k(\cdot)$ is the newly accumulated Hamming distance between two reference structures at each transition, and indicates the lower limit of $g_k(\cdot, S_1) + g_k(\cdot, S_2)$.

This formulation empirically increases the speed of calculation several times, depending on the RNA sequence and reference structures. It is always at least twice as fast as the original algorithm (see Appendix A for the proof).

Chapter 3

RESULTS

In this chapter, we present the results of analyzing the RNA secondary structure mainly using algorithms for the distribution of the Hamming distance from a specified structure. First, we review the performance of the implemented algorithms. Then, further analyses are used to demonstrate the utility of our technique.

3.1 Performance Evaluation

3.1.1 Comparison of Calculation Cost

We summarize the calculation cost for the Hamming distance distributions of the 1D and 2D algorithms. Non-DFT indicates that the algorithm applied all speedup techniques except DFT. The DFT-based algorithm is shown as a DFT approach. Since we can apply distributed processing to the DFT-based algorithm, we also describe the cost of multi-unit applications. Simultaneously, we show RNAbor and RNA2Dfold as examples of conventional objective software.

Table 3.1: Calculation cost of 1D algorithm.

	RNAbor	non DFT	DFT approach		
			single core	k units†	∞ units
Time	$O(n^3 d_{max}^2)$	$O(n^3 d_{max}^2)$	$O(n^3 d_{max})$	$O(\frac{n^3 d_{max}}{k})$	$O(n^3)$
Memory	$O(n^3)$	$O(n^3)$	$O(n^2)$	$O(n^2 k)$	$O(n^2 d_{max})$

† k should be around a divisor of d_{max}

Table 3.2: Calculation cost of 2D algorithm.

	RNA2Dfold	non DFT	DFT approach		
			single core	k units†	∞ units
Time	$O(n^7)$	$O(n^3 d_{1max}^2 d_{2max}^2)$	$O(n^3 d_{1max} d_{2max})$	$O(\frac{n^3 d_{1max} d_{2max}}{k})$	$O(n^3)$
Memory	$O(n^4)$	$O(n^2 d_{1max} d_{2max})$	$O(n^2)$	$O(n^2 k)$	$O(n^2 d_{1max} d_{2max})$

† k should be around a divisor of $d_{1max} d_{2max}$

3.1.2 Runtime Evaluation

We implemented a distributed processing application with OpenMP and evaluated the runtime of the calculations for the 1D and 2D Algorithms on a dual quad-core Intel® Xeon® E5540 @2.53GHz CPU with 17.6 GB RAM. The runtime was calculated from the mean of 10 random sequences, and we adopted a minimum free energy (MFE) structure as the reference structure. The second reference structure for the 2D algorithm was the open chain, which is a structure with no base pairing. We measured the runtime in this case with a single core, 8 cores and 8 threads, and 8 cores and 16 threads, though theoretically the process could be distributed to a maximum of d_{max} (1D) or $d_{1max} d_{2max}$ (2D). We also compared our algorithm with conventional software. RNA2Dfold in Vienna RNA version 2.0 supports OpenMP and was executed using 8 cores and 16 threads. RNAbor is available only on a web server, and the runtime of the non-DFT algorithm is therefore shown as a computationally equivalent algorithm.

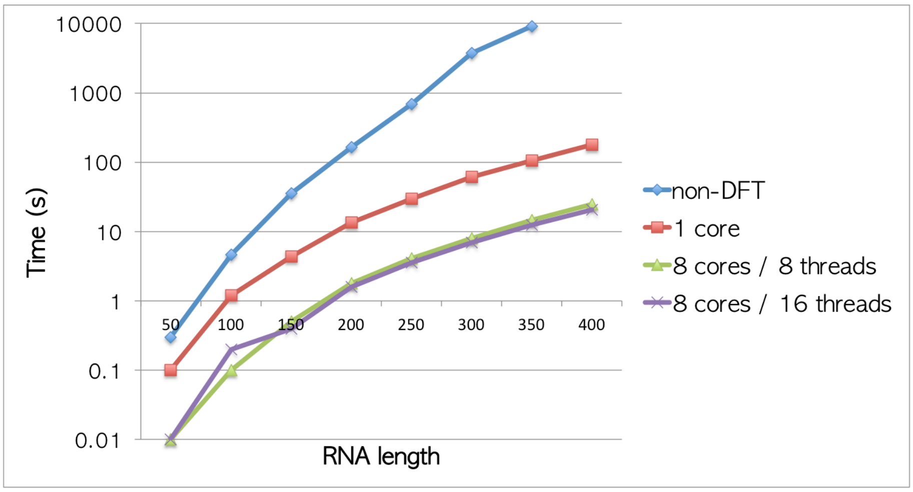


Figure 3.1: Run time of the 1D Algorithm.

Figure 3.1 suggests that our proposed 1D algorithm improved the calculation cost significantly compared with the non-DFT application. We achieved an approximately eightfold acceleration of the computing speed when using 8 cores, though hyper threading contributed little. Since it is rare to treat RNAs longer than 400nt when discussing secondary structures, our algorithm should be sufficiently fast for use in comprehensive analyses. The 2D algorithm also improved the runtime compared with the conventional method. However RNA2Dfold reduces the processing by utilizing the property of matrix sparseness, and our 2D algorithm might not be significantly different to RNA2Dfold when analyzing short RNA sequences. In fact, it may even be slightly worse (Figure 3.3), as the RNA2Dfold may be too time-consuming to be fully usable with longer RNAs. Because our methods can distribute the processing with almost no overhead, their speed increases as the computational resources increase.

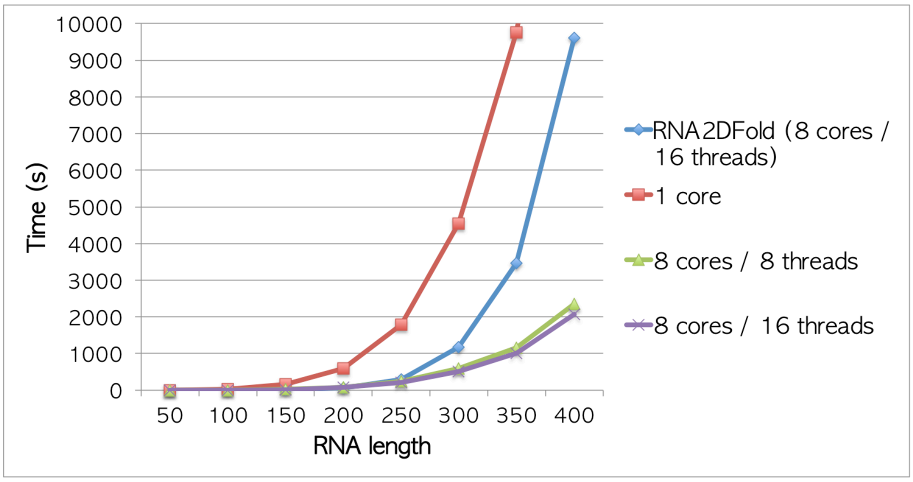


Figure 3.2: Runtime of 2D algorithm.

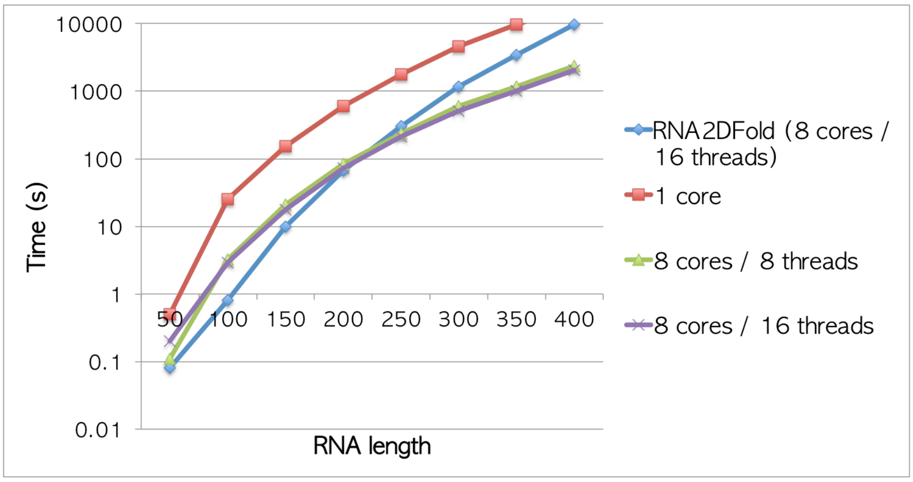


Figure 3.3: Semilog plot of Figure 3.2.

3.2 Applying the Algorithms to RNA sequences

We applied our algorithms to a range of RNA sequences. Sequences for the evaluation of ambiguity of conventional structure estimation methods were randomly chosen from Rfam for different RNA families. Rfam is one of the largest open access databases on RNA families and is hosted at the Wellcome Trust Sanger Institute [37]. For structure estimation, we used RNAfold [38] to calculate the minimum free energy structures (MFE structures) and CentroidFold [39] to calculate γ-centroid structures. According to the canonical distribution, the MFE structure is considered to be the most frequent among the whole ensemble(2.27). CentroidFold contains an arbitrary parameter γ, which regulates the weight of base pairs so that they are optimized for accuracy measures. In this study, we employed the default parameter $\gamma = 1$ if no previous notice was given. The γ-centroid structures with $\gamma = 1$ are also called centroid structures.

3.3 Unavoidable Ambiguity

In this section, we show the unavoidable ambiguity of RNA secondary structure estimation. First, we discuss the credibility limits of point estimated structures obtained by conventional methods. The credibility limit is an index of estimation uncertainty, where an α% credibility limit is defined as the minimum Hamming distance radius of a hyper-sphere containing $\alpha\%$ of the distribution [40]:

$$\alpha\% \text{ Credibility limit} = \min[d] \tag{3.1}$$

$$\text{s.t.} \sum_{d'=0}^{d} p(distance = d') \geq \frac{\alpha}{100}. \tag{3.2}$$

Since we can calculate this index using the results from our 1D algorithm, we evaluated the uncertainty of the two conventional estimation methods; the minimum free energy (MFE) and γ-centroid estimations.

Tables 3.3 and 3.4 list the credibility limits of the predicted structures for the different RNA families. Although MFE structures tend to have much higher probabilities than γ-centroid structures in proportion to the sequence length, the credibility limits suggest that MFE structures are not always more certain. For example, the MFE structure of RNaseP_nuc had an existence probability approximately 10^9 times larger than as that of the γ-centroid structures, but the 50% credibility limit of the MFE structure was much larger than the 95% credibility limit of the γ-centroid structure. Even when using the same estimation method, the credibility limits varied widely. Although the existence probabilities of the reference structures of Leu_leader were not as high, their credibility limits were very small even at the 95% credibility limit.

In contrast, some RNAs, such as tRNA or IRES_hcv, had quite large credibility limits compared with other RNAs of similar length, which suggest that they have several discrete structure clusters or lack significant stable structures. Although credibility limits have high-potency for filtering structure reliability, the complete distributions provided by our 1D algorithm enabled us to analyze in detail the whole structure distributions. Tables 3.5 and 3.6 show the results from our 1D algorithm, where the starting point on the left is the existence probability of a reference structure, and the end point on the right is the sum of the existence probabilities of the maximum Hamming distance structures. The scale of the x-axis is shown at the top. It can be seen that the γ-centroid structures tended to plot mono phasic distributions compared with the MFE structures. The principal reason is that the γ-centroid estimator chooses structures of similar distance from each structure cluster because many likely structures are widely spread, or several large clusters have approximately the same probabilities. If a point estimated structure is to be identified, the γ-centroid structure is more suitable since it tends to be a center of thermodynamic fluctuation. However, a potential danger is that locally stable functional structures may be overlooked. The distributions show that γ-centroid estimation may miss interesting structures for RNAs such as let-7, snoR1, or RatA. This suggests that MFE structures are more appropriate for seeking structure clusters.

Structures whose greatest probability concentrates around the origin are quite reliable, and we can expect there to be a relationship between the structure and its biological functions. However, RNAs which appear to have several discrete clusters (like MINT1_1) or no significant peaks (like IRES_HCV) suggest other analyses, for example that all or some of the clusters are related to their functions, the folding shapes are not important for their functions, they are so unstable that they are disassembled immediately, or that three dimensional folding contributes stability. In a later section, we discuss structure of some RNA families in detail.

Table 3.3: Credibility limit of various RNA families 1.

family	length	reference	Prob. of reference	50% CL	90% CL	95% CL
HIV_FS2	45	MFE	0.108002	3	7	10
		γ-centroid	0.108002	3	7	10
ROSE_2	73	MFE	0.000939968	9	20	21
		γ-centroid	9.30E-08	11	14	15
Xist_exon1	77	MFE	0.113858	3	9	22
		γ-centroid	0.0941742	3	9	22
let-7	82	MFE	0.00679927	7	11	13
		γ-centroid	1.01E-06	8	10	11
tRNA	85	MFE	0.000153493	39	41	41
		γ-centroid	0.00126515	29	30	30
snoR1	87	MFE	0.0138394	16	35	40
		γ-centroid	2.86E-11	20	25	27
RatA	92	MFE	0.000153156	24	27	28
		γ-centroid	5.69E-05	12	16	17
tRNA-Sec	92	MFE	0.041937	26	30	31
		γ-centroid	2.72E-09	21	23	24
sraA	94	MFE	0.00821323	17	33	39
		γ-centroid	5.74E-15	20	29	32
MINT1_1	97	MFE	8.59E-05	34	56	57
		γ-centroid	1.34E-13	25	46	47
5S_rRNA	121	MFE	0.0316842	11	22	27
		γ-centroid	0.000149788	8	18	29
NEAT1_1	121	MFE	0.00535367	11	24	27
		γ-centroid	9.66E-10	15	29	32
Hammerhead_HH10	126	MFE	0.0103061	7	33	43
		γ-centroid	1.28E-09	11	25	34
Leu_leader	148	MFE	0.0678398	3	7	10
		γ-centroid	0.000167593	3	7	9

Table 3.4: Credibility limit of various RNA families 2.

family	length	reference	Prob. of reference	50% CL	90% CL	95% CL
snoR134	150	MFE	7.09E-05	14	24	33
		γ-centroid	6.64E-16	44	46	47
AdoCbl_riboswitch	150	MFE	8.83E-05	27	34	36
		γ-centroid	5.62E-07	12	28	31
Pinc	154	MFE	0.00110353	18	30	32
		γ-centroid	1.47E-16	37	43	44
NrrF	157	MFE	0.00269495	27	40	41
		γ-centroid	3.62E-10	20	28	29
U1	161	MFE	0.0120026	9	26	28
		γ-centroid	3.58E-05	10	25	26
Collinsella-1	193	MFE	8.55E-06	59	63	65
		γ-centroid	3.19E-16	64	67	69
IRES_HCV	231	MFE	1.46E-08	106	126	128
		γ-centroid	≈0	64	76	78
PCA3_1	258	MFE	1.72E-11	101	116	120
		γ-centroid	≈0	59	64	66
RUF1	264	MFE	1.69E-08	55	77	82
		γ-centroid	1.24E-16	33	51	55
STnc150	267	MFE	3.06E-08	94	110	113
		γ-centroid	≈0	49	62	65
SCARNA13	273	MFE	0.000672026	19	83	98
		γ-centroid	≈0	65	71	80
RNaseP_nuc	341	MFE	3.57E-07	119	149	151
		γ-centroid	1.03E-16	70	80	83
Intron_gpI	342	MFE	1.15E-10	66	94	97
		γ-centroid	≈0	73	83	86

Table 3.5: 1D distributions of various RNA families 1.

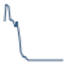
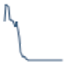
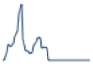
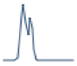
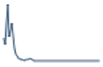
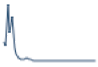
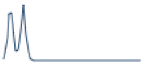
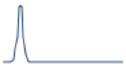
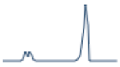
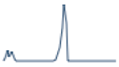
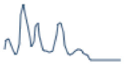
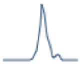
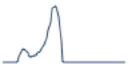
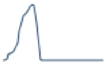
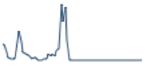
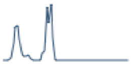
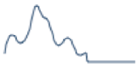
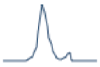
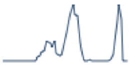
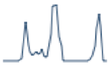
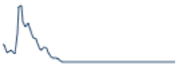
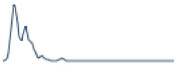
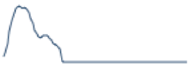
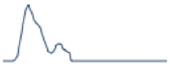
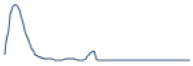
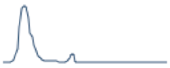
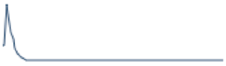
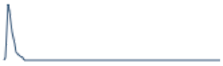
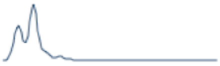
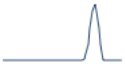
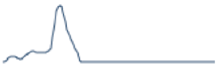
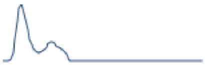
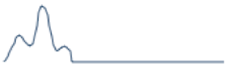
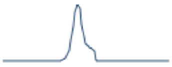
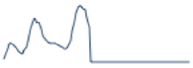
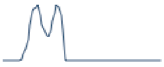
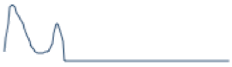
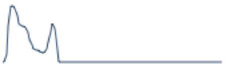
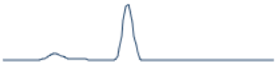
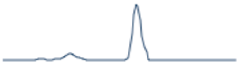
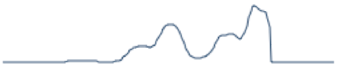
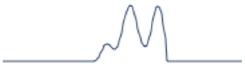
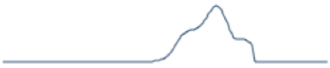
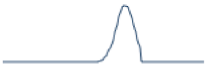
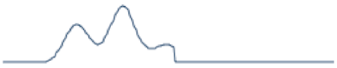
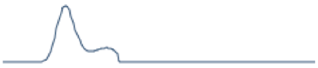
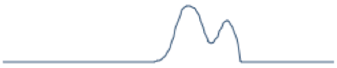
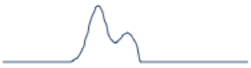
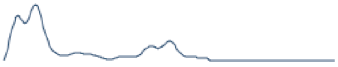
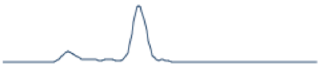
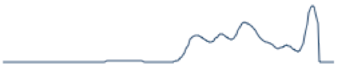
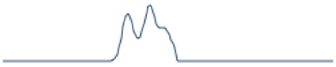
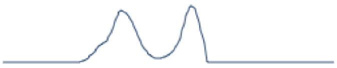
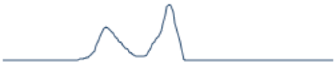
family	length	reference	
		MFE	γ-centroid
		0__________50_________100_________155	0__________50_________100_________155
HIV_FS2	45		
ROSE_2	73		
Xist_exon1	77		
let-7	82		
tRNA	85		
snoR1	87		
RatA	92		
tRNA-Sec	92		
sraA	94		
MINT1_1	97		
5S_rRNA	121		
NEAT1_1	121		
Hammerhead_HH10	126		
Leu_leader	148		

Table 3.6: 1D distributions of various RNA families 2.

family	length	reference	
		MFE	γ-centroid
		0__________50__________100__________155	0__________50__________100__________155
snoR134	150		
AdoCbl_riboswitch	150		
Pinc	154		
NrrF	157		
U1	161		
Collinsella-1	193		
IRES_HCV	231		
PCA3_1	258		
RUF1	264		
STnc150	267		
SCARNA13	273		
RNaseP_nuc	341		
Intron_gpI	342		

3.4 Model Selection Based on Credibility Limits

Those observations suggest that existence probabilities themselves do not guarantee reliability, and that credibility limits should be taken into account when a point estimated structure is required. Table 3.7 shows the credibility limits of various γ-centroid structures of SCARNA13, since CentroidFold can regulate the weight of the base pairs by the parameter γ. Indeed, the existence probability or free energy of a structure cannot be an index of its reliability, but credibility limits might provide a novel basis for parameter selection. In this case, the structure $\gamma = 0.25$ should provide the most reliable point estimation.

Table 3.7: Credibility limits of SCARNA13 γ-centroid structures.

γ	Prob. of reference	50% CL	90% CL	95% CL
0.03125	6.65863E-17	72	76	78
0.0625	9.5081E-17	67	72	73
0.125	≈0	63	69	71
0.25	1.48952E-16	62	68	70
0.5	≈0	66	71	75
1	≈0	65	71	80
2	≈0	68	76	86
4	1.83374E-8	72	83	94
6	2.3773E-8	74	85	96
8	1.12075E-12	75	87	98
16	1.81707E-11	80	93	104
32	7.23459E-15	85	97	109
64	≈0	87	99	111
128	≈0	89	101	113
512	1.41107E-18	91	103	115
MFE	6.72026E-4	19	83	98

We compared different γ γ-centroid structures ($\gamma = 2^k : k = -5, -4, \cdots, 10$) and determined the optimal γ for various RNAs based on the credibility limits. Table 3.8 shows the optimal γ and 50 or 95% credibility limits of the RNAs. The 95% credibility limits indicate that more credible structures are attained using the γ-centroid estimation, as MFE estimation is prone to bias from the point of view of the whole distribution. However, the γ-centroid estimator sometimes returns worse predictions at 50% credibility limits. Interestingly, the 50 and 95% credibility limits have similar values in such cases (see snoR134, Pinc, SCARNA13, or Intron_gpI for examples). These observations suggest the possibility that there are several discrete structure clusters and that the γ-centroid estimator incorrectly selects a structure on the potential barrier.

Table 3.8: Optimal γ based on credibility limits.

family	length	50% Credibility Limit			95% Credibility Limit		
		γ	γ-centroid	MFE	γ	γ-centroid	MFE
HIV_FS2	45	1	3	3	0.5	9	10
ROSE_2	73	4	10	9	1	15	21
Xist_exon1	77	1	3	3	0.25	21	22
let-7	82	1	8	7	1	11	13
tRNA	85	0.0625	25	39	0.0625	27	41
snoR1	87	2	19	16	2	26	40
RatA	92	1	12	24	1	17	28
tRNA-Sec	92	0.5	19	26	0.25	21	31
sraA	94	2	15	17	2	30	39
MINT1_1	97	0.5	19	34	0.0625	31	57
5S_rRNA	121	1	8	11	0.25	21	27
NEAT1_1	121	1	15	11	0.5	31	27
Hammerhead_HH10	126	16	8	7	0.5	25	43
Leu_leader	148	1	3	3	1	9	10
snoR134	150	1	44	14	1	47	33
AdoCbl_riboswitch	150	4	11	27	0.5	29	36
Pinc	154	0.25	36	18	0.25	40	32
NrrF	157	1	20	27	0.5	25	41
U1	161	2	9	9	0.5	23	28
Collinsella-1	193	0.0625	47	59	0.0625	51	65
IRES_HCV	231	1	64	106	0.25	72	128
PCA3_1	258	2	58	101	1	66	120
RUF1	264	1	33	55	1	55	82
STnc150	267	1	49	94	0.5	66	113
SCARNA13	273	0.25	62	19	0.25	70	98
RNaseP_nuc	341	2	67	119	0.5	83	151
Intron_gpI	342	1	73	66	0.125	85	97

3.5 Structure Validation Based on Credibility Limits

The credibility limits can be used for structure validation as well as for estimation model selection. We plotted the RNA sequences of various species annotated as hammerhead ribozyme HH9 from Rfam. These are shown in Figure 3.4, where the x and y axes are the 50 and 95% credibility limits from the reference structure defined in Rfam. It is reasonable that almost all the sequences have comparatively small credibility limits since the structure of Hammerhead ribozyme is closely related to its activity. However, the sequence of *Taeniopygia guttata* seems to lose its structural identity (the red arrowed point), which suggests that automatically annotated HH9 of *Taeniopygia guttata* might be not accurate, because of its structural instability.

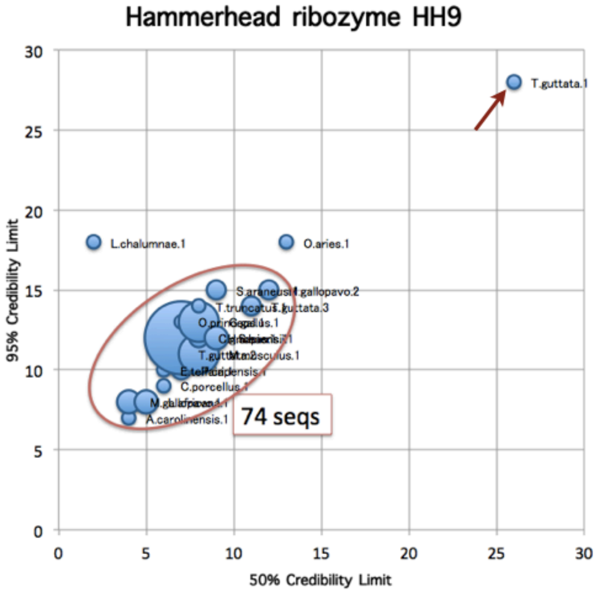
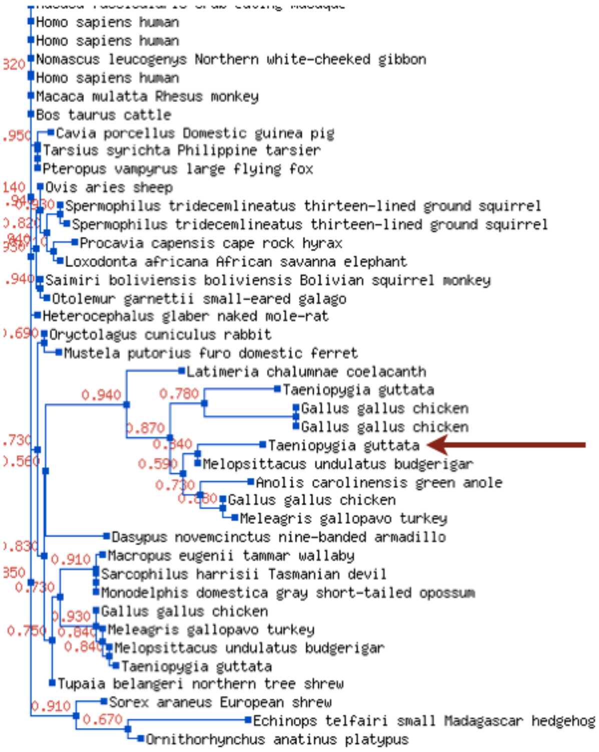


Figure 3.4: Credibility limits of the RNA sequences of various species annotated as hammerhead ribozyme.

However, it is difficult to distinguish the above sequence from the viewpoint of the sequence-based phylogenetic tree (Figure 3.5). The sequence of *Taeniopygia guttata* is indicated by the red arrow in Figure 3.5. This shows that sequence homology is insufficient for estimating the function, and that the use of credibility limits will increase the reliability of the annotation.



Homo sapiens human
Homo sapiens human
320
Nomascus leucogenys Northern white-cheeked gibbon
Homo sapiens human
Macaca mulatta Rhesus monkey
Bos taurus cattle
.950
Cavia porcellus Domestic guinea pig
Tarsius syrichta Philippine tarsier
Pteropus vampyrus large flying fox
140
Ovis aries sheep
0.930
Spermophilus tridecemlineatus thirteen-lined ground squirrel
0.820
Spermophilus tridecemlineatus thirteen-lined ground squirrel
Procavia capensis cape rock hyrax
Loxodonta africana African savanna elephant
.940
Saimiri boliviensis boliviensis Bolivian squirrel monkey
Otolemur garnettii small-eared galago
Heterocephalus glaber naked mole-rat
0.690
Oryctolagus cuniculus rabbit
Mustela putorius furo domestic ferret
Latimeria chalumnae coelacanth
0.940
0.780
Taeniopygia guttata
Gallus gallus chicken
Gallus gallus chicken
0.870
.730
0.840
Taeniopygia guttata
0.560
0.590
Melopsittacus undulatus budgerigar
0.730
Anolis carolinensis green anole
0.880
Gallus gallus chicken
Meleagris gallopavo turkey
Dasypus novemcinctus nine-banded armadillo
0.830
0.910
Macropus eugenii tammar wallaby
350
Sarcophilus harrisii Tasmanian devil
0.730
Monodelphis domestica gray short-tailed opossum
0.930
Gallus gallus chicken
0.750
0.840
Meleagris gallopavo turkey
0.840
Melopsittacus undulatus budgerigar
Taeniopygia guttata
Tupaia belangeri northern tree shrew
0.910
Sorex araneus European shrew
0.670
Echinops telfairi small Madagascar hedgehog
Ornithorhynchus anatinus platypus

Figure 3.5: A part of the phylogenetic tree of the sequences in Figure 3.4, drawn by PHYLIP[12].

3.6 Validation of Our Real Feature Value Model

In this section we briefly validate our real feature value model. The previous section showed the implementation of the algorithm for the free energy distribution of the RNA secondary structure ensemble. Figure 3.6 gives the energy distributions of the H/ACA snoRNA sequence, where the blue, red, and green lines represent the histogram based on the sampling technique, our proposed algorithm assuming the constant side lobe effect, and our proposed algorithm averaging the 10x higher resolution, respectively. Sampling was executed 500 times utilizing RNAsubopt[41][42] with the probabilistic traceback option. We can see that our models produced distributions that approximately coincided with the sampling technique. A negative probability was observed around -44.5kcal/mol in the red-colored distribution. This is considered to be bias introduced by the existence of the neighboring bin, whose probability is highly concentrated. The green-colored distribution indicates clearly that such bias vanishes when averaging the bins at a higher-resolution distribution.

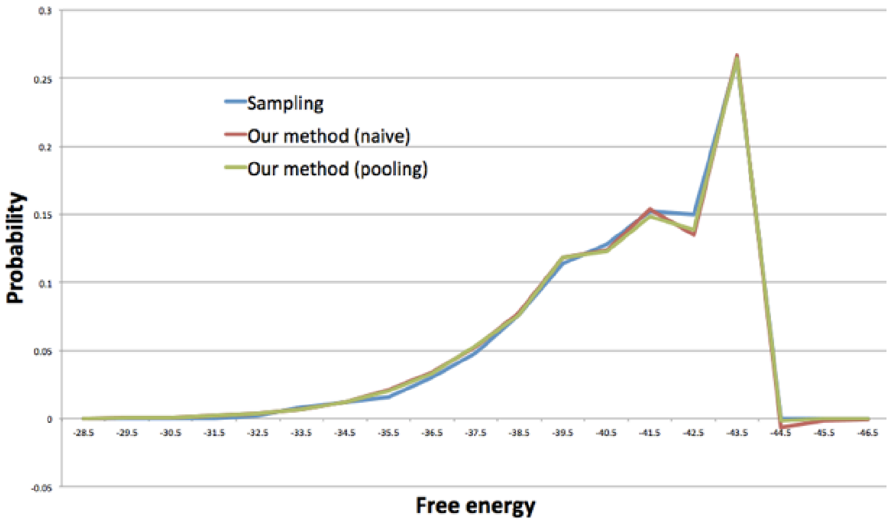


Figure 3.6: Free energy distribution of H/ACA snoRNA structure ensemble.

3.7 Application to RNA Families

In this section, we provide a more detailed analysis to confirm the significance of our distributions. First, a 1D distribution of H/ACA snoRNA demonstrated that the 1D distribution indicates the reliability or uncertainty of point estimated structures. Next, we demonstrated the potential of our method to find biologically meaningful structures which were missed by conventional methods by exemplifying the tRNA cloverleaf structure. Finally, we showed the 2D distributions of riboswitches when comparing two structure clusters.

3.7.1 H/ACA snoRNA

H/ACA snoRNA forms snoRNP by combining its specific sequence, called a box sequence, with proteins and guides pseudouridine modification of rRNAs [43]. Because its internal loop sequence is complementary to the target rRNAs, its structure is closely related to its activity as well as the sequence itself. Figure 3.7 shows the γ-centroid structure of H/ACA snoRNA.

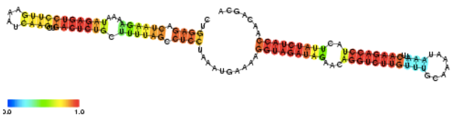


Figure 3.7: The γ-centroid structure of H/ACA snoRNA.

As noted above, we cannot understand the structural behavior of RNA by one estimated structure. Let us observe the probability distribution around the γ-centroid structure in our 1D algorithm. Figure 3.8 shows the structure existence probability distribution whose x-axis represents the Hamming distance from the γ-centroid structure, and Figure 3.9 gives the cumulative probability distribution.

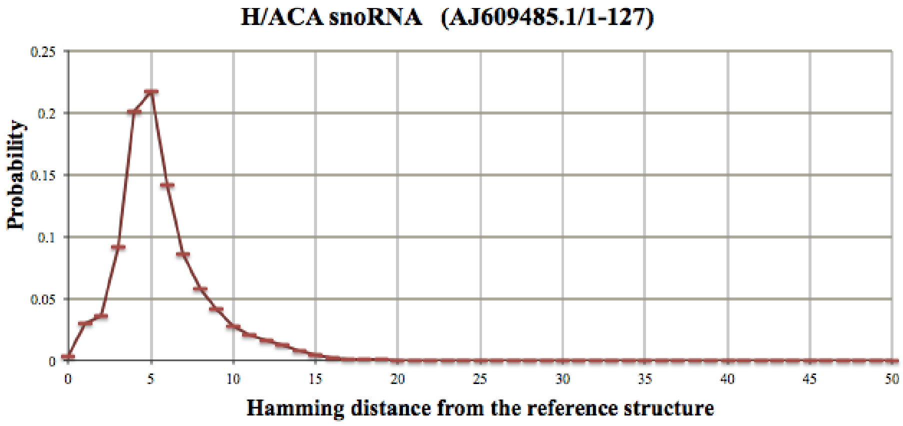


Figure 3.8: H/ACA snoRNA structure existence probability landscape from the γ-centroid structure.

We can observe that the probability that the RNA folds within 10 Hamming distances from the estimated structure is approximately 95% though the probability that this RNA folds into estimated γ-centroid structure itself is less than 1%. We might conclude that the γ-centroid structure represents the structural feature accurately enough. This representation can provide a novel estimator for the reliability of the estimated structure and the thermodynamic stability.

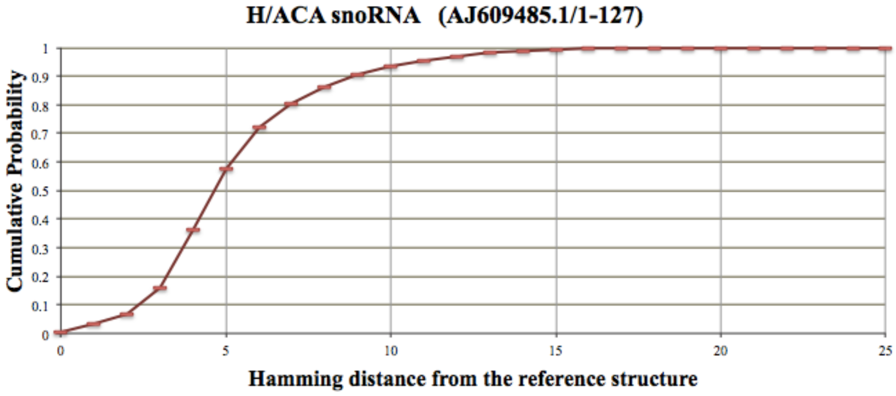


Figure 3.9: Cumulative probability distribution of Figure 3.8.

3.7.2 tRNA

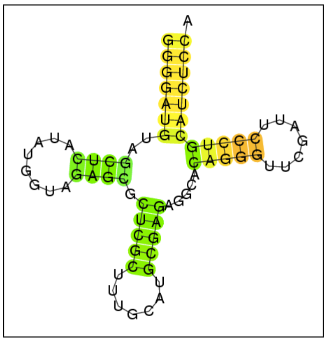


Figure 3.10: Cloverleaf structure.

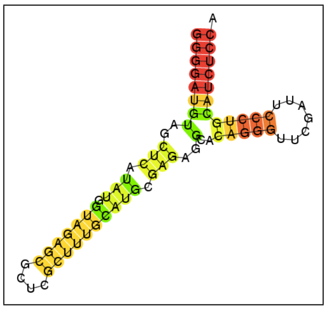


Figure 3.11: Centroid structure predicted by CentroidFold.

The secondary structure of tRNA is one of the best-known structures and is called the cloverleaf structure (Figure 3.10). However, even CentroidFold, which according to CompaRNA is one of the most accurate software products, does not always identify the cloverleaf structure (Figure 3.11). This biologically important structure would be missed if we did not have prior information on the structure.

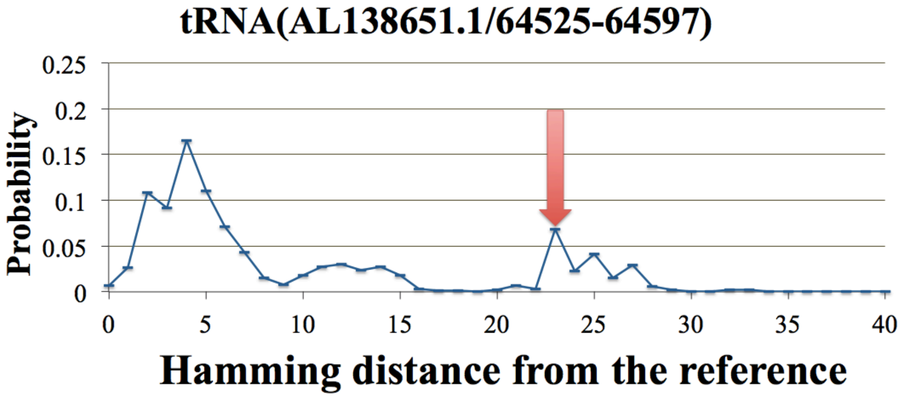


Figure 3.12: tRNA structure existence probability landscape from the γ-centroid structure.

We next show a probability distribution from our 1D algorithm whose reference structure is the above γ-centroid structure (Figure 3.12). The probability landscape suggests that this RNA might have sub-optimal structures around 25 nucleotides from the γ-centroid structure.

Confirmation is needed that there exists a sub-optimal structure cluster, because massive structures are included which are far from the origin. Figure 3.13 shows the number of structures at each distance. These are easily counted by applying our 1D algorithm (see Appendix C).

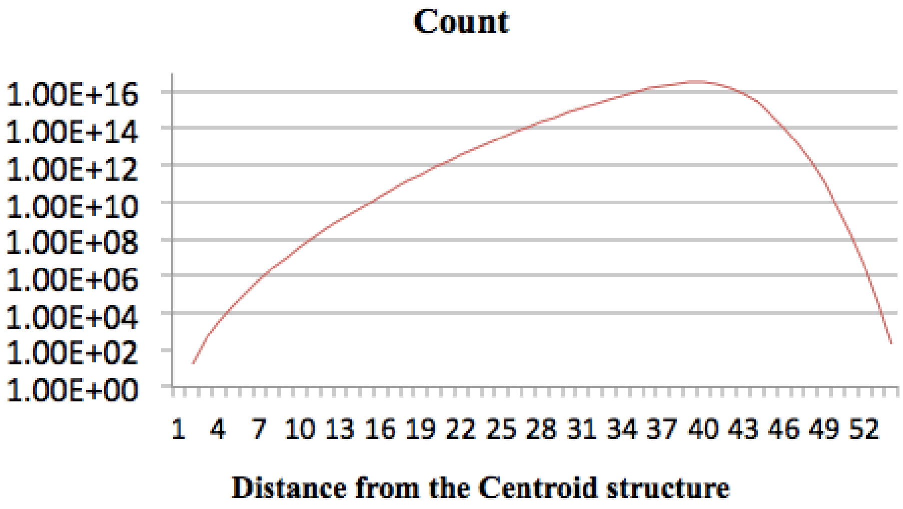


Figure 3.13: The number of possible structures.

It was confirmed that the estimated structure and its surroundings can explain only a limited proportion of the whole ensemble. In this case, we could identify the well-known cloverleaf structure around the secondary peak (shown by the red arrow). Using this cloverleaf structure as the reference, the existence probability could be drawn around the cloverleaf (Figure 3.14).

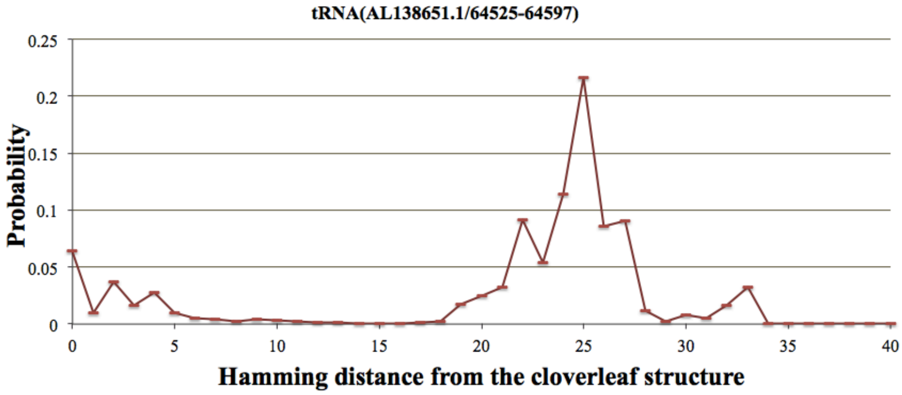


Figure 3.14: tRNA structure existence probability landscape from the cloverleaf structure.

Comparing the sum of probabilities around the secondary peak of Figure 3.12 and the origin of Figure 3.14, it can be seen that the observed peak is one structure cluster around the cloverleaf structure.

Figure 3.15 compares the γ-centroid structure and the cloverleaf structure. There appears to be a high potential barrier between the γ-centroid and the cloverleaf structure. Because the number of structures near the x and y axis is relatively small, we can guess the existence of two clusters and their relations more clearly than from the one dimensional analysis. Although the biological function of such a large structure cluster remains unclear, we might consider its relevance to tRNA base modification, which is known to contribute to structural stability [44][45].

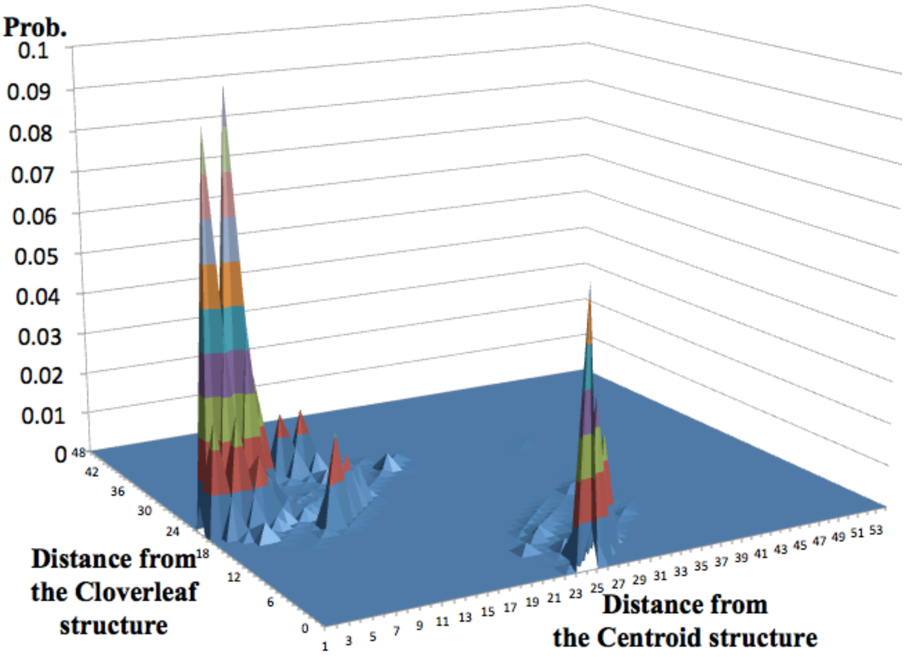


Figure 3.15: 2D extraction of tRNA structure existence probability landscape.

Figure 3.16 shows the distribution of the $5'-3'$ distance for the tRNA sequence. It can be seen that more than 99.7% of the structures have the same $5'-3'$ distance, although the 2D analysis suggested the presence of various structures in the ensemble. This tRNA is therefore expected to fold into a compact structure near the $5'-3'$ ends.

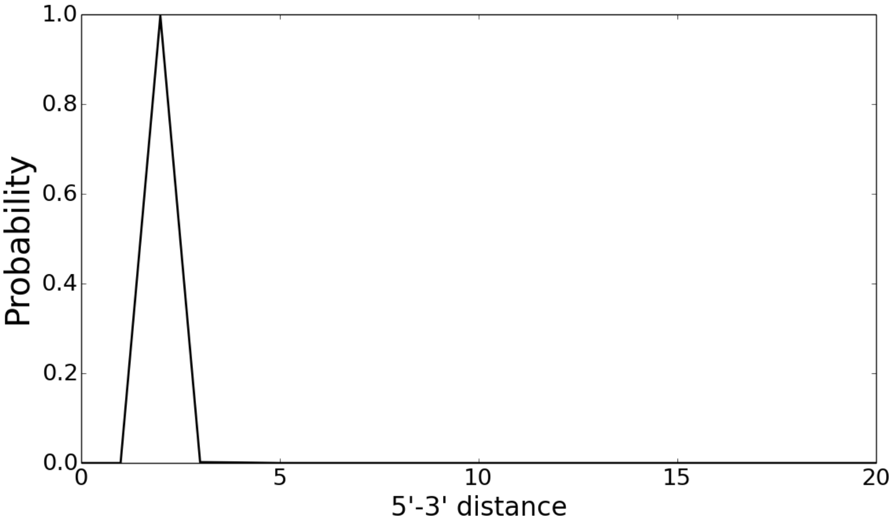


Figure 3.16: The $5' - 3'$ distance distribution of the tRNA.

3.7.3 Riboswitch

(1) SMK box translational riboswitch

Figure 3.17 shows the distribution of the SMK box translational riboswitch around its two important structures. This riboswitch is known to change its conformation dynamically in the presence of SAM [46], and the SAM$^+$ and SAM$^-$ structures correspond to these discrete peaks. The two clusters appear to be clearly separated, but there might exist a channel which associates the peaks, contrary to the distribution of tRNA.

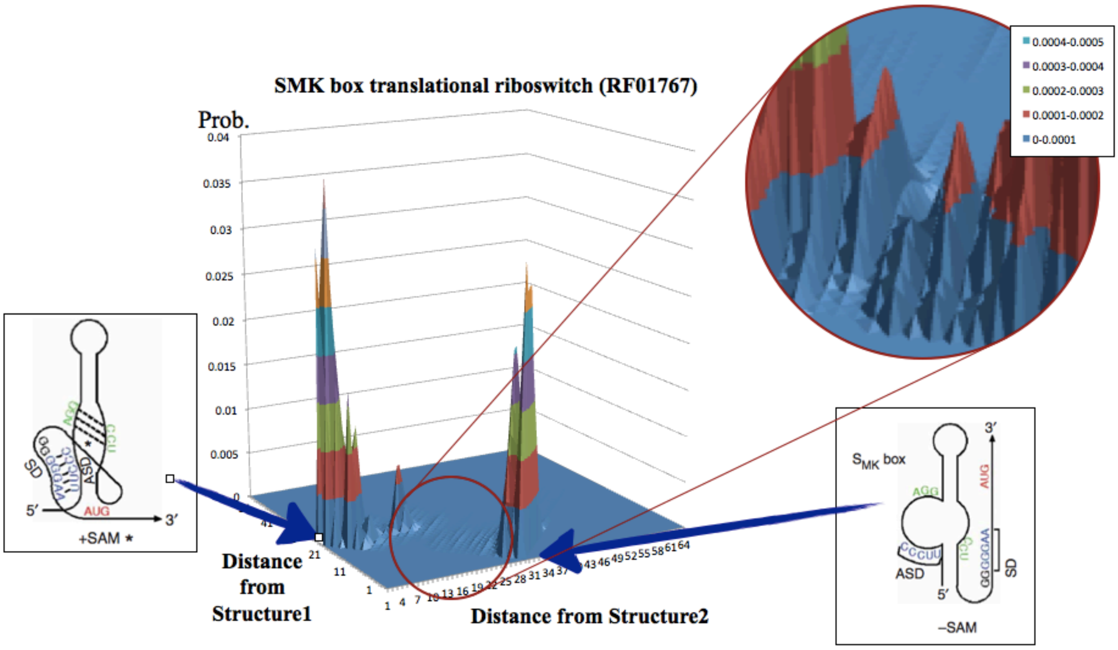


Figure 3.17: 2D distribution of SMK box translational riboswitch structure landscape around the SAM$^+$ and SAM$^-$ structures.

(2) TPP riboswitch

This example demonstrates the significance of our 2D extraction. Figure 3.18 shows the distribution derived by our 1D algorithm whose reference structure is its MFE structure. We can observe two overlapping peaks. We then chose another structure which has minimum free energy at the secondary peak, and drew a 2D distribution using the MFE and the chosen structure. In Figure 3.19, structure1 and structure2 correspond to the MFE and chosen structures, respectively. This figure indicates that at least not a little structures are apart from the primary structure cluster.

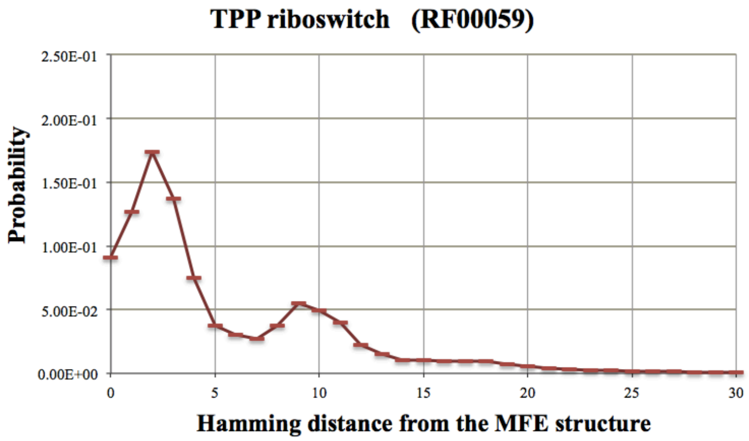


Figure 3.18: 1D distribution of TPP riboswitch structure landscape around the MFE structure.

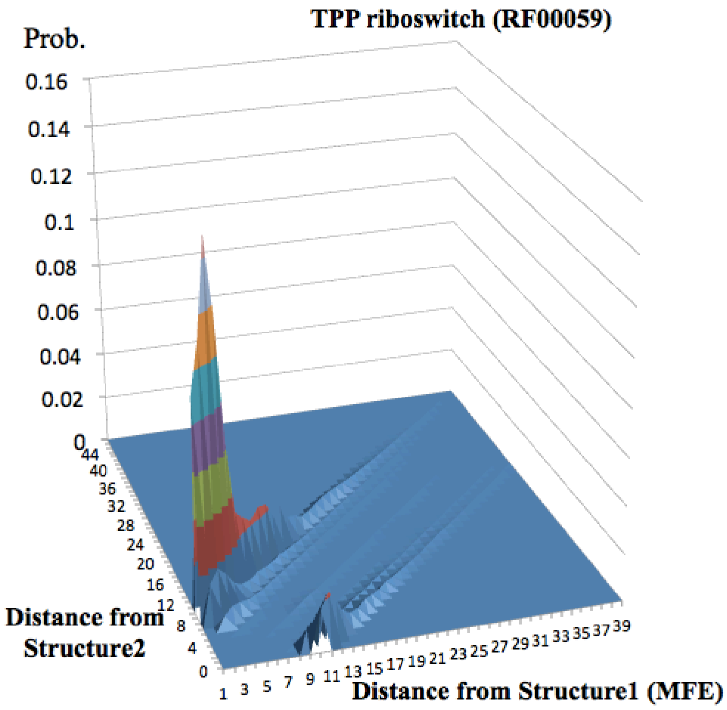


Figure 3.19: 2D distribution of TPP riboswitch structure landscape around MFE and local MFE structures.

Chapter 4

CONCLUSIONS AND DISCUSSION

4.1 Usefulness of Our Proposing Methods

As we have demonstrated, our proposed methods are sufficiently fast to be applied to practical RNA sequences and yield profound insights into RNA secondary structures. Our 1D algorithm can be used to evaluate the reliability and stability of a specific secondary structure, and credibility limits provide a suitable index for quantitative analysis. This index successfully revealed the unavoidable ambiguity of point estimated structures. We therefore evaluated the reliability by our method when estimating a secondary structure. The 1D algorithm also proved useful in finding biologically important structures which were not detected by conventional point estimation methods. If a second peak is found in the 1D distribution, its properties can be specified by the following procedure. First, we apply a stochastic sampling technique like RNAsubopt to the object, and extract the structures which belong to the secondary peak by counting the Hamming distance from the reference. Then, we use each structure as a reference and recalculate the 1D distribution. The constitution of the secondary peak is unmasked by combining the Hamming distance between each structure and the level of concentration around the structures.

Our 2D algorithm allows comparison of several structures of interest. 2D distributions imply communicability of structures and the possibility that another structure cluster exists. By applying these algorithms, we can construct an evaluation method for sub-optimal structures by combining sampling and clustering techniques.

We can also extend the range of analysis, as well as estimating the RNA folding pattern itself. For example, the expression level might be required to correct the bias according to its distribution of structures because biologically functional structures might represent only a part of the whole distribution. We can also observe the structure distribution dynamics in response to temperature change by altering the temperature parameter. A similar method to our 1D algorithm was recently published[47], but we believe there remains great scope for the application of our method to biological analysis. We are now applying our method to a large human ncRNA dataset from GENCODE v.13 and Rfam to categorize RNAs from the viewpoint of thermal fluctuation or the existence of sub-optimal structures.

4.2 Applying Our Strategies to Other Fields

As the work of Newberg *et al.* has already suggested, the general theory behind our algorithm on Algorithm 7 has the potential to be applied to a range of problems in bioinformatics [30]. It can be used for calculating any distribution with feature values accompanying the transition and emission probabilities of general dynamic programming problems, including Hidden Markov Models, which are frequently used in the field of bioinformatics. We have shown here the solution of a problem related to RNA secondary structure as an example of the applications of our general theory, and to exemplify the construction of a concrete algorithm.

Our proposed real feature expansion is also expected to have useful applications. An example would be the evaluation of the credibility of chromHMM[48] output by defining a distance matrix, as the distance between chromosome states is naturally defined by a real number. We also intend to implement more pragmatic applications than the free energy of RNA secondary structures.

Acknowledgment

My heartfelt appreciation goes to Prof. Kiyoshi Asai whose insightful comments were invaluable during the course of my study. Advice given by Prof. Michiaki Hamada, Dr. Martin Frith, Dr. Tomoshi Kameda, and Dr. Junichi Iwakiri has also been a great help in my study. I would also like to express my gratitude to Prof. Tetsuro Shibuya, Prof. Shinichi Morishita, Prof. Hisanori Kiryu, and Prof. Koji Tsuda for their constructive feedback. I also thank Dr. Toutai Mituyama and Dr. Yukiteru Ono for their help in integration of the software to the web page. Ivo Hofacker *et al.*, who developed Vienna RNA Package lightened the implementation burden drastically. I also express special thanks to all the members of the Asai Lab. and CBRC who gave me warm encouragements. I would also like to express my gratitude to my family for their moral support and tolerance. Finally, I would like to thank Grant-in-Aid for JSPS Fellows for their financial support.

Bibliography

[1] S B Needleman and C D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–453, Mar 1970.

[2] R Nussinov and A B Jacobson. Fast algorithm for predicting the secondary structure of single-stranded rna. *Proc Natl Acad Sci U S A*, 77(11):6309–6313, Nov 1980.

[3] J Skolnick, A Kolinski, C L 3rd Brooks, A Godzik, and A Rey. A method for predicting protein structure from sequence. *Curr Biol*, 3(7):414–423, Jul 1993.

[4] Mark Yandell and Daniel Ence. A beginner's guide to eukaryotic genome annotation. *Nat Rev Genet*, 13(5):329–342, May 2012.

[5] Jennifer Ripplinger and Jack Sullivan. Assessment of substitution model adequacy using frequentist and bayesian methods. *Mol Biol Evol*, 27(12):2790–2803, Dec 2010.

[6] Yuki Kato, Kengo Sato, Michiaki Hamada, Yoshihide Watanabe, Kiyoshi Asai, and Tatsuya Akutsu. Ractip: fast and accurate prediction of rna-rna interaction using integer programming. *Bioinformatics*, 26(18):i460–6, Sep 2010.

[7] Matea Hajnic, Juan Iregui Osorio, and Bojan Zagrovic. Computational analysis of amino acids and their sidechain analogs in crowded solutions of rna nucleobases with implications for the mrna-protein complementarity hypothesis. *Nucleic Acids Res*, 42(21):12984–12994, Dec 2014.

[8] Arunkumar Venkatesan, Sameer Hassan, Kannan Palaniyandi, and Sujatha Narayanan. In silico and experimental validation of protein-protein interactions between pkni and rv2159c from mycobacterium tuberculosis. *J Mol Graph Model*, 62:283–293, Oct 2015.

[9] Kyung-Ah Sohn, Joshua W K Ho, Djordje Djordjevic, Hyun-Hwan Jeong, Peter J Park, and Ju Han Kim. hihmm: Bayesian non-parametric joint inference of chromatin state maps. *Bioinformatics*, 31(13):2066–2074, Jul 2015.

[10] M Zuker. On finding all suboptimal foldings of an rna molecule. *Science*, 244(4900):48–52, Apr 1989.

[11] Danny Barash and Alexander Churkin. Mutational analysis in rnas: comparing programs for rna deleterious mutation prediction. *Brief Bioinform*, 12(2):104–114, Mar 2011.

[12] J. Felsenstein. Phylip - phylogeny inference package (version 3.2). *Cladistics*, 5:164–166, 1989.

[13] Michiaki Hamada, Hisanori Kiryu, Wataru Iwasaki, and Kiyoshi Asai. Generalized centroid estimators in bioinformatics. *PLoS One*, 6(2):e16450, 2011.

[14] Luis E Carvalho and Charles E Lawrence. Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proc Natl Acad Sci U S A*, 105(9):3209–3214, Mar 2008.

[15] Martin C Frith, Ryota Mori, and Kiyoshi Asai. A mostly traditional approach improves alignment of bisulfite-converted dna. *Nucleic Acids Res*, 40(13):e100, Jul 2012.

[16] Christina Piperi and Athanasios G Papavassiliou. Strategies for dna methylation analysis in developmental studies. *Dev Growth Differ*, 53(3):287–299, Apr 2011.

[17] O.V. Dyachenko, T.V. Shevchuk, and Ya.I. Buryanov. Structural and functional features of the 5-methylcytosine distribution in the eukaryotic genome. *Molecular Biology*, 44(2):171–185, 2010.

[18] H M Martinez. Detecting pseudoknots and other local base-pairing structures in rna sequences. *Methods Enzymol*, 183:306–317, 1990.

[19] Eleonora De Leonardis, Benjamin Lutz, Sebastian Ratz, Simona Cocco, Remi Monasson, Alexander Schug, and Martin Weigt. Direct-coupling analysis of nucleotide coevolution facilitates rna secondary and tertiary structure prediction. *Nucleic Acids Res*, 43(21):10444–10455, Dec 2015.

[20] T. Puton, K. Rother, L. Kozlowski, E. Tkalinska, and J. Bujnicki. A server for continuous benchmarking of automated methods for RNA structure prediction, 2011. `http://comparna.amu.edu.pl/`.

[21] Sabrina Lusvarghi, Joanna Sztuba-Solinska, Katarzyna J Purzycka, Jason W Rausch, and Stuart F J Le Grice. Rna secondary structure prediction using high-throughput shape. *J Vis Exp*, (75):e50243, 2013.

[22] Joseph M Watts, Kristen K Dang, Robert J Gorelick, Christopher W Leonard, Julian W Jr Bess, Ronald Swanstrom, Christina L Burch, and Kevin M Weeks. Architecture and secondary structure of an entire hiv-1 rna genome. *Nature*, 460(7256):711–716, Aug 2009.

[23] Henry van den Bedem and James S Fraser. Integrative, dynamic structural biology at atomic resolution–it's about time. *Nat Methods*, 12(4):307–318, Apr 2015.

[24] D Marion and K Wuthrich. Application of phase sensitive two-dimensional correlated spectroscopy (cosy) for measurements of 1h-1h spin-spin coupling constants in proteins. *Biochem Biophys Res Commun*, 113(3):967–974, Jun 1983.

[25] T de Beer, C W van Zuylen, K Hard, R Boelens, R Kaptein, J P Kamerling, and J F Vliegenthart. Rapid and simple approach for the nmr resonance assignment of the carbohydrate chains of an intact glycoprotein. application of gradient-enhanced natural abundance 1h-13c hsqc and hsqc-tocsy to the alpha-subunit of human chorionic gonadotropin. *FEBS Lett*, 348(1):1–6, Jul 1994.

[26] D Marion, M Genest, and M Ptak. Reconstruction of noesy maps. a requirement for a reliable conformational analysis of biomolecules using 2d nmr. *Biophys Chem*, 28(3):235–244, Dec 1987.

[27] Kengo Sato, Yuki Kato, Michiaki Hamada, Tatsuya Akutsu, and Kiyoshi Asai. Ipknot: fast and accurate prediction of rna secondary structures with pseudoknots using integer programming. *Bioinformatics*, 27(13):i85–93, Jul 2011.

[28] Waterman MS. *Introduction to Computational Biology: Maps, Sequences and Genomes.* Chapman & Hall, London, 1995.

[29] Y. Ding, C. Y. Chan, and C. E. Lawrence. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, 11(8):1157–1166, Aug 2005.

[30] L. A. Newberg and C. E. Lawrence. Exact calculation of distributions on integers, with application to sequence alignment. *J. Comput. Biol.*, 16(1):1–18, Jan 2009.

[31] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990.

[32] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. U.S.A.*, 101(19):7287–7292, May 2004.

[33] D. H. Turner and D. H. Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, 38(Database issue):D280–282, Jan 2010.

[34] A. M. Yoffe, P. Prinsen, W. M. Gelbart, and A. Ben-Shaul. The ends of a large RNA molecule are necessarily close. *Nucleic Acids Res.*, 39(1):292–299, Jan 2011.

[35] H. S. Han and C. M. Reidys. The 5′-3′ distance of RNA secondary structures. *J. Comput. Biol.*, 19(7):867–878, Jul 2012.

[36] P. Clote, Y. Ponty, and J. M. Steyaert. Expected distance between terminal nucleotides of RNA secondary structures. *J Math Biol*, 65(3):581–599, Sep 2012.

[37] P. P. Gardner, J. Daub, J. G. Tate, E. P. Nawrocki, D. L. Kolbe, S. Lindgreen, A. C. Wilkinson, R. D. Finn, S. Griffiths-Jones, S. R. Eddy, and A. Bateman. Rfam: updates to the RNA families database. *Nucleic Acids Res.*, 37(Database issue):D136–140, Jan 2009.

[38] I. L. Hofacker and P. F. Stadler. Memory efficient folding algorithms for circular RNA secondary structures. *Bioinformatics*, 22(10):1172–1176, May 2006.

[39] M. Hamada, H. Kiryu, K. Sato, T. Mituyama, and K. Asai. Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25(4):465–473, Feb 2009.

[40] B. J. Webb-Robertson, L. A. McCue, and C. E. Lawrence. Measuring global credibility with application to local sequence alignment. *PLoS Comput. Biol.*, 4(5):e1000077, May 2008.

[41] R. Lorenz, S. H. Bernhart, C. Honer Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. ViennaRNA Package 2.0. *Algorithms Mol Biol*, 6:26, 2011.

[42] S Wuchty, W Fontana, I L Hofacker, and P Schuster. Complete suboptimal folding of rna and the stability of secondary structures. *Biopolymers*, 49(2):145–165, Feb 1999.

[43] J. Ni, A. L. Tien, and M. J. Fournier. Small nucleolar RNAs direct site-specific synthesis of pseudouridine in ribosomal RNA. *Cell*, 89(4):565–573, May 1997.

[44] L. A. Copela, G. Chakshusmathi, R. L. Sherrer, and S. L. Wolin. The La protein functions redundantly with tRNA modification enzymes to ensure tRNA structural stability. *RNA*, 12(4):644–654, Apr 2006.

[45] F. Tuorto, R. Liebers, T. Musch, M. Schaefer, S. Hofmann, S. Kellner, M. Frye, M. Helm, G. Stoecklin, and F. Lyko. RNA cytosine methylation by Dnmt2 and NSun2 promotes tRNA stability and protein synthesis. *Nat. Struct. Mol. Biol.*, 19(9):900–905, Sep 2012.

[46] R. T. Fuchs, F. J. Grundy, and T. M. Henkin. The S(MK) box is a new SAM-binding RNA for translational regulation of SAM synthetase. *Nat. Struct. Mol. Biol.*, 13(3):226–233, Mar 2006.

[47] E. Senter, S. Sheikh, I. Dotu, Y. Ponty, and P. Clote. Using the fast fourier transform to accelerate the computational search for RNA conformational switches. *PLoS ONE*, 7(12):e50506, 2012.

[48] Jason Ernst and Manolis Kellis. Chromhmm: automating chromatin-state discovery and characterization. *Nat Methods*, 9(3):215–216, Mar 2012.

[49] R. Nussinov, G. Pieczenik, J. R. Griggs, and D. J. Kleitman. Algorithms for loop matching. *SIAM J Appl Math*, 35(1):68–82, Jul 1978.

Appendix A

Proofs

A.1 Maximum Hamming Distance between Structure Vectors

We prove here that the Hamming distance d (equation (2.49)) never exceeds its sequence length n. First, structure vectors have upper limit of non-zero elements:

$$0 \leq ||S_1||_1 \leq \left\lfloor \frac{n}{2} \right\rfloor, 0 \leq ||S_2||_1 \leq \left\lfloor \frac{n}{2} \right\rfloor \tag{A.1}$$

Thus:

$$\begin{aligned} d &= \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} S_1[i][j] \oplus S_2[i][j] \\ &= ||S_1 - S_2||_1 \leq ||S_1||_1 + |-1|||S_2||_1 \leq 2 \left\lfloor \frac{n}{2} \right\rfloor \leq n \end{aligned} \tag{A.2}$$

A.2 Lower Limit of Acceleration Rate by (2)

Our original and modified 2D algorithm requires $d_{1max}d_{2max}$ and $\frac{\delta(d_{1max}+d_{2max}-\delta)}{2}$ -time continuous calculations respectively. Thus, our modification contributes the following acceleration rate r:

$$r = \frac{d_{1max}d_{2max}}{\frac{\delta(d_{1max}+d_{2max}-\delta)}{2}} = \frac{2d_{1max}d_{2max}}{\delta(d_{1max}+d_{2max}-\delta)} \tag{A.3}$$

r is a monotonic decrease function of δ from $\delta = 0$ to $\delta = \frac{d_{1max}+d_{2max}}{2}$:

$$\frac{\partial r}{\partial \delta} = \frac{2\delta - d_{1max} - d_{2max}}{(\delta(d_{1max}+d_{2max}-\delta))^2} \leq 0 \quad (0 < \delta \leq \tfrac{d_{1max}+d_{2max}}{2}) \tag{A.4}$$

On the other hand, we have the following inequality from equations (2.166) - (2.167) and the property of arithmetic mean:

$$\delta \leq \min(d_{1max}, d_{2max}) \leq \frac{d_{1max}+d_{2max}}{2} \tag{A.5}$$

Accordingly, we obtain the lower limit of r as follows:

$$\begin{aligned} r &= \frac{2d_{1max}d_{2max}}{\delta(d_{1max}+d_{2max}-\delta)} \\ &\geq \frac{2d_{1max}d_{2max}}{\min(d_{1max}, d_{2max})(d_{1max}+d_{2max}-\min(d_{1max}, d_{2max}))} = 2 \end{aligned} \tag{A.6}$$

Appendix B

A Detailed Explanation of the McCaskill Model

The origin of the McCaskill model is a simple classical dynamic programming algorithm by Nussinov and colleagues [49], which maximizes the number of base pairs. The McCaskill model calculates the whole energy contribution instead of counting base pairs. Here we provide a detailed explanation since this model is closely related to the foundation of our algorithms. First, we describe interpretations of $Z(i, j)$ and $Z^{\bullet}(i, j)$.

1. $Z(i, j)$ is the summation of energy contribution of all possible structures from i to j. Our ultimate goal is to obtain $Z(1, n)$ as partition function Z.

2. $Z^{b}(i, j)$ is the summation of energy contribution of all the possible structures from i to j under the condition of $S[i][j] = 1$.

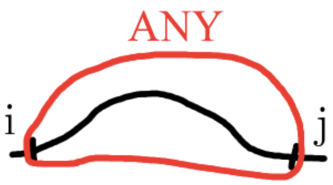


Figure B.1: $Z(i, j)$.

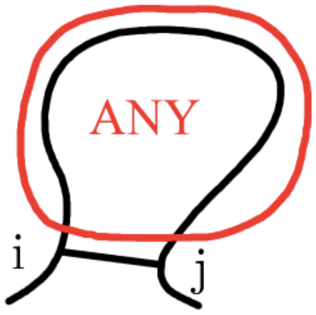


Figure B.2: $Z^{b}(i, j)$.

3. $Z^1(i, j)$ is the summation of energy contribution of all the possible structures from i to j under the condition that the i-th base makes a base pair with the k-th base and no pairs from $k + 1$ to j.

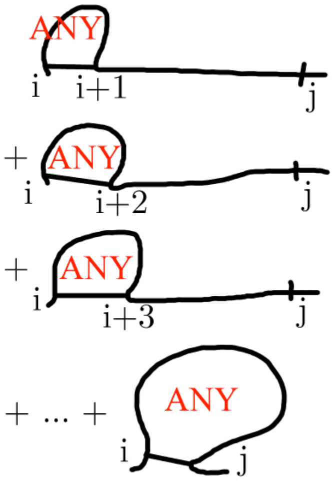


Figure B.3: $Z^1(i, j)$.

4. $Z^m(i, j)$ is the summation of energy contribution of all the possible structures from i to j under the condition that they are in a multi-loop and include at least one base pair.

5. $Z^{m1}(i, j)$ is the same with $Z^1(i, j)$ except that they are in a multi-loop.

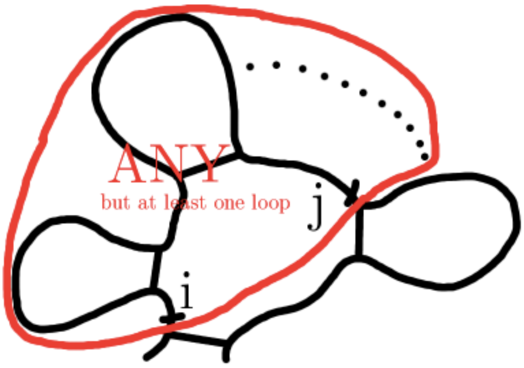


Figure B.4: $Z^m(i, j)$.

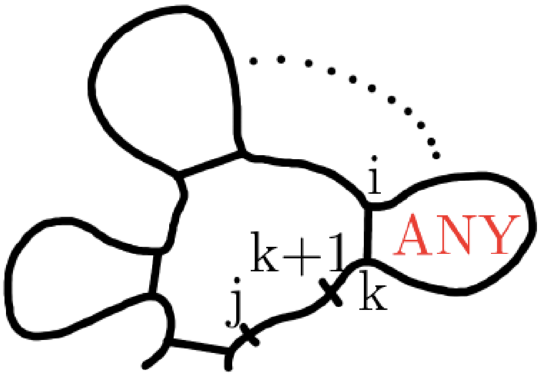


Figure B.5: $Z^{m1}(i, j)$.

Now we can understand the meaning of each recursion.

1. $Z(i,j) = 1.0 + \sum_{k=i}^{j-1} Z(i,k)Z^1(k+1,j)$

   The first term corresponds to the open chain. The other term means the case that they have at least one base pair whose the most right side loop begins with the $(k+1)$-th base.

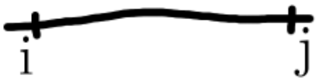


Figure B.6: 1.0.

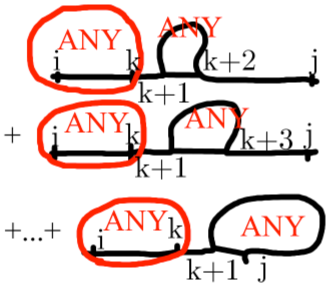


Figure B.7: $Z(i,k)Z^1(k+1,j)$.

2. $Z^b(i,j) = e^{-[f_1(i,j)/k_BT]} + \sum_{k=i+1}^{j-2} \sum_{l=k+1}^{j-1} Z^b(k,l)e^{-[f_2(i,j,k,l)/k_BT]}$
   $+ \sum_{k=i+2}^{j-1} Z^m(i+1,k-1)Z^{m1}(k,j-1)e^{-[f_3(i,j)/k_BT]}$

   These three terms correspond to the states of hairpin loop, internal or bulge loop, and multi-loop respectively. Here, $e^{-[f_1(i,j)/k_BT]}$ is the hairpin loop energy contribution and $e^{-[f_2(i,j,k,l)/k_BT]}$ is the internal or bulge loop energy contribution. The energy contribution of multi-loop is included in $Z^m(i+1,k-1)Z^{m1}(k,j-1)$ for the most part, but $e^{-[f_3(i,j)/k_BT]}$ represents the multi-loop energy around i, j.

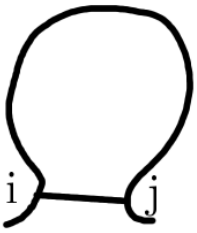


Figure B.8: $e^{-[f_1(i,j)/k_BT]}$.

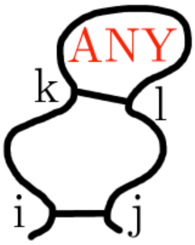


Figure B.9: $Z^b(k,l)e^{-[f_2(i,j,k,l)/k_BT]}$.

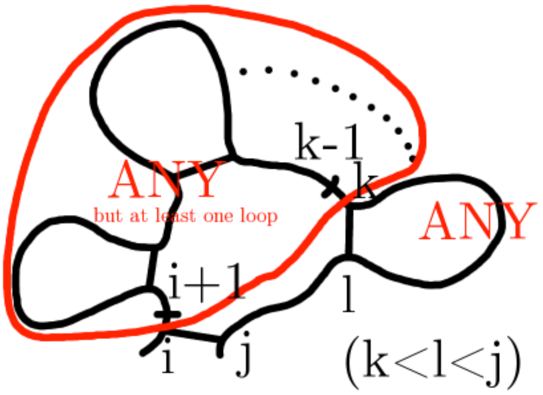


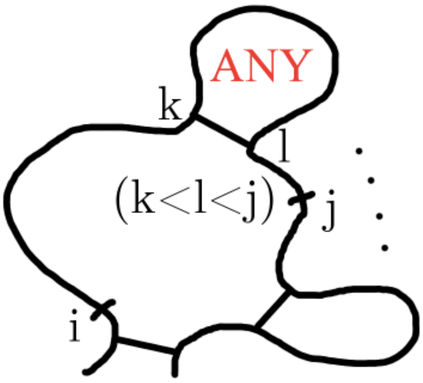
Figure B.10: $Z^m(i+1,k-1)Z^{m1}(k,j-1)e^{-[f_3(i,j)/k_BT]}$.

3. $Z^1(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)$
4. $Z^{m1}(i,j) = \sum_{k=i+1}^{j} Z^b(i,k)e^{-[f_4(j-k)/k_BT]}$

   They can be easily derived from their definitions. The only difference between these equations is $e^{-[f_4(j-k)/k_BT]}$, which is a part of multi-loop energy contribution from $k+1$ to j (see Figure B.5).

5. $Z^m(i,j) = \sum_{k=i}^{j-1}\left(e^{-[f_4(k-i)/k_BT]} + Z^m(i,k-1)\right)Z^{m1}(k,j)$
   $= \sum_{k=i}^{j-1} e^{-[f_4(k-i)/k_BT]}Z^{m1}(k,j) + \sum_{k=i}^{j-1} Z^m(i,k-1)Z^{m1}(k,j)$

   These terms correspond to the case of including only one loop or at least two loops from i to j, respectively. Here, $e^{-[f_4(k-i)/k_BT]}$ is the energy contribution of open chain in a multi-loop from i to $k-1$. This recursive representation enables to contain any number of loops in $Z^m(i,j)$.



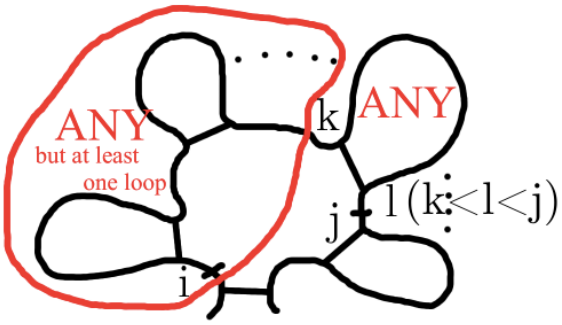


Figure B.11: $e^{-[f_4(k-i)/k_BT]}Z^{m1}(k,j)$.

Figure B.12: $Z^m(i,k-1)Z^{m1}(k,j)$.

Appendix C

Counting the Number of Structures

We can count the number of structures included in each Hamming distance by remodeling our developed algorithms. We show here only the 1D algorithm, but of course, we can extend this to the 2D algorithm.

Initialization $(1 \leq i \leq n)$ **:**

$$
\begin{aligned}
N(i,i) &= 1 && \text{(C.1)}\\
N^1(i,i) &= N^b(i,i) = N^m(i,i) = N^m(i,i-1) = 0 && \text{(C.2)}
\end{aligned}
$$

Recursion $(1 \leq i < j \leq n)$ **:**

$$
\begin{aligned}
N(i,j) &= x^{g_1(i,j,S)} + \sum_{k=i}^{j-1} N(i,k)N^1(k+1,j)x^{g_2(i,j,k,S)} && \text{(C.3)}\\
N^1(i,j) &= \sum_{k=i+1}^{j} N^b(i,k)x^{g_3(i,j,k,S)} && \text{(C.4)}\\
N^b(i,j) &= x^{g_4(i,j,S)} + \sum_{k=i+1}^{j-2}\sum_{l=k+1}^{j-1} N^b(k,l)x^{g_5(i,j,k,l,S)} \\
&\quad + \sum_{k=i+2}^{j-1} N^m(i+1,k-1)N^1(k,j-1)x^{g_6(i,j,k,S)} && \text{(C.5)}\\
N^m(i,j) &= \sum_{k=i}^{j-1}\left(x^{g_7(i,j,k,S)} + N^m(i,k-1)x^{g_8(i,j,k,S)}\right)N^1(k,j) && \text{(C.6)}
\end{aligned}
$$

Finally, $N(1,n)$ represents a polynomial whose factor of x^i corresponds to the number of structures which is at i Hamming distance from a reference structure.

Nomenclature

n	:	length of RNA sequence
d_{max}	:	maximum Humming distance between the reference and all candidates (1D)
d_{imax}	:	maximum Humming distance between the i-th reference and all candidates (2D)
E_i	:	free energy for the structure i
k_b	:	Boltzmann constant
T	:	temperature constant
Z	:	partition function
$f_i(\cdot)$	:	functions corresponding to energy contributions
$s(i)$	:	arbitrary integer score for an emission of i
$s(i, j)$	:	arbitrary integer score for a transition from i to j
$s(i, j, k)$	:	arbitrary integer score for a transition from (i,j) to k
$t(j\|i)$	:	proportional to the probability of a transition from i to j
$t(k\|i, j)$	:	proportional to the probability of a transition from (i,j) to k
N	:	length of DP array
c_k	:	arbitrary constant
S	:	structure vector
$\cdot \oplus \cdot$	:	exclusive disjunction
$\lfloor \cdot \rfloor$	:	floor function
$\|\| \cdot \|\|_1$	:	1-norm
$g_i(\cdot)$	:	Hamming distance gain by a transition
C	:	cumulative structure vector for $O(1)$ calculation of $g_i(\cdot)$
$\mathbb{N}$	:	a set of natural numbers
$\mathcal{S}$	:	a set of all possible secondary structures
$d(S_1, S_2)$	:	Hamming distance between S_1 and S_2
$\Delta\delta_i(\cdot)$	:	hamming distance gain between two reference structures by a transition
E_{S_i,S_j}	:	vector for $O(1)$ calculation of $\Delta\delta_i(\cdot)$
$d_{5'-3'}$	:	$5' - 3'$ distance
c_{ext}	:	the number of covalent bonds in the exterior loop
h_{ext}	:	the number of hydrogen bridges in the exterior loop