Development of RNA informatics for RNA sequence and structure analysis

February 2022

Manato Akiyama

A Thesis for the Degree of Ph.D. in Science

Development of RNA informatics for RNA sequence and structure analysis

February 2022

Graduate School of Science and Technology Keio University

Manato Akiyama

Abstract

Non-coding RNAs (ncRNAs) that are not translated into proteins were formerly considered as junk regions. However, various functions have been revealed in recent years ranging from development and cell differentiation processes to cause of diseases. Elucidation of ncRNA structural information is an indispensable step for understanding the function of ncRNA through RNA informatics, which is information science for RNA molecules. However, existing methods to obtain structural information of RNAs are not accurate, and the development of better methods is an active field of study. In this dissertation, I set out to develop more accurate methods for two different use cases: RNA secondary structure prediction and RNA sequence embedding. The background necessary for the explanation of these methods is given in Chapter 1.

The first method in this dissertation focuses on the development of a highly accurate RNA secondary structure prediction algorithm. Since the functions of ncRNAs are believed to be closely related to the structures of ncRNAs, it is possible to infer their biological functions from their structures. A popular approach for predicting RNA secondary structure is the thermodynamic nearest-neighbor model that finds a thermodynamically most stable secondary structure with minimum free energy (MFE). An alternative approach based on machine learning has been developed that can employ a fine-grained model that includes much richer feature representations. Rich feature representation is achieved by modeling more detailed substructures for RNA secondary structure. Although the machine learning-based fine-grained model achieved extremely high performance in prediction accuracy, the possibility of the risk of overfitting has been reported. In Chapter 2 of this dissertation, I propose a novel algorithm for RNA secondary structure prediction that integrates both the thermodynamic approach and the machine learning-based weighted approach. My benchmark showed that my algorithm achieved the best prediction accuracy compared with existing methods and resolved heavy overfitting.

"Embedding" is a popular technique that vectorizes DNA sequences and amino acid sequences, and is known to be useful for detecting DNA sequence motifs and predicting protein functions but embedding for RNA sequences has not been developed so far. In Chapter 3 of the dissertation, I showcase the development of a pre-training algorithm with the aim of acquiring an embedded vector of an RNA sequence that contains abundant structural information and sequence context information. Finally, to verify the quality of embedding, I performed two basic RNA informatics tasks (structural alignment and gene clustering), and in the process, achieved greater accuracy than existing state-of-the-art methods.

To conclude, I have succeeded in obtaining effective analytical methods of ncRNA using two approaches: RNA secondary structure prediction and RNA sequence vectorization. Each approach can be applied to analysis in all fields of RNA informatics including RNA-protein interaction and RNA-RNA interaction and can be expected to have a large spillover effect. In Chapter 4, the conclusions of this dissertation and the ripple effects are described in detail.

Content

Abbreviation

A: Adenine

ARI: Adjusted Rand Index

BERT: Bidirectional Encoder Representations from Transformers

BL-FR: Boltzmann Likelihood algorithm with Feature Relationships between parameters

C: Cytosine

CBOW: Continuous Bag-Of-Words

CFG: Context-Free Grammar

CLLM: Conditional Log-Linear Model

CNN: Convolutional Neural Network

DNA: Deoxyribonucleic acid

FN: False Negative

FP: False Positive

G: Guanine

GCE: Generalized Centroid Estimator

GO: Gene Ontology

MEA: Maximal Expected Accuracy

MFE: Minimum Free Energy

miRNA: microRNA

ML: Machine Learning

MLM: Masked Language Modelling

ncRNA: Non-Coding RNA

NLP: Natural Language Processing

NSP: Next Sentence Prediction

PPV: Positive Predictive Value

RNA: Ribonucleic acid

RNN: Recurrent Neural Network

rRNA: ribosomal RNA

SAL: Structural Alignment Learning

SCFG: Stochastic Context-Free Grammar

SEN: Sensitivity

SGD: Stochastic Gradient Descent

snoRNA: Small nucleolar RNA

snRNA: Small nuclear RNA

SSVM: Structured Support Vector Machine

t-SNE: T-distributed Stochastic Neighbor Embedding

T: Thymine

TF: True False

TM: Thermodynamic Model

TP: True Positive

tRNA: transfer RNA

U: Uracil

UTR: UnTranslated Region

Chapter 1 Introduction

Non-coding RNA (ncRNA) is RNA that is not translated into protein. The major non-coding RNAs include ribosomal RNA, transfer RNA, microRNA, snoRNA, snRNA, and piRNA. In recent years, these non-coding RNAs were revealed to have functions in cells, such as translation control and methylation of genes (Flanagan and Wild, 2007). Transfer RNA and ribosomal RNA are involved in translation events, transporting amino acids and synthesizing proteins. Both snoRNA and snRNA function in the nucleus, and are involved in regulating chemical modifications such as methylation of ribosomal RNA and mRNA splicing. Micro-RNA is 18 to 24 bases long and is causally implicated in various diseases by suppressing gene expression.

Next-generation sequencers have made it possible to acquire large amounts of RNA-related data. However, it is difficult to extract biological meanings from a large number of sequences, so information analysis is essential. RNA informatics is a field that aims to reveal RNA-related biological phenomena using algorithms from information science and statistics. RNA informatics includes RNA structure prediction (Do *et al.*, 2006; Lorenz *et al.*, 2011), RNA alignment (Sato *et al.*, 2012; Will *et al.*, 2007), RNA family classification (Morita *et al.*, 2009; Sato *et al.*, 2008), and prediction of interactions with other molecules (Kato *et al.*, 2010; Pan *et al.*, 2018). RNA structure is a key element in many of the topics in RNA informatics because the function of RNA molecules is related to their structure (Hirose and Tomari, 2016). In recent years, the development of RNA informatics and RNA interaction analysis has elucidated how RNA structures work in the formation of complexes with proteins and other molecules (Moore and ’t Hoen, 2019). RNA expresses its function by forming a three-dimensional structure analogous to a protein. RNA forms secondary structure consisting of base pairing and tertiary structure resulting from steric interaction. While DNA exists as a perfectly paired double helix, most RNA is single-stranded and therefore forms base pairs by hydrogen bonds within the RNA molecule. Since ribose has one more hydroxyl group than deoxyribose, RNA has a higher ability to form hydrogen bonds than DNA. For this reason, RNA forms complex base pair interactions in its secondary structure. The structural units that form these higher-order structures are called RNA structural motifs (Hendrix *et al.*, 2005). Unlike RNA sequence motifs found in primary sequences, RNA

structural motifs are defined by higher-order structures. For this reason, primary sequences that appear to be irrelevant may have the same RNA structural motif. RNA structural motifs often play an essential role as functional RNA. For example, one of the RNA tertiary structural motifs, kissing stem-loop, is formed by the intramolecular interaction of two loops and is involved in recombination (Balakrishnan *et al.*, 2001). Since these RNA structural motifs control various biological phenomena, the identification of RNA structures lead to the inference of their functions.

Hundreds of thousands of ncRNAs have been discovered with the help of high-throughput RNA sequencing. However, due to the vast amount of ncRNA data and limitations in experimental designs, finding the function of ncRNA remains a difficult task. Extraction of ncRNA structural information is an indispensable step for understanding the functions of ncRNAs in RNA informatics. In this dissertation, I explore effective analytical methods for ncRNA by two different approaches: RNA secondary structure prediction and RNA sequence embedding.

This dissertation is organized in the following manner. This chapter describes the basics of RNA secondary structure prediction and sequence embedding vectorization techniques, as well as the basic tasks of RNA informatics. In Chapter 2, I describe the results of the novel RNA secondary structure prediction algorithm. In Chapter 3, I describe the results of RNA structural alignment and clustering with informative RNA base embedding using deep representation learning. Finally, Chapter 4 presents the conclusions and future perspectives of this study.

1.1 Non-coding RNA

Untranslated regions are genomic regions that are transcribed into RNA but are not translated into proteins. In the human genome, 98% of the transcribed regions are untranslated regions (Nakamura, 2003). The percentage of untranslated regions in the entire genome for each species is 1 % for E. coli, 10 % for yeast, 30 % for C. elegans, and 43 % for human (Nakamura and Siomi, 2004). These facts suggest that there could be some correlation between the complexity of an organism and the size of its untranslated region. NcRNA genes, which are abundant in untranslated regions, are thought to explain this relationship. NcRNAs are known to have various functions such as catalyzing RNA processing and repressing mRNA translation. However, the loci and mechanisms of ncRNAs are still largely unknown, and have been the subject of extensive research in recent years.

1.1.1 RNA secondary structure

NcRNAs exert their functions by forming various RNA secondary structures. RNA secondary structures are folded structures formed by hydrogen bonds between bases based on Watson-Crick complementarity, such as "A" and "U", "G" and "C", and "G" and "U" in the molecule. The formation of these base pairs gives the RNA molecule its stability. RNA molecules can be divided into characteristic substructures, like protein domains. The double-stranded structure formed by the stacking of adjacent base pairs is called the stem. The single-stranded region sandwiched between the base pairs is called a loop. The loop at the end of the stem is called a hairpin loop. The unpaired nucleotides of the bulge loop appear on one side of the base pair. The internal loop occurs in the middle of the elongation of the double-stranded RNA. A loop that branches into three or more stems is called a multi-branched loop (Durbin *et al.*, 1998). In most RNA secondary structures, base pairs form a nested structure, while secondary structures that contain base pairs that are not nested are called pseudoknots.

1.1.2 RNA family

RNAs are classified into RNA families according to their functions. In addition to protein-coding RNA, there are various ncRNA families (Fig. 1). In this section, I introduce the representative ncRNA families.

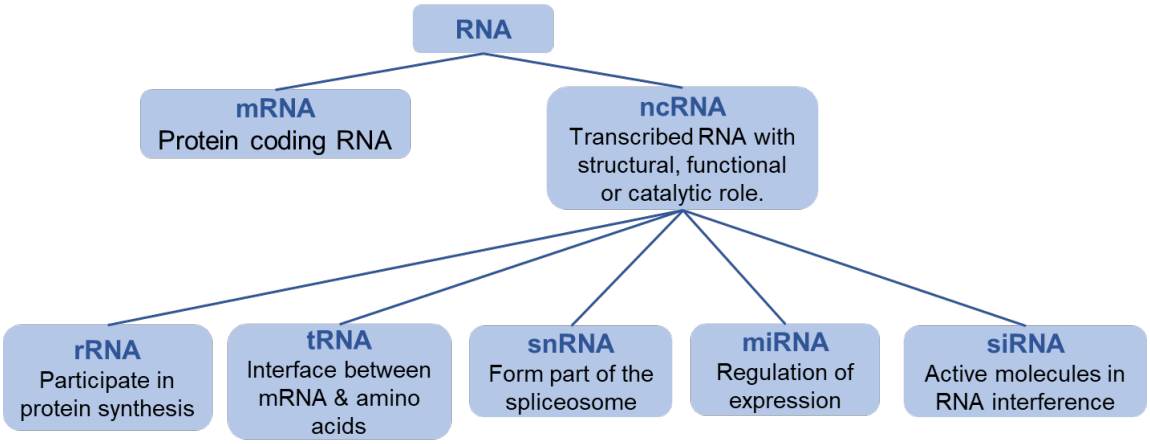


Fig. 1. RNA families

transfer RNA

Transfer RNA (tRNA) transports amino acids corresponding to codons of mRNA to the elongating polypeptide chain in translation. This function can be interpreted as the conversion of nucleic acid sequence information (codon) into amino acid residue information (protein). The tRNA has a cloverleaf-shaped higher-order structure with a region called an anticodon at one of its four edges. Anticodons bind to mRNA codons by nucleotide complementarity. The mapping between tRNA and amino acids is performed by an enzyme called aminoacyl-tRNA synthetase. Aminoacyl-tRNA synthetases are prepared for each type of amino acid, and bind tRNA and amino acid in the correct combination (Lodish *et al.*, 2019).

ribosomal RNA

Ribosomal RNA (rRNA) constitutes the ribosome, a large molecule that plays a central role in translation. The ribosome is composed of a large subunit and a small subunit, each of which is a complex consisting of rRNA and protein. When the two units bind to mRNA, the polymerization of amino acids proceeds and proteins are synthesized. There are three binding sites: A, P, and E. The A-site is the site where aminoacyl-tRNAs are most abundant; the P-site is the site where tRNAs that carry the elongating peptide chain are located; and the E-site

is the site where tRNAs that are ejected from the ribosome are located.

small nuclear RNA

Small nuclear RNAs (snRNAs) are small RNAs that are localized in the nucleoplasm of eukaryotes. The primary function of snRNAs is the processing of mRNA precursors (pre-mRNA) in the nucleus. snRNAs always form complexes with a set of corresponding proteins (snRNPs). These complexes include U1 snRNP, U2 snRNP, U4 snRNP, U5 snRNP, and U6 snRNP. These complexes are called spliceosomes, and they jointly catalyze splicing of mRNAs (Lodish *et al.*, 2019).

Small nucleolar RNA

Small nucleolar RNA (snoRNA) is a small RNA localized in the eukaryotic nucleolus. snoRNAs are involved in the maturation of rRNAs and small-nuclear RNAs (snRNAs) by catalyzing their methylation and pseudouridination. These chemical modifications are thought to enhance the function of RNA. Unlike snRNAs, of which there are only a few types in eukaryotes, snoRNAs are thought to exist in about 150,200 types. The complex formed by snoRNAs and proteins (snoRNPs) catalyzes the modification of RNA molecules. The snoRNAs bind complementarily to the sequence of the target RNA molecule. This binding leads the snoRNP to the target site.

microRNA

MicroRNAs (miRNAs) are small RNAs that do not form base pairs and are involved in the regulation of gene expression in eukaryotes. It is thought that more than 1000 miRNAs are encoded in the human genome. MicroRNAs bind to mRNAs that have complementary sequences to their own and degrade them, thereby suppressing the expression of specific genes. It has been reported that the expression levels of microRNAs are abnormal in various human cancers. Therefore, it is thought that microRNAs may be deeply involved in the development of cancer (Bushati and Cohen, 2007; Chang and Mendell, 2007).

1.2 Structure and Function of RNA

In this section, I will introduce the intracellular events related to RNA structure. This section was written based on a technical book (Hirose and Tomari, 2016).

Riboswitches are functional RNAs found in the 5′-untranslated region (UTR) of mRNAs that exert transcriptional control through direct binding to small molecule ligands. The typical riboswitch consists of an "aptamer motif" that binds to a specific ligand and an "expression control motif". In mRNAs with a riboswitch in the 5'UTR, the target ligand binds to the aptamer motif. This binding induces structural changes in the aptamer motif and the expression control motif, and regulates the expression of the genes. As with proteins, these RNA receptors facilitate the proper reaction by identifying chemically related metabolites with high selectivity. For example, in the S-adenosylmethionine-type riboswitch, the binding of SAM to the aptamer motif induces structural changes in RNA, followed by the formation of a transcription terminator that induces transcriptional repression (Fig. 2). SAM riboswitches are located upstream of genes encoding proteins involved in the biosynthesis of methionine and cysteine in Gram-positive bacteria, and repress the expression of these genes by forming transcription terminators. While SAM riboswitch is a riboswitch that suppresses gene expression, some have been found to promote gene expression (Serganov and Nudler, 2013). In order to bind specifically to the various molecules, riboswitches have different structures depending on the molecules to which they bind. Recently, research has been conducted to target riboswitches as a target for antibiotics, because riboswitches do not exist in humans and thus there is little concern about side effects. In fact, recent studies have revealed that some antibiotics, whose mechanisms have long been unknown, target the riboswitch (Howe *et al.*, 2015).

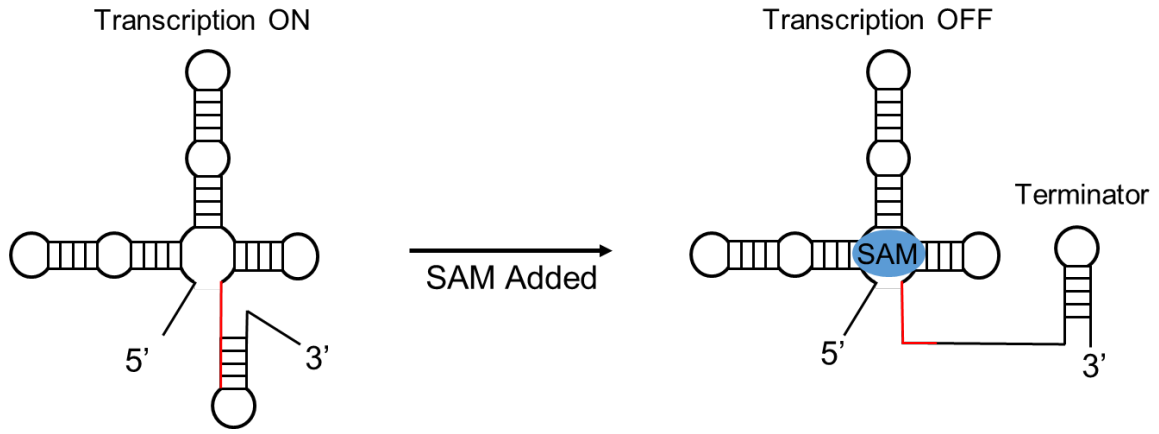


Fig. 2. Gene regulation by SAM riboswitch.

RNA rarely expresses its function by itself. RNA often controls intracellular events by interacting with other RNAs and proteins. RNA structure is also important in RNA-protein interactions. LIN28 is an RNA-binding protein with CSD (cold shock domain) and ZKD (Zn knuckle domain), and is involved in the maintenance of pluripotency of ES cells (embryonic stem cells) (Mayr and Heinemann, 2013). Lin28 binds to specific sequences present in the loop structure of pre-miRNAs (precursor miRNAs) belonging to the let-7 family, which consists of 12 types of miRNAs (Fig. 3). Let-7 is one of the first miRNAs to be discovered and is a very important gene in tumor suppressor function and development. This binding inhibits miRNA production by preventing Dicer from cleaving pre-miRNA. At this time, the recognition of LIN28 relies on cooperative targeting by both ZKD, which binds to the GGAG sequence motif in the internal loop structure, and CSD, which binds to the GNGAY sequence motif in the hairpin loop structure (N for one of the AUCG base and Y for one of the CU bases). The presence of these sequences on a particular structure is required for the interaction of Lin28 and pre-miRNA. In fact, altering the structure of the RNA loop region weakens binding to Lin28. Thus, not only the primary sequence but also the structural background is an important factor for RNA-protein interaction.

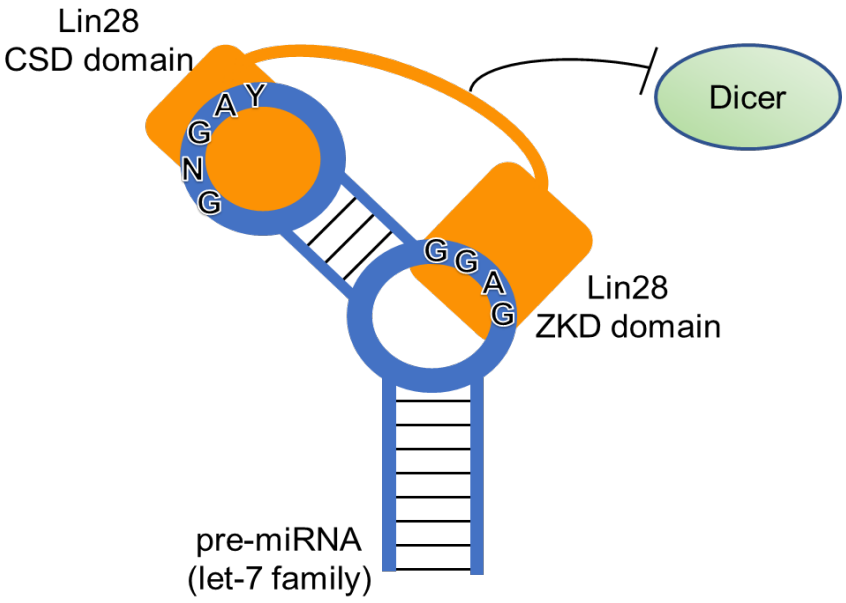


Fig. 3. The binding of lin28 prevents the maturation (production) of the let-7 gene by Dicer. This binding occurs in undifferentiated cells because let-7 is a gene that represses developmental processes.

1.3 RNA secondary structure prediction

RNA secondary structure prediction is defined as determining the secondary structure of x when any input of RNA base sequence $x = x_1x_2...x_n$ is given. The RNA secondary structure prediction algorithm can be decomposed into four elements, “architecture”, “scoring scheme”, “parameterization”, and “folding algorithms” (Rivas, 2013). The architecture determines the units of the substructure parameters that form the secondary structure. The architecture depends on how detailed the modeling of the parameters is. A scoring scheme is a way of scoring a secondary structure using parameters. The parameterization depends on how the parameter values are determined. Different parameterizations are possible even for the same scoring scheme. There are several folding algorithms that predict the secondary structure using the parameters obtained from the above three factors. Below, I explain the four components in detail according to the paper by Rivas (Rivas, 2013).

Architecture

The RNA secondary structure can be represented using context-free grammar (CFG). CFG is a system for generating character strings, consisting of the rules and symbols shown below.:

- terminal symbol: collection of characters that cannot be rewritten. The final sequence generated by the rule consists only of terminal symbols.
- nonterminal symbol: the grammar symbols for rule applications. Nonterminal symbols are taken part in the generation of sentences but are not components of the sentence.
- productions: rules for replacing non-terminal symbols with a combination of non-terminal symbols and terminal symbols.
- start symbol: special non-terminal symbol that begins a string generation by the grammar.

For example, the production rule for base pair formation can be represented as ($A \rightarrow xA\hat{x}$). Here, A is the non-terminating symbol and x and $\hat{x}$ are the terminating symbols for the two paired bases. The production rule is expressed as "non-terminal symbol → symbol sequence of non-terminal and terminal symbols". For example, when the production rule ($A \rightarrow xA\hat{x}$) is

applied to the symbol string (xA), the symbol string ($xA \Rightarrow xxA\hat{x}$) is generated. In this case, one unpair base and one base pair is generated on the right side. One of the most famous grammars, the Nussinov grammar (Nussinov and Jacobson, 1980) for generating RNA sequences, is defined as follows.

$$A \rightarrow Ax|AxA\hat{x}|\text{end}$$

In addition to Nussinov grammar, the g6 grammar introduced three non-terminal as follows.

1. $A \rightarrow BC|B|\text{end}$
2. $B \rightarrow xC\hat{x}|x$
3. $C \rightarrow xC\hat{x}|BA$

Modeling of secondary structures with g6 grammar achieves better prediction accuracy than nussinov grammar. More detailed modeling of these basic grammars is currently used in thermodynamic or probabilistic or weight schemes. The representative grammar adopted by various methods is shown below.

- Base helix ($A^{y\hat{y}} \rightarrow xA^{x\hat{x}}\hat{x}$)

Contiguous structure of base-pairs. A base-pair $x\hat{x}$ is developed next to the base-pair $y\hat{y}$.

- Dangles ($B^{y\hat{y}} \rightarrow xC|Cx$)

Represents a single base adjacent to a base-pair $y\hat{y}$. $B^{y\hat{y}}$ is a non-terminal symbol representing a base pair. That is, the terminal symbol x generated by this rule is adjacent to the base pair $y\hat{y}$. C can be converted to any non-terminal symbol.

- Hairpin loops ($B \rightarrow x_1, x_2, x_3, \ldots, x_n$)

A series of single stranded bases of length n closed by a base-pair. In contrast to $B^{y\hat{y}}$, which represents a specific base pair, B is a non-terminal symbol that represents all base pairs.

- Internal loops ($B \rightarrow (x_1, x_2, x_3, \ldots, x_n)C(x_1, x_2, x_3, \ldots, x_m)$)

Internal loops occur in middle of a stretch of double stranded RNA. As shown in Fig. 4, two single strands are sandwiched between base pairs. The length of each single strand is n and m.

- Bulge loops ($B \rightarrow (x_1, x_2, x_3, \ldots, x_n)C \mid C(x_1, x_2, x_3, \ldots, x_n)$)

The unpaired bases appear on one side of the base-pair. Depending on where the loop exists, there are two types of bulge loops: 5' bulge loops and 3' bulge loops.

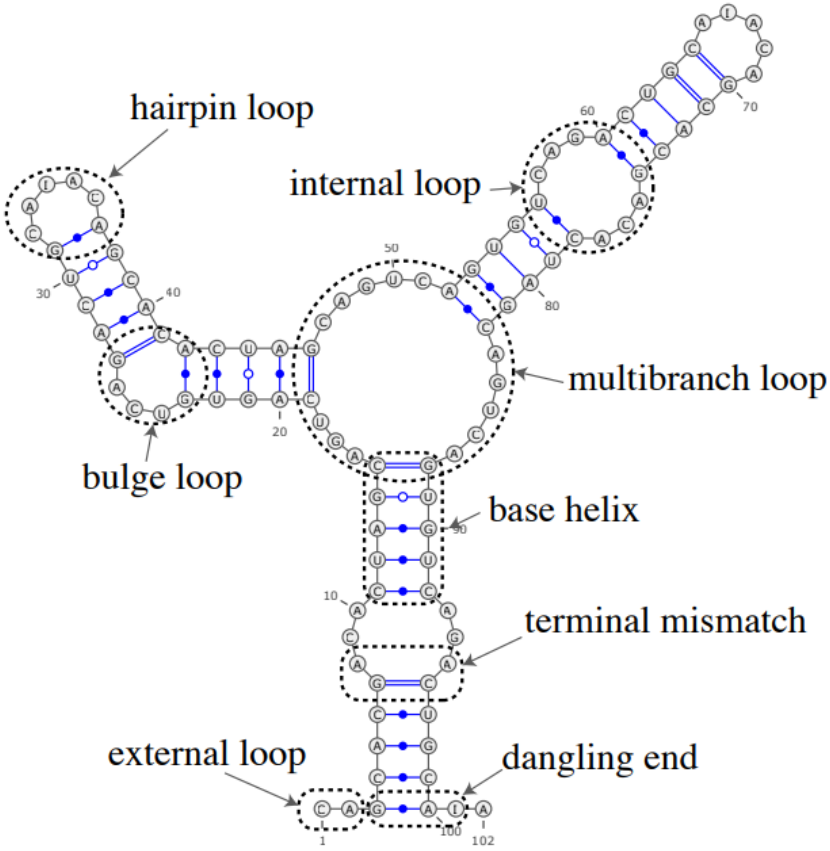


Fig. 4. Examples of substructures defined in the standard nearest neighbor model

Scoring Scheme, Parameterization

There are three methods for scoring the RNA secondary structure based on the parameters defined in the previous section. Each scoring method is closely related to the method of determining the values of the parameters.

- Models based on Thermodynamic scheme

In the thermodynamic approach, Gibbs free energy is assigned to the parameter as a score. Measurable parameters including loops and stacking are obtained by finding the equilibrium constant before and after melting (Schroeder and Turner, 2009). At higher temperatures, the hydrogen bonds of double-stranded RNA are broken and it becomes single-stranded. The melting temperature, at which the ratio of double-stranded to single-stranded RNA is equal, is an indicator of the thermal stability of nucleic acids. By measuring the melting temperature at various concentrations, a van't Hoff plot can be drawn. In the van't Hoff plot, the y-axis is the reciprocal of the melting temperature and the x-axis is the logarithm of the concentration. The enthalpy change can be obtained from the slope of the van't Hoff plot. The entropy change can be obtained from the y-intercept. The free energy and the equilibrium constant can be calculated using the enthalpy change and the entropy change. The free energy is calculated for each substructure. The energy of the whole RNA molecule is the sum of the free energies of the substructures.

- Models based on Probabilistic parameters

In the stochastic scheme, the probability that each base forms a base-pair is estimated. In many ways, these probabilities are empirically calculated from the known secondary structure. In the stochastic schemes, stochastic CFGs (SCFGs) is applied to model RNA secondary structure. SCFG (Stochastic context-free grammar) is a context-free grammar whose probability is assigned to each production rule. The probability of parse is expressed as the product of probabilities of production rules used in the derivation. The parameters for generating the desired secondary structure can be calculated by maximizing the joint probability of the sequence and its secondary structure.

- Models based on weights

In the weight scheme, instead of the probability, the weight taken by each base-pair is

calculated. In the simplest weight scheme, the secondary structure is estimated where the number of base-pairs increases as much as possible by setting the canonical base-pair weight to ”1” and the rest to ”0”. In many methods, weights are obtained by learning from large amounts of data. In the weighted scheme, unlike the stochastic scheme, the parameters are real values. Therefore, the conditional probability of the secondary structure given the sequence is calculated instead of the joint probability. Algorithms that use machine learning to determine weights search for parameter values that maximize this conditional probability.

Although models based on probabilistic parameters or weights achieved high performance in prediction accuracy, a possibility of the risk of overfitting for such models has been reported. In Chapter 2, I propose an algorithm for RNA secondary structure prediction that integrates the thermodynamic approach and the machine learning-based weighted approach. If RNA secondary structure prediction can be performed with high accuracy, RNA structural motifs can be detected more easily. As shown in Chapter 1.2, not only the primary sequence but also the secondary structure is important for RNA function, so the development of an accurate secondary structure prediction algorithm is useful for function estimation.

Folding Algorithms

Between scoring schemes, the folding algorithm used for secondary structure determination is essentially identical. Dynamic programming (DP) algorithms are frequently used as the folding algorithms. Here, I introduce two policies for RNAfolding and a dynamic programming method for their derivation.

- Minimum free-energy (MFE) structure

The optimal secondary structure is returned by performing score calculation using DP algorithms. In thermodynamic method, DP algorithms calculate the MFE structure. In probabilistic method, DP algorithms calculate the structure that takes the highest probability, known as the CKY algorithm.

- Maximal expected accuracy (MEA) structure

The MEA structure is an RNA secondary structure in which the sum of the posterior probabilities of base pairs is maximized. To calculate the posterior probability of a base pair, it is necessary to obtain a partition function. The sum of possible secondary structure scores

for an RNA sequence is defined as the partition function. The partition function is the sum of the Boltzmann factors of all structures (See Chapter 2.2.4). Thermodynamic Partition function is calculated efficiently using the McCaskill algorithm which is a DP algorithm. The McCaskill algorithm derives the distribution of the energies of all possible structures, not the energies of individual structures. For details of the McCaskill Algorithm, please refer to the original paper (McCaskill, 1990). In probabilistic method, the partition function is the sum of probabilities of all the structures. Probabilistic partition function is calculated using the inside-outside algorithm which is very similar to McCaskill algorithm.

Base-pairing probability (Posterior probabilities) can be calculated by using the inside-outside algorithm and the partition function. By obtaining the posterior distribution, it is possible not only to predict the secondary structure but also to sample the structures.

After calculating the posterior probabilities for each base-pair, MEA structure is calculated by dynamic programming. In exchange for the computation time, the MEA structure provides more accurate secondary structure prediction than the MFE structure. In addition, the trade-off between sensitivity and PPV can be controlled by introducing a parameter for the degree of base pair formation.

1.4 Word embeddings for natural language processing and bioinformatics

Because bioinformatics and natural language processing (NLP) have much in common, there are techniques and algorithms that can be transferred between the two fields. Research is being actively conducted to obtain effective representation of DNA sequences, RNA sequences, and amino acid sequences by using deep learning, especially utilizing techniques developed in the field of natural language processing. These studies are based on the idea that nucleotide composition and sequence structure determine the motif and function of a gene sequence, just as the complex grammatical structure of natural language determines the meaning of a sentence.

Word embedding is a technique developed in the field of natural language processing that embeds words in a low-dimensional continuous vector space. Different from one-hot expression, which is the simplest word vectorization method, word embedding can obtain a distributed representation. The one-hot expression first obtains a list of words to be used and prepares a vector with a predetermined index corresponding to each word. Each word expression is obtained by setting the element of vector index corresponding to a specific word to 1 and setting the other dimensions to 0.

There are studies in which ncRNA features are extracted using deep learning techniques after applying one hot encoding to RNA sequences (Baek *et al.*, 2018; Aoki and Sakakibara, 2018). However, one-hot expressions have drawbacks such as being vulnerable to the curse of dimensionality and not being able to reflect the interrelationships between words. To make matters worse, the distance between any pair of one-hot vectors is equidistant.

How are word-to-word similarities in word embedding defined? The most common way to measure similarity is to use word co-occurrence. This presupposes the hypothesis that words with similar meanings will appear in similar contexts. Methods developed based on this hypothesis include count-based and predictive methods. The count-based method calculates the frequency of words that appear in various contexts and creates a co-occurrence matrix. The predictive method is a method of learning a word vector by predicting a target word from the words before and after in the context. In these methods, words are mapped into a space of latent variables compressed to any predefined number of dimensions. One-hot

expressions are expressions in which one dimension corresponds to one word, while word embedding shares multiple concepts in multiple dimensions.

Word2vec is one of the most famous predictive word embedding techniques (Mikolov *et al.*, 2013). Word2vec is a method of vectorizing the meaning of a word using a large amount of text data. Since the distributed representations of words are related to each other, it is possible to perform operations between different word vectors. Word2vec uses one of two different tasks, CBOW (Continuous Bag of Words) and Skip-Gram, to get word embedding. In CBOW, words are predicted from surrounding words in a sentence. Skip-Gram, on the other hand, predicts the words around a word. These are based on the hypothesis that words with similar meanings have similar peripheral words. Skip-Gram learns the weights of a two-layer neural network that outputs peripheral words of an input word. dna2vec (Ng, 2017) is a method in which Word2vec is applied to a DNA sequence. In dna2vec, the learning framework of Word2vec was applied to obtain the distributed representation of k-mer.

Context-independent techniques such as Word2Vec have problems in vectorizing polysemous words. For example, "book" has two meanings, "pieces of paper" and "reserve", but Word2Vec gives them the same distributed representation. This indicates that when Word2vec is applied to a DNA sequence, each k-mer has the same distributed representation in any context.

ELMo and BERT generate context-sensitive word-distributed representations (Peters *et al.*, 2018; Devlin *et al.*, 2019). In these methods, the same word is assigned different distributed representations that depend on surrounding words. BERT is a pre-learning algorithm for obtaining word embedding and sentence embedding by performing multiple tasks. The BERT learning algorithm consists of two tasks: a mask language modeling (MLM) task and a next sentence prediction (NSP) task. The MLM task predicts multiple masked tokens (words) in a sentence. The NSP task determines if two statements are consecutive. PLUS (Min *et al.*, 2021) is a method of obtaining embedding of each amino acid by applying a task inspired by BERT to proteins. PLUS uses this embedding to achieve highly accurate homology prediction. Table 1 shows a comparison of each embedding method. One-hot encoding requires very large feature dimensions, but Word2vec, BERT, and ELMo can achieve small feature dimensions. In addition, BERT and ELMo allow for context-dependent embedding that is not possible with Word2vec. Obtaining better embedding enhances the quality of downstream analysis. In Chapter 3, I propose RNABERT for effective embedding of RNA

bases by applying BERT training to non-coding RNA. In that section, I aim to obtain an embedding vector of RNA bases that incorporates RNA secondary structure and primary sequence. As shown in Chapter 1.2, the functions of ncRNAs are correlated with RNA secondary structures and primary sequences. Therefore, obtaining such embedding vectors is useful for estimating the functions of ncRNAs. Specifically, by using the embedding vector of bases as an input to the function for estimating the function of ncRNAs, it becomes possible to estimate the function based on the secondary structure and primary sequence.

Table 1 A set of models used to generate word embeddings

Methods	Memory saving	Meaning-sensitive	Context-sensitive	Dimensions
One-hot Encoding	-	-	-	~1,000,000
Word2Vec (Mikolov *et al.*, 2013) Glove (Pennington *et al.*, 2014)	+	+	-	~100
ELMo (Peters *et al.*, 2018) BERT (Devlin *et al.*, 2019)	+	+	+	~1,000

Meaning-sensitive: It is possible to quantify the similarity of word meanings.

Context-sensitive: Embedding is variable depending on the context.

Dimensions: Dimensional order of an embedded vector

1.5 RNA structural alignment

The structural alignment of RNA sequences calculates the alignment of not only RNA sequences but also their secondary structures. Structural alignment of RNA sequences seeks to establish homology between two or more structures based on the RNA secondary structure. Structural alignment of RNA sequences allows us to identify functionally important regions and track the evolutionary history of related molecules. The most influential method for the

structural alignment of RNA sequences is the Sankoff algorithm, which simultaneously performs secondary structure prediction and alignment (Sankoff, 1985). However, the time complexity of the naive implementation of the Sankoff algorithm is $O(n^6)$ for a length n of input RNA sequences, and accelerating the Sankoff algorithm is an unsolved hard problem (Lalwani *et al.*, 2014). While Sankoff-style algorithms such as LocARNA (Will *et al.*, 2007) and Dynalign (Fu *et al.*, 2014) calculate the alignment considering the secondary structure, a standard sequence-based (non-structural) alignment method such as the Needleman-Wunsch algorithm (Needleman and Wunsch, 1970) determines only the correspondence between each base position of two input sequences, and its time complexity is only $O(n^2)$ using the dynamic programming technique. Hence, I aim to apply the informative base embedding to determine the position-dependent and secondary structure-dependent score matrix in calculating alignments so that the structural alignment is obtained using a simple Needleman-Wunsch algorithm instead of the computationally expensive Sankoff-style algorithm. In Chapter 3, I perform a pairwise alignment test using RNA base embedding with RNABERT.

1.6 RNA family clustering

Building an appropriate clustering algorithm for ncRNAs is an effective step towards unsupervised analysis of ncRNA sequences without their family labels (Heyne *et al.*, 2012; Saito *et al.*, 2011), as high-throughput sequencing continues to generate a large number of RNA sequences, including novel transcripts. With the recent increase in deep learning usage, many algorithms for ncRNA classification (supervised clustering) using convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have been proposed (Baek *et al.*, 2018; Aoki and Sakakibara, 2018). These algorithms adopt a simple embedding technique, one-hot encoding of RNA bases. Most of these algorithms utilize supervised learning using ncRNA families as labels for training. Nevertheless, since supervised learning requires the data to be labelled, this approach is not practical when analyzing ncRNA sequences without their family labels. In Chapter 3, RNA family classification is performed as a second test to confirm the quality of embedding.

Chapter 2 A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model

In this chapter, I show the results of A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model. A popular approach for predicting RNA secondary structure is the thermodynamic nearest neighbor model that finds a thermodynamically most stable secondary structure with the minimum free energy (MFE). For further improvement, an alternative approach that is based on machine learning techniques has been developed. The machine learning based approach can employ a fine-grained model that includes much richer feature representations with the ability to fit the training data. Although a machine learning based fine-grained model achieved extremely high performance in prediction accuracy, a possibility of the risk of overfitting for such model has been reported. Results: In this dissertation, I propose a novel algorithm for RNA secondary structure prediction that integrates the thermodynamic approach and the machine learning based weighted approach. My fine-grained model combines the experimentally determined thermodynamic parameters with a large number of scoring parameters for detailed contexts of features that were trained by the structured support vector machine (SSVM) with the $l1$ regularization to avoid overfitting. My benchmark showed that my algorithm achieves the best prediction accuracy compared with existing methods, and heavy overfitting cannot be observed. Availability: The implementation of my algorithm is available at https://github.com/ keio-bioinformatics/mxfold.

2.1 Background

NcRNAs that are not translated into proteins were formerly considered as junk regions. However, these various functions have been revealed in recent years ranging from the process of development and cell differentiation to the cause of disease. Since the functions of ncRNAs are believed to be closely related to the structures of ncRNAs, it is possible to infer their biological functions from their structures. RNA tertiary structures can be determined by experimental assays including X-ray crystal structure analysis and nuclear magnetic

resonance (NMR). However, there are severe difficulties of these experimental assays such as high experimental cost and low throughput. In addition, the computational techniques to predict RNA tertiary structures have still been immature. Therefore, the computational prediction of RNA secondary structures, which can be easily modeled by a set of hydrogen bonds between nucleotides, has frequently been used instead.

As shown in Chapter 1, RNA secondary structure prediction methods are roughly classified into three approaches from the viewpoint of the scoring scheme: a thermodynamic approach, a probabilistic approach, and a weighted approach (Rivas, 2013). The thermodynamic approach has been the most popular approach that finds a thermodynamically most stable secondary structure with the minimum free energy (MFE) and has been utilized by a number of tools including UNAfold (Zuker, 1989), RNAfold (Lorenz *et al.*, 2011), and RNAstructure (Reuter and Mathews, 2010). RNA secondary structures can be decomposed into characteristic substructures such as hairpin loops and base-pair stacking according to the nearest neighbor model (Zuker and Stiegler, 1981). Free energy of each substructure was determined by experimental methods such as the optical melting experiment (Schroeder and Turner, 2009). The free energy of the secondary structure is calculated by summing up the free energy of each substructure in the secondary structure. The dynamic programming technique enables us to efficiently find the MFE structure from all possible secondary structures for a given RNA sequence.

The probabilistic approach has employed generative models including stochastic context-free grammars (SCFGs) for modeling RNA secondary structures. SCFGs are defined by a set of derivation rules, or grammar, whose probabilities are trained by the maximum likelihood (ML) estimation from the training data, and were applied to RNA secondary structure prediction (Sakakibara *et al.*, 1994; Eddy and Durbin, 1994; Knudsen and Hein, 1999; Dowell and Eddy, 2004). Sato et al. proposed a non-parametric Bayesian extension of SCFGs with the hierarchical Dirichlet process that can find an optimal RNA grammar from the training data (Sato *et al.*, 2010). Rivas et al. developed a framework called TORNADO for flexibly describing RNA grammars, and showed that a complex RNA grammar that simulates the nearest neighbor model can achieve as accurate predictions as the weighted models can (Rivas *et al.*, 2012).

The weighted approach has utilized machine learning techniques instead of the experimental techniques in order to determine weights for decomposed substructures, i.e.,

scoring parameters. CONTRAfold was developed based on the conditional log-linear models (CLLMs) that find scoring parameters that can most probably discriminate between correct structures and incorrect structures (Do *et al.*, 2006). Simfold implemented Boltzmann likelihood algorithm with feature relationships between parameters (BL-FR), which is similar to CLLMs, but incorporated free energy parameters (Andronescu *et al.*, 2010). ContextFold employed a fine-grained model that includes much richer contexts of features with the ability to fit the training data, combined with a machine learning algorithm (Zakov *et al.*, 2011). Although ContextFold achieved extremely high performance in prediction accuracy, Rivas et al. reported a possibility of the risk of overfitting for ContextFold (Rivas, 2013). From this observation, I can see that an important issue for further improving prediction accuracy is to effectively learn a large number of scoring parameters for a fine-grained model without overfitting.

In this dissertation, I propose a novel algorithm for RNA secondary structure prediction that integrates the thermodynamic approach and the machine learning based weighted approach. My fine-grained model combines the experimentally-determined thermodynamic parameters with a large number of scoring parameters for detailed contexts of features. In order to train the scoring parameters of the fine-grained model, I employed the structured support vector machine (SSVM) (Tsochantaridis *et al.*, 2005) with the L1 regularization to avoid overfitting. My benchmark showed that my algorithm achieves the best prediction accuracy compared with existing methods, and heavy overfitting as shown in ContextFold cannot be observed.

The major advantages of my work are summarized as follows: (i) The max-margin based training algorithm learns my fine-grained model that can perform accurate secondary structure prediction, and (ii) my scoring model that integrates the thermodynamic and machine learning based model enables accurate and robust structure prediction even for unobserved substructures in the training dataset.

2.2 Materials and Methods

2.2.1 Preliminaries

Let $\Sigma = \{A, C, G, U\}$ and Σ^* denote the set of all finite RNA sequences consisting of bases in Σ. For a sequence $x = x_1 x_2 \cdots x_n \in \Sigma^*$, let $|x|$ denote the number of symbols appearing in x, which is called the length of x. Let $S(x)$ be a set of all possible secondary structures of x. A secondary structure $y \in S(x)$ is represented as a $|x| \times |x|$ binary-valued triangular matrix $y = (y_{ij})_{i<j}$, where $y_{ij} = 1$ if and only if bases x_i and x_j form a base-pair composed by hydrogen bonds including the Watson-Crick base-pairs (A-U and G-C), the Wobble base-pairs (G-U).

2.2.2 Scoring model

A scoring model $f(x, y)$ is a function that assigns real-valued scores to an RNA secondary structure $y \in S(x)$ for an RNA sequence $x \in \Sigma^*$. My aim is to find a secondary structure $y \in S(x)$ that maximizes the scoring function $f(x, y)$ for a given RNA sequence $x \in \Sigma^*$.

RNA secondary structures can be decomposed into characteristic substructures, or features, such as hairpin loops and base-pair stacking. I denote by $\Phi(x, y)$ the feature representation vector of (x, y), which consists of the number of occurrence of every feature in (x, y). Each feature in Φ is associated with a corresponding score or weight. I assume a linear scoring model of RNA secondary structures as:

$$f(x, y) = \boldsymbol{\lambda}^{\top} \Phi(x, y), \tag{1}$$

where $\boldsymbol{\lambda}$ is a weight vector in which λ_i is the weight of the *i*-th feature in Φ.

Note that the thermodynamic approach can be represented by this linear scoring model if I define Φ as the nearest neighbor model and the corresponding weights as the negative of experimentally determined free energy parameters.

I propose a novel scoring model that integrates the thermodynamic approach and the machine learning based weighted approach. I define my scoring model as:

$$\begin{aligned} f(x, y) &= f_T(x, y) + f_W(x, y) \\ f_T(x, y) &= \boldsymbol{\lambda}_T^{\top} \Phi_T(x, y) \\ f_W(x, y) &= \boldsymbol{\lambda}_W^{\top} \Phi_W(x, y), \end{aligned} \tag{2}$$

where $f_T(x, y)$(resp. $f_W(x, y)$) is the contribution of the thermodynamic model (resp. the machine learning model) to my scoring model. For the thermodynamic model, I employed the nearest neighbor model as Φ_T and the negative of the Turner free energy parameters (Turner and Mathews, 2010) as $\boldsymbol{\lambda}_T$. For the machine learning model, I constructed a fine-grained model as Φ_W (see Chapter 2.2.3) and corresponding weights $\boldsymbol{\lambda}_W$ that are trainable from training data by using SSVM (see Chapter 2.2.5).

2.2.3 Feature representations

Both feature representations Φ_T and Φ_W are based on the nearest neighbor model (Zuker and Stiegler, 1981), including base helices, dangling ends, terminal mismatches, hairpin loops, bulge loops, internal loops, multibranch loops and external loops (Fig. 4). In order to calculate the free energy of RNA secondary structures more precisely, some specialized loop parameters have been adopted in frequently used free energy parameter sets for the standard nearest neighbor model. For example, the Turner 1999 and 2004 models contain several sequential features such as hairpin loops with 3, 4 or 6 nucleotides and internal loops with (1, 1) nucleotides (1 nucleotide at 5' loop and 1 nucleotide at 3' loop), (1, 2) nucleotides and (2, 2) nucleotides (Turner and Mathews, 2010). As the fine-grained feature representation Φ_W, I employed much longer sequential features for hairpin loops with m nucleotides, bulge loops with m nucleotides and internal loops with (m, n) nucleotides ($m \leq L \, and \, m + n \leq L$) in addition to the standard nearest neighbor model. I used L = 7 by default as described in Results. See Chapter 2.3.5 for more details.

2.2.4 Decoding algorithm

Viterbi decoding:

Since both Φ_T and Φ_W are based on the nearest neighbor model, any secondary structures can be decomposed into the same substructures for both representations. Therefore, the most probable secondary structure that maximizes Eq. (2) can be obtained by the Zuker-style dynamic programming algorithm (Zuker and Stiegler, 1981).

Posterior decoding:

The posterior probability of the secondary structure y given RNA sequence x, $p(y|x)$, under the scoring model $f(x, y)$ is calculated by:

$$p(y|x) = \frac{\exp[f(x,y)/RT]}{Z(x)}$$
$$Z(x) = \sum_{y \in S(x)} \exp[f(x,y)/RT], \quad \textbf{(3)}$$

where R is the gas constant and T is the absolute temperature. The basepairing probability p_{ij} is the probability that the i-th and j-th nucleotides form a base-pair, which is defined as follows:

$$p_{ij} = E_{y|x}[I(y_{ij} = 1)] = \sum_{y \in S(x)} I(y_{ij} = 1) p(y \mid x), \quad \textbf{(4)}$$

where I (condition) is an indicator function which takes a value of 1 or 0 depending on whether the condition is true or false. The McCaskill algorithm (McCaskill, 1990) can be utilized to efficiently calculate the base-pairing probabilities (4) by the dynamic programming techniques.

I define a gain function between a true structure y and a candidate structure $\hat{y}$ by

$$G(y, \hat{y}) = \sum_{1 \le i \le j \le |x|} \{\gamma I(y_{ij} = 1) I(\hat{y}_{ij} = 1) + I(y_{ij} = 0) I(\hat{y}_{ij} = 0)\}, \quad \textbf{(5)}$$

where $\gamma > 0$ is a weight for base-pairs. The gain function (5) is equal to the weighted sum of the number of true positives and the number of true negatives of base-pairs.

The expectation of the gain function (5) with respect to an ensemble of all possible secondary structures under a given posterior distribution $p(y|x)$ is

$$E_{y|x}\left[G(y,\hat{y})\right] = \sum_{y \in S(x)} G(y,\hat{y}) p(y \mid x) \tag{6}$$

$$= \sum_{1 \le i \le j \le |x|} \left((\gamma + 1)\, p_{ij} - 1\right) I(\hat{y}_{ij} = 1) + C\,,$$

where C is a constant independent of $\hat{y}$.

Then, $\hat{y}$ that maximizes the expected gain (6) can be obtained using the recursive equations:

$$M_{i,j} = \max \begin{cases} M_{i+1,j} \\ M_{i,j-1} \\ M_{i+1,j-1} + (\gamma + 1) p_{ij} - 1, \\ \max\limits_{i<k<j} M_{i,k} + M_{k+1,j} \end{cases} \tag{7}$$

and tracing back from $M_{1,|x|}$.

The trade-off between specificity and sensitivity can be contraolled by γ. I call the maximization of Eq. (6) the generalized centroid estimator (GCE) since this is equivalent to the centroid estimator (Ding *et al.*, 2005; Carvalho and Lawrence, 2008) for γ=1. The generalized centroid estimator is very similar to the maximum expected accuracy (MEA) estimator (Do *et al.*, 2006). See (Hamada *et al.*, 2009; Sato *et al.*, 2009) for more details.

2.2.5 Learning algorithm

To optimize the feature parameter $\boldsymbol{\lambda}_W$, I employed a max-margin framework called structured support vector machines (SSVM) (Tsochantaridis *et al.*, 2005). Given a training dataset $D = \{(x^{(k)}, y^{(k)})\}_{k=1}^{K}$, where $x^{(k)}$ is the k-th RNA sequence and $y^{(k)} \in S(x^{(k)})$ is the correct secondary structure for the k-the sequence $x^{(k)}$, I aim to find $\boldsymbol{\lambda}_W$ that minimizes the objective function

$$L(\boldsymbol{\lambda}_W) = \sum_{(x,y) \in D} \left(\max_{\hat{y} \in S(x)} [f(x,\hat{y}) + \Delta(y,\hat{y})] - f(x,y) + C \|\boldsymbol{\lambda}_W\|_1 \right), \tag{8}$$

where $\|.\|_1$ is the l_1 norm and C is a weight for the l_1 regularization term to avoid overfitting to training data (I used C = 0.001 by default). Here, $\Delta(y,\hat{y})$ is a loss function of $\hat{y}$ for y

defined as

$$\Delta(y, \hat{y}) = \delta^{\mathrm{FN}} \times (\#\text{ of false negative base pairs}) + \delta^{\mathrm{FP}} \times (\#\text{ of false positive base pairs}), \quad (9)$$

where δ^{FN} and δ^{FP} are tunable hyperparameters to control the trade-off between sensitivity and specificity for learning the parameters. I used $\delta^{\mathrm{FN}} = 8.0$ and $\delta^{\mathrm{FP}} = 1.0$ by default. In this case, the first term of Eq. (9) can be calculated using the Zuker-style dynamic programming algorithm modified by the loss-augmented inference (Tsochantaridis *et al.*, 2005). To minimize the objective function (8), stochastic subgradient descent (Fig. 5) or its variant can be applied.

1: $\boldsymbol{\lambda}_W \leftarrow \mathbf{0}$
2: **repeat**
3: **for all** $(x, y) \in \mathcal{D}$ **do**
4: $\hat{y} \leftarrow \arg\max_{\hat{y}} [f(x, \hat{y}) + \Delta(y, \hat{y})]$
5: **for all** $\lambda_{Wi} \in \boldsymbol{\lambda}_W$ **do**
6: $\lambda_{Wi} \leftarrow \lambda_{Wi} - \eta(\phi_{Wi}(x, \hat{y}) - \phi_{Wi}(x, y) + C\mathrm{sgn}\lambda_{Wi})$
7: **end for**
8: **end for**
9: **until** all the parameters converge

Fig. 5. The stochastic subgradient descent algorithm for SSVMs. sgn is the sign function. $\eta > 0$ is the predefined learning rate

2.3 Results

2.3.1 Implementation

My algorithm was implemented as a program called MXfold, which is short for the MaX-margin based rna FOLDing algorithm. The source code is available at https://github.com/keio-bioinformatics/ mxfold. The free energy parameters $\boldsymbol{\lambda}_T$ was implemented using the Vienna RNA package version 2.3.5 (Lorenz *et al.*, 2011).

2.3.2 Datasets

In order to evaluate my algorithm, I performed computational experiments on the four datasets assembled by (Rivas *et al.*, 2012), TrainSetA/TestSetA and TrainSetB/TestSetB. TrainSetA and TestSetA were collected from the literature(Dowell and Eddy, 2004; Do *et al.*, 2006; Andronescu *et al.*, 2007; Lu *et al.*, 2009; Andronescu *et al.*, 2010). TrainSetB and TestSetB were extracted from Rfam (Gardner *et al.*, 2010), which contain 22 families with 3D structures. The literature-based sets "A" and the Rfam-based sets "B" are structurally diverse. Furthermore, highly identical sequences were removed from all the four datasets. I excluded a number of sequences that contain pseudoknotted secondary structures in the original data sources from all the four datasets since all algorithms evaluated in this dissertation were designed for RNA secondary structure prediction without pseudoknots. The dataset is also available at https://github.com/keio-bioinformatics/mxfold.

2.3.3 Evaluation measures

I evaluated the accuracy of predicting RNA secondary structures through the sensitivity (SEN) and the positive predictive value (PPV), defined as:

$$SEN = \frac{TP}{TP + FN}, \qquad PPV = \frac{TP}{TP + FP}, \tag{10}$$

where TP is the number of correctly predicted base-pairs (true positives), FP is the number of incorrectly predicted base-pairs (false positives), and FN is the number of base-pairs in the true structure that were not predicted (false negatives). I also used the F-value as the balanced measure between SEN and PPV, which is defined as their harmonic mean:

$$F = \frac{2 \times SEN \times PPV}{SEN + PPV}. \tag{11}$$

2.3.4 Effects of scoring models

In order to confirm the effects of integration of the thermodynamic model and the machine leaning based model, I performed computational experiments on the datasets described in Chapter 2.3.2. The trainable parameters of the machine learning based model were trained from TrainSetA. Each model was evaluated with the prediction accuracy of the Viterbi decoding on TestSetA and TestSetB. Table 2 shows the prediction accuracy of three models: the thermodynamic model (TM) that employs only $f_T(x, y)$ in Eq. (2), the machine learning model (ML) only with $f_W(x, y)$, and my model that integrates the thermodynamic model and the machine learning based model (TM+ML), indicating that my model (TM+ML) performed the most accurate prediction. On TestSetA, my models were slightly better than ML only model. On TestSetB that contains structurally dissimilar RNAs from TrainSetA, the difference of the accuracy between TM+ML and ML is larger.

Table 2. The accuracy of each scoring model

Model	TestSetA			TestSetB		
	SEN	PPV	F1	SEN	PPV	F1
TM	0.682	0.659	0.670	0.598	0.485	0.536
ML	0.703	**0.764**	0.732	0.575	0.550	0.563
TM+ML	**0.715**	0.761	**0.737**	**0.617**	**0.565**	**0.590**

TM: the thermodynamic model, ML: the machine learning based model trained with TrainSetA, and TM+ML: the integrated model.

2.3.5 Effects of feature representations

I evaluated the prediction accuracy of the Viterbi decoding on TestSetA and TestSetB for several feature representations. Fig. 6 shows the accuracy for each feature representation with different context lengths $L = \{0, 3, 5, 7, 10, 15, 20\}$. This indicates that the difference of the accuracy on $L \geq 7$ is negligible although longer sequential features enable more accurate prediction. In addition, as shown in Fig. 7 that shows the running time for each context length, sequential features of longer context lengths need more calculation time. Therefore, I set the default context length $L = 7$ since shorter sequential features decrease the number of

trainable features reducing the risk of overfitting

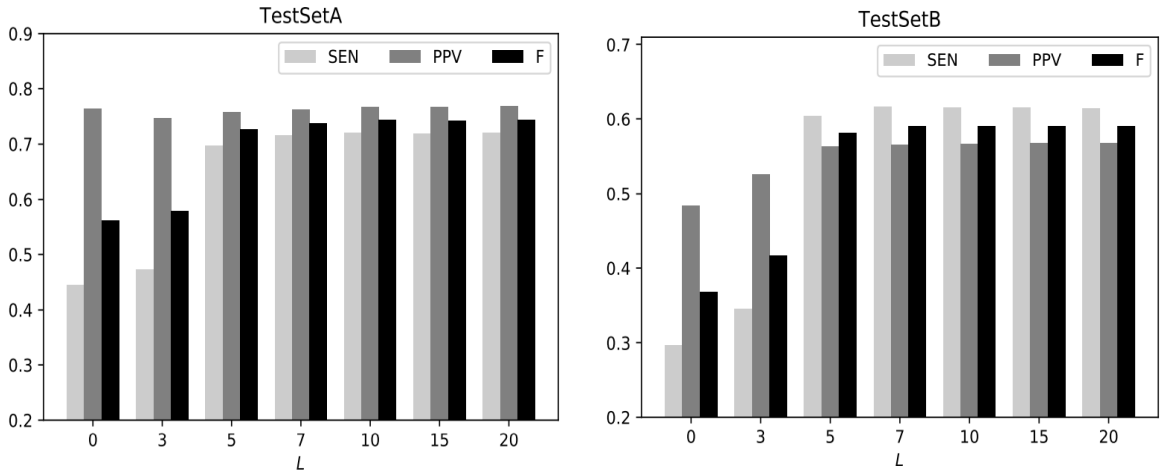


Fig. 6. The accuracy for each feature representation with different context lengths ***L*** on TestSetA (left) and TestSetB (right).

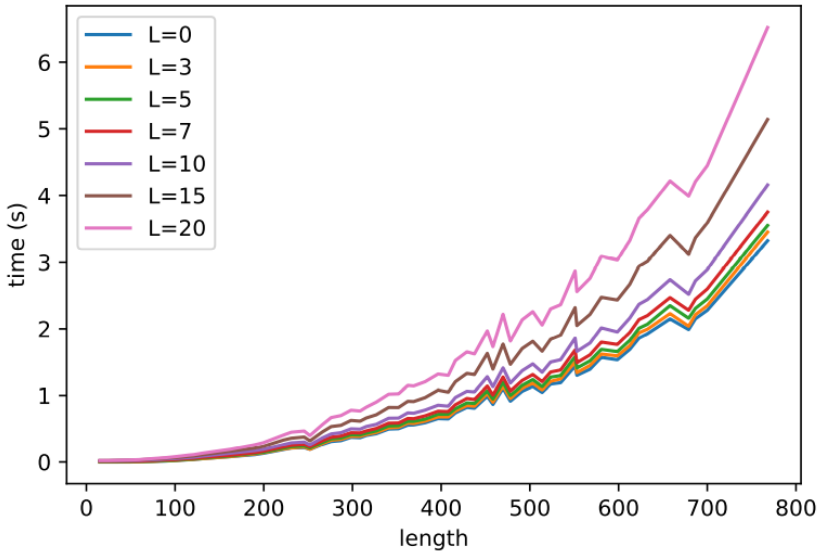


Fig. 7. The running time for each feature representation with different context lengths *L* measured on Red Hat Linux v2.6.32 with Intel Xeon E5-2680 (2.80 GHz) and 64 GB memory.

2.3.6 Comparison with competitive methods

I compared my algorithm with the competitive methods including CentroidFold version

0.0.15 (Hamada *et al.*, 2009; Sato *et al.*, 2009), CONTRAfold version 2.02 (Do *et al.*, 2006), RNAfold in the Vienna RNA package version 2.3.5 (Lorenz *et al.*, 2011) and ContextFold version 1.00 (Zakov *et al.*, 2011). For the posterior decoding methods with the trade-off parameter γ in Eq. (6), I used $\gamma \in \{2^n | n \in Z, -5 \leq n \leq 10\}$.

Fig. 8 shows PPV-SEN plots for each method, indicating that my algorithm works accurately on TestSetA and TestSetB. On TestSetA, ContextFold (F=0.742) is slightly better than MXfold with Viterbi decoding trained from TrainSetA (F=0.737). Whereas, on TestSetB, ContextFold (F=0.496) is much worse than MXfold with Viterbi decoding trained from TrainSetA (F=0.590) and others. Furthermore, MXfold with Viterbi decoding trained from both training datasets performed the most accurate prediction (F=0.626).

Fig. 9 shows the running time for each method for the lengths of input sequences in TestSetA, indicating that my algorithm with the Viterbi decoding is comparable with the other methods in the running time although my algorithm with the posterior decoding is much slower than the other methods.

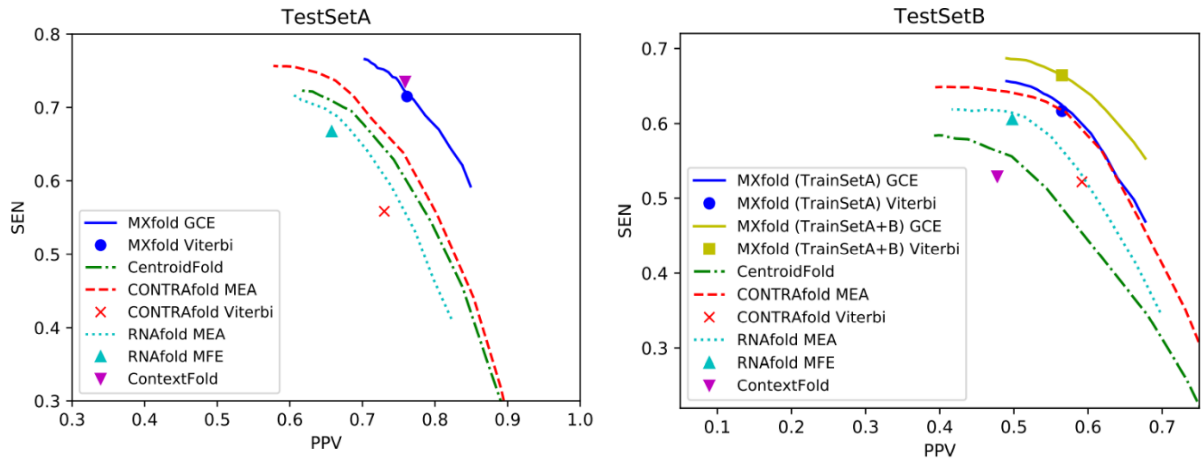


Fig. 8. PPV-SEN plots comparing my algorithm with the competitive methods on TestSetA (top) and TestSetB (bottom).

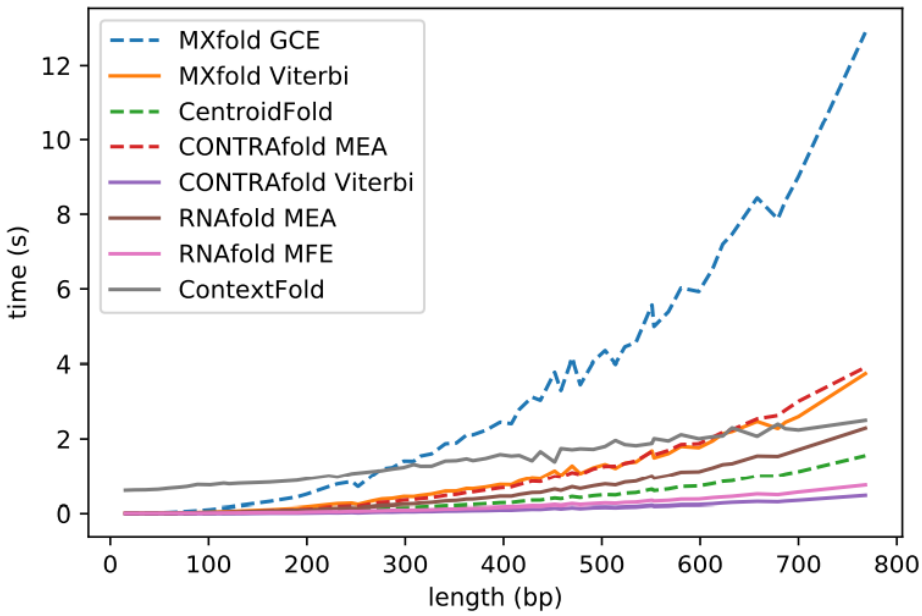


Fig. 9 The running time for the lengths of input sequences measured on Red Hat Linux v2.6.32 with Intel Xeon E5-2680 (2.80 GHz) and 64 GB memory.

2.4 Discussion

Table 2 compares the three models: the thermodynamic model (TM), the machine learning based model (ML) and the integrated model (TM+ML). Since the thermodynamic model $f_T(x, y)$ is implemented using the Vienna RNA package, the prediction result of TM is similar to that of RNAfold. The result on TestSetA indicates that the difference between ML and TM+ML is very small. I can explain that this is because the trainable parameters of ML and TM+ML are identical to each other, and the learning algorithm works well on both models. On the other hand, since the literature-based TrainSetA and the Rfam-based TestSetB are structurally diverse as described in Chapter 2.3.2, TestSetB includes a number of substructures whose scoring parameters cannot be trained from TrainSetA. TM+ML model can calculate scores for such "unobserved" substructures using the thermodynamic energy parameters although ML only model cannot. My integrated model can improve the prediction accuracy by complementing missing parts each other.

I compared the learnability of my model for several context lengths L of sequential features in Fig. 6. Most existing models including RNAfold and CONTRAfold use the context length

$3 \leq L \leq 5$, whose accuracy shown in Fig. 8 is close to that of my model with the same range of the context length. Although Fig. 6 shows that longer context length of sequential features enables us to improve the prediction accuracy, its effects tend to be saturated at $L = 7$. The objective function of my algorithm contains the l_1 regularization term, by which rarely used parameters (e.g., sequential features with $L > 7$) quickly shrink toward zero at line 6 of Fig. 5. Hereby, the risk of overfitting caused by rarely observed features can be reduced. Fig. 8 shows that ContextFold achieved the best accuracy on TestSetA, but the worst on TestSetB. Similarly, the accuracy of CentroidFold on TestSetB remarkably deteriorated compared with that on TestSetA. The common point between ContextFold and CentroidFold is the training data: ContextFold and the Boltzmann likelihood (BL) parameter set used in CentroidFold were trained from the S-Full dataset (Andronescu *et al.*, 2010), which is one of the datasets included in TrainSetA. This suggests that ContextFold and the BL parameter set fell into the overfitting. There is a possibility that ContextFold trained from TrainSetA+B achieves more accurate prediction than MXfold trained from TrainSetA+B. However, ContextFold might not work well for other sequences dissimilar from TrainSet A and B because of the overfitting. Meanwhile, I can expect that my algorithm that integrates the thermodynamic model still performs robust and accurate prediction without overfitting for such sequences due to the integrated thermodynamic model. The posterior decoding algorithms are known to be one of effective approaches for many combinatorial optimization problems (Carvalho and Lawrence, 2008). In fact, the posterior decoding with CONTRAfold (MEA) achieves much better accuracy than its counterpart of the Viterbi decoding as shown in Fig. 8. However, I can surprisingly observe no advantage for the posterior decoding for MXfold (GCE). CONTRAfold was trained by the conditional log-linear models (CLLMs) in which the expectation of the occurrence of features is used for calculating gradients of the objective function. The posterior decoding algorithms employ the base-pairing probabilities that are also calculated by the expectation of the occurrence of base-pairs. This can be interpreted that the optimization with CLLMs is appropriate for the posterior decoding. SSVM used by my algorithm considers only the optimal structure with the (loss augmented) Viterbi algorithm for each training step. This means that SSVM is optimized for the Viterbi decoding, but not for the posterior decoding that considers not only the optimal structures but also the distribution of all possible structures. As shown in Fig. 9, the posterior decoding algorithms are much time-consuming compared with their counterparts of the Viterbi and MFE

algorithms. Therefore, although the posterior decoding with the parameters learned by CLLMs is one of the best solution from the viewpoint in the prediction accuracy, the Viterbi algorithm with SSVM is a practical alternative.

Chapter 3 Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning

In this chapter, I propose Informative RNA base embedding by deep representation learning. Afterwards, I show the results of RNA structural alignment and clustering using informative RNA base embedding. Effective embedding is actively conducted by applying deep learning to biomolecular information. Obtaining better embeddings enhances the quality of downstream analyses, such as DNA sequence motif detection and protein function prediction. In this dissertation, I adopt a pre-training algorithm for the effective embedding of RNA bases to acquire semantically rich representations and apply this algorithm to two fundamental RNA sequence problems: structural alignment and clustering. By using the pre-training algorithm to embed the four bases of RNA in a position-dependent manner using a large number of RNA sequences from various RNA families, a context-sensitive embedding representation is obtained. As a result, not only base information but also secondary structure and context information of RNA sequences are embedded for each base. I call this "informative base embedding" and use it to achieve accuracies superior to those of the existing state-of-the-art methods on RNA structural alignment and RNA family clustering tasks. Furthermore, upon performing RNA sequence alignment by combining this informative base embedding with a simple Needleman-Wunsch alignment algorithm, I succeed in calculating structural alignments with a time complexity of $O(n^2)$ instead of the $O(n^6)$ time complexity of the naive implementation of Sankoff-style algorithm for the input RNA sequence of length n.

3.1 Background

Unstructured data, such as biological sequences and networks, require an embedding operation that encodes the unstructured data into a high-dimensional numerical vector space. This is a necessary step for processing unstructured data in downstream analysis using computational models such as neural networks. In the deep learning field, embedding using the pre-training framework with a large set of unlabelled data has been shown to be effective for the downstream supervised learning task even when smaller size of labelled data is available. When embedding an RNA sequence, each nucleotide (A, C, G, U) is usually encoded to a numerical representation so that the RNA sequence is embedded into a numerical vector. An effective embedding method further attempts to encode contextual information into the numerical vector representation (Fig. 10).

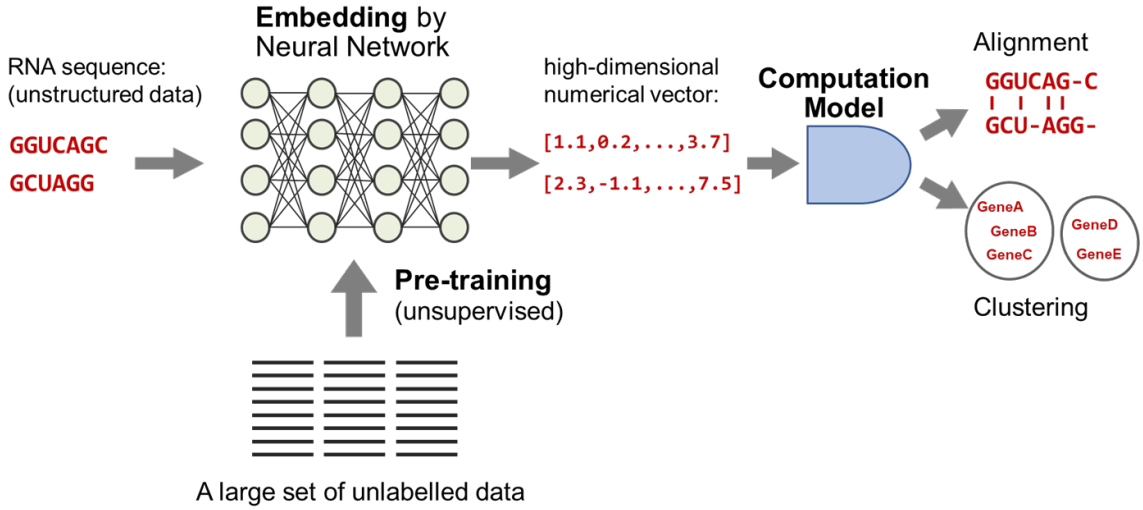


Fig. 10. Schematic view of the pre-training-based embedding and its downstream analysis. The pre-trained neural network with a large set of unlabelled data encodes input DNA sequences into high-dimensional numerical vectors. The embedding by pre-trained neural networks is effective for downstream analysis such as DNA sequence alignment and clustering.

Recently, DNA, RNA, and amino acid sequences have been attempted to be effectively embedded using deep representation learning, especially techniques developed in the field of

natural language processing (Asgari *et al.*, 2019; Heinzinger *et al.*, 2019; Rives *et al.*, 2021). These studies are based on the idea that nucleotide composition and sequence structure determine the motif and function of a gene sequence, just as the complex grammatical structure of natural language determines the meaning of a sentence. As a consequence, word embedding techniques for natural language have been applied to nucleotides for DNA sequences. In the dna2vec method (Ng, 2017), Word2vec is applied to a DNA sequence to obtain the distributed representation of k-mers (a DNA subsequence of length k). Word2vec, an effective word embedding technique (Mikolov *et al.*, 2013) that vectorizes the context and meaning of a word using a large amount of text data, is based on the hypothesis that words with similar meanings have similar peripheral words. Dna2vec adopts the Word2vec technique by defining a k-mer as a word in the DNA sequence; however, since dna2vec assumes a sufficient number of different words used for embedding, the four nucleotides (four words) are not large enough to obtain an effective embedding when dna2vec is applied to base-by-base DNA sequence embedding.

Two recently developed state-of-the-art embedding methods, namely, embeddings from language models (ELMo) and bidirectional encoder representations from transformers (BERT), are used to generate context-sensitive distributed word representations (Peters *et al.*, 2018; Devlin *et al.*, 2019). In these methods, the same word is assigned to different distributed representations depending on the context. In particular, BERT is a pre-training algorithm that obtains word and sentence embeddings by performing two tasks: a masked language modelling (MLM) task and a next sentence prediction (NSP) task. The MLM task predicts multiple masked tokens (words) in a sentence, whereas the NSP task determines whether two statements are consecutive. UniRep (Alley *et al.*, 2019) and PLUS (Min *et al.*, 2021) are representative examples of applying BERT to protein sequence representation; specifically, UniRep obtains the embedding of each amino acid in a protein sequence and uses this embedding to achieve accurate structural and functional predictions of proteins.

In this dissertation, I propose RNABERT for the effective embedding of RNA bases by adopting the pre-training BERT algorithm to ncRNA. I applied informative base embedding to encode the characteristics of each RNA family and structure. To see whether this informative base embedding technique successfully captures these characteristics, I applied RNABERT to two basic RNA sequence analysis tasks: structural alignment and clustering. Then, I evaluated the quality of the informative base embedding results by structural

alignment of RNA sequences and by RNA family clustering.

As shown in Chapter 1, the current RNA structure alignment requires a long calculation time, and a deep learning method for unsupervised RNA family classification has not been developed. For my goals of RNA structural alignment of lower computational complexity and accurate RNA family clustering, I constructed an informative base embedding method, RNABERT, for RNA sequences that takes into account the context and secondary structure of RNA sequences through two training tasks: MLM and structural alignment learning (SAL). In RNABERT, pre-training was performed using a large number of unlabeled ncRNA sequences. RNABERT introduces a novel pre-training task, SAL, in addition to the usual MLM task to more explicitly incorporate RNA secondary structure information into the base embedding for structural alignments. The SAL task employs pre-training using seed alignments obtained from the Rfam database (Kalvari *et al.*, 2018) so that the bases aligned in the seed structural alignment are expected to have more similar embeddings. By alternately training the MLM and SAL tasks, RNA base embedding can be expected to adequately capture the structural differences among RNA families. I compared the accuracy and computational complexity of the structural alignment of RNA sequences between my method and the state-of-the-art methods. Furthermore, I demonstrate that my clustering method is more accurate than the existing state-of-the-art methods in the clustering of RNA families.

3.2 Materials and Methods

3.2.1 The architecture of the RNABERT model

The architecture of the RNABERT model (Fig. 11) consists of three components: token and position embedding, a transformer layer, and pre-training tasks. The input to RNABERT is an RNA sequence. First, the token embedding randomly generates a 120-dimensional numerical vector that encodes four RNA bases (A, C, G, U) and assigns the same vector to each base in the input RNA sequence. Second, the position embedding generates a 120-dimensional vector that encodes the position information of each base in the input RNA sequence. Third, the element-wise sum of token embedding and position embedding for each

base in the input RNA sequence is fed to the transformer layer. The transformer layer component consists of a stack of 6 transformer layers, each of which is composed of a multi-head self-attention mechanism followed by a feedforward neural network. The final output from the transformer layer is an informative base embedding, denoted Z. The weight parameters of the transformer layer are trained by alternately training two different tasks (MLM and SAL) on top of the output of the transformer layer.

The self-attention mechanism (Vaswani *et al.*, 2017) is a central component of the transformer layer. For the transformer layer that takes the output of the previous layer $X = [x_1, \ldots, x_n]$ as input, the multi-head self-attention mechanism with H heads compute the output sequence $C = [c_1, \ldots, c_n]$ with the following formula:

$$C = Concat(head_1, \ldots, head_H)W^O \ ,$$
$$head_i = softmax\left(\frac{(Q_i)(K_i)^\top}{\sqrt{D}}\right)V_i \ , \quad \textbf{(12)}$$

where

$$Q_i = XW_i^Q, K_i = XW_i^K, V_i = XW_i^V.$$

The self-attention mechanism is described as mapping a query and a set of key-value pairs to an output sequence, where the query, key, and value are all matrices: query $Q_i = [q_1^i, \ldots, q_n^i]$, key $K_i = [k_1^i, \ldots, k_n^i]$ and value $V_i = [v_1^i, \ldots, v_n^i]$. These matrices are the inner products of X and the weight matrices W_i^Q, W_i^K, and W_i^V of size $D \times D$ that are learned, where D is the input and output vector dimension (D=120 in this dissertation). In the scaled dot-product attention mechanism, each $head$ calculates the next hidden state by computing the attention-weighted sum of the value vector v. An attention coefficient is the output of the softmax function applied to the dot product of the query and key $(Q_i)(K_i)^\top$ divided by $\sqrt{D}$. Finally, the H $head$ results calculated by different sets of $\{W_i^Q, W_i^K, W_i^V\}$ are concatenated, and the inner product between this concatenated matrix and W^O yields the output sequence C. After the transformer layer process including multi-head attention is performed six times, the informative base embedding denoted Z is obtained. (See the supplementary information and Supplemental Figure S1 for more detailed explanation about the self-attention mechanism.)

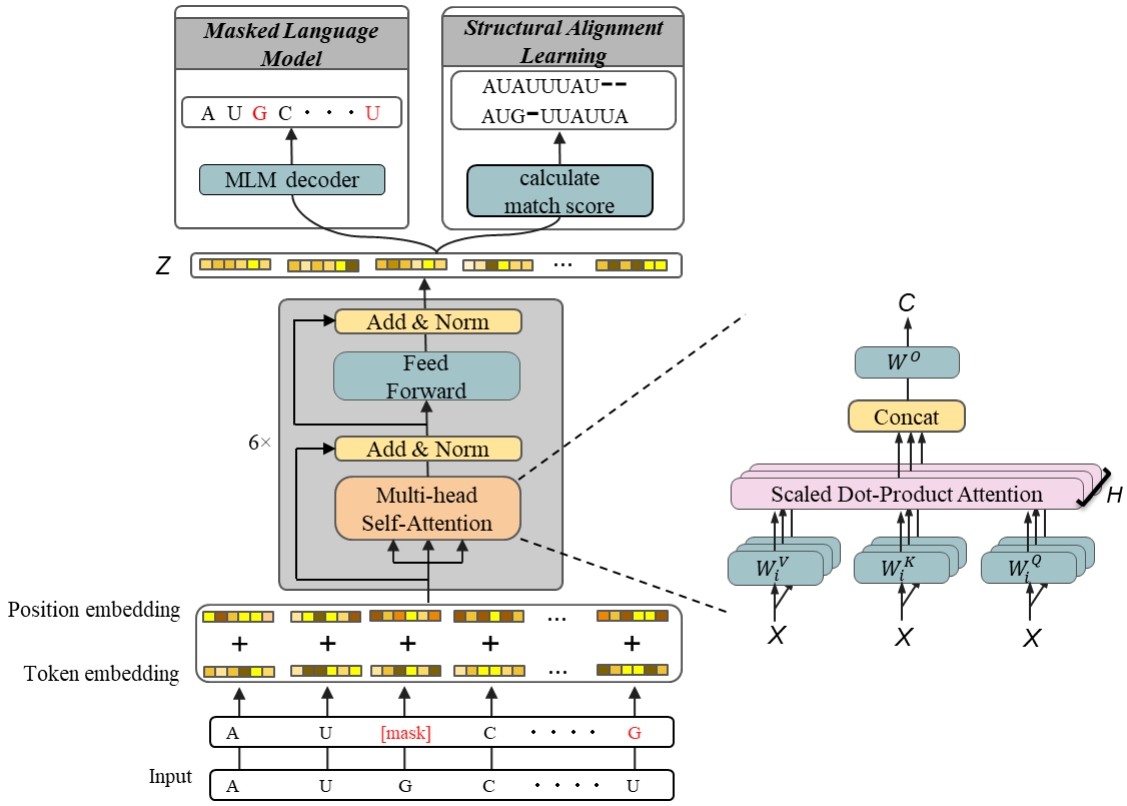


Fig. 11. Architecture of the RNABERT model. The RNABERT model consists of three components: token and position embedding, a transformer layer, and pre-training tasks. Token and position embedding randomly generates a 120-dimensional vector representing four RNA bases. The transformer layer component consists of a stack of 6 transformer layers, each of which is composed of a multi-head self-attention mechanism followed by a feedforward neural network. The final output from the transformer layer is an informative base embedding, denoted Z. The weights of the transformer layer are trained by alternately training two different tasks (MLM and SAL) on top of the output of the transformer layer.

3.2.2 Masked language modelling (MLM)

MLM is a task that masks a part of the input RNA sequence and predicts the masked part using the surrounding bases. The MLM task performs a base embedding so that the masked part can be restored, which enables context-sensitive embedding. First, 15% of the bases are randomly selected in a given RNA sequence for training. Next, one of the following three actions is performed on the selected base in the input: 80% of the selected bases are replaced with a token indicating an unspecified base (denoted [mask] in Fig. 11), 10% are randomly substituted with one of the other three bases, and the remaining 10% of the selected bases are unchanged from their original base. The MLM task trains the model to maximize the probability of correctly predicting the selected 15% of the RNA bases at the output. In this training model, a classification layer is built on top of the output of the transformer layer. Finally, the output probability of each base is calculated using the softmax function. The cross-entropy function is used as the loss function. The pre-training set for the MLM task consists of 762,370 sequences generated from 76,237 human ncRNA sequences obtained from RNAcentral (The RNAcentral Consortium *et al.*, 2017) by taking 10 copies of each ncRNA and applying 10 different mask patterns to each.

3.2.3 Structural alignment learning (SAL)

The SAL task, which performs a base embedding task to learn the relationship between two RNA sequences, is based on RNA structural alignment. RNA structural alignment aligns multiple RNA sequences by inserting gaps between bases so that the conserved secondary structures are aligned in the same column. The SAL task aims to obtain closer embeddings for bases in the same column of reference alignment and obtain secondary structure embeddings by training based on the RNA structural alignment. The Rfam seed alignment for each family is downloaded from Rfam (Kalvari *et al.*, 2018) as the reference structural alignment for the SAL task. To define the loss function in the SAL task, I introduce the Ω matrix, which is defined for a pairwise alignment of two RNA sequences and is intended to be used as a score matrix when calculating the pairwise alignment. Let $Z = [z_1, \ldots, z_n]$ and $Z' = [z'_1, \ldots, z'_m]$ denote the embedded representations output from the transformer layer for

the input of two RNA sequences of length n and m. Each element ω_{ij} in the Ω matrix is defined to be the normalized inner product between z_i and z'_j:

$$\omega_{ij} = \frac{z_i \cdot z'_j}{\|z_i\| \|z'_j\|}. \tag{13}$$

The loss function in the SAL task is defined to increase ω_{ij} at the matched position in the reference alignment so that a sequence alignment algorithm such as the Needleman-Wunsch algorithm produces the reference alignment.

A simple way to implement this loss function in the SAL task is to apply binary classification learning with respect to ω_{ij}. That is, ω_{ij} in the aligned position is trained to 1, and ω_{ij} in an unaligned position is trained to 0. However, this causes strong overfitting. To alleviate this problem, I apply a machine learning method called a structured support vector machine (Akiyama *et al.*, 2018; Tsochantaridis *et al.*, 2005) to the pre-training phase in the SAL task. Let the alignment between a pair of RNA sequences $x = x_1, \ldots, x_n$ and $x' = x'_1, \ldots, x'_m$ be represented by a series of matched (aligned) positions (i, j) and gap insertion positions $(i, -)$ or $(-, j)$, where $1 \leq i \leq n, 1 \leq j \leq m$. For a given training dataset D consisting of triplets (x, x', y), where x and x' are a pair of RNA sequences and y is the corresponding reference alignment between x and x', I aim to find a set of parameters w that minimize the following loss function L:

$$L = \sum_{(x,x',y)\in D} \{f(x, x', \hat{y}) + \Delta(y, \hat{y}) - f(x, x', y) + \lambda\|w\|_2\}, \tag{14}$$

where f is the function that returns the alignment score y between x and x'. The term $\lambda\|w\|_2$ in the above formula is the L2 regularization term to avoid overfitting, where w refers to the parameters of the entire model, $\|w\|_2$ is the squared value of the model parameters and λ is a parameter that controls the strength of regularization. The alignment score is calculated as the sum of the ω_{ij} value at the matched position (i, j) and the gap score at the gap insertion positions $(i, -)$ or $(-, j)$. $\hat{y}$ is the predicted alignment path calculated by the Needleman–Wunsch algorithm to maximize the sum of the alignment score $f(x, x', \hat{y})$and the margin

term $\Delta(y, \hat{y})$. The margin term $\Delta(y, \hat{y})$ defines the difference between the reference alignment and the predicted alignment as follows:

$$\Delta(y, \hat{y}) = \delta^{FN} \times (\text{the number of positions included in } y \text{ but not in } \hat{y}) + \delta^{FP} \times (\text{the number of positions included in } \hat{y} \text{ but not in } y). \quad \textbf{(15)}$$

Here, δ^{FN} and δ^{FP} are hyperparameters that control the trade-off between sensitivity and specificity for learning parameters. By default, I used δ^{FN} = 0.05 and δ^{FP} = 0.1, which were determined by the grid-search optimization in the range 0.01-0.30. Decreasing the loss function *L* brings the predicted alignment closer to the reference alignment.

3.2.4 RNA structural alignment

A pairwise RNA sequence alignment based on the base embedding is calculated using the Needleman-Wunsch algorithm using the Ω matrix as the score matrix, which is trained in the SAL and MLM tasks. The match score in position (i, j) is ω_{ij} in the score matrix Ω, and the gap opening score and gap extension score are set to -1 and -0.1, respectively. As the MLM task enables the position- and context-sensitive embedding and SAL task enables the structural information embedding, the Needleman-Wunsch algorithm, a simple sequence alignment algorithm, is expected to generate RNA structure alignments using the Ω matrix trained in the SAL and MLM tasks. Note that the time complexity of the Needleman-Wunsch algorithm is $O(n^2)$ for the input RNA sequence of length *n*.

3.2.5 RNA family clustering

RNA family clustering is performed as the second evaluation test to confirm the quality of the informative base embedding. A similarity measure between two RNA sequences with respect to soft symmetric alignment (Bepler and Berger, 2019) is defined as follows. Let $Z = [z_1, \ldots, z_n]$ and $Z' = [z'_1, \ldots, z'_m]$ denote the embedded representations output from the transformer layer for the input of a pair of RNA sequences of length *n* and *m*. The similarity $\hat{s}$ between the two RNA sequences is defined to be the weighted sum of the normalized inner

product between all z_i and z_j' pairs:

$$\hat{s} = \frac{1}{A}\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij}\omega_{ij} \ , \qquad \omega_{ij} = \frac{z_i \cdot z_j'}{\|z_i\|\|z_j'\|} \ , \qquad A = \sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \tag{16}$$

where a_{ij} is

$$a_{ij} = \alpha_{ij} + \beta_{ij} - \alpha_{ij}\beta_{ij} \ ,$$

$$\alpha_{ij} = \frac{exp(\omega_{ij})}{\sum_{k=1}^{m} exp(\omega_{ik})} \ ,$$

$$\beta_{ij} = \frac{exp(\omega_{ij})}{\sum_{k=1}^{n} exp(\omega_{kj})} \ .$$

The similarity $\hat{s}$ is calculated for all pairs of ncRNA sequences to be clustered, and a classification matrix of size $N \times N$ is created, where N is the number of RNA sequences in the test dataset. I applied spectral clustering to the rows of the classification matrix by considering each row of the N-dimensional vector a cluster indicator. To confirm the improvement in the embedding quality by the SAL task, I compared the clustering accuracy when using only the MLM task with that when using the two tasks together.

3.2.6 Existing methods for RNA structural alignment

There is a family of Sankoff-style algorithms for structural alignment that simultaneously predicts the optimal alignment and the consensus secondary structure. For example, Dynalign and Foldalign (Sundfeld *et al.*, 2015; Fu *et al.*, 2014) use thermodynamic models to find MFE consensus structures, while PARTS (Harmanci *et al.*, 2008) uses a probabilistic model based on the pseudo-energy obtained from base-pairing probabilities and alignment probabilities to find the most likely structural alignment. While Sankoff-style algorithms yield a high alignment accuracy, the naive implementation is computationally expensive, with a time complexity of $O(n^6)$ for RNA sequences of length n. PMcomp takes base-pairing probability matrices generated using McCaskill's algorithm as the input and incorporates the energy information of each sequence into these matrices to quickly find common secondary structures and alignments (Hofacker *et al.*, 2004). Although LocARNA (Will *et al.*, 2007) is

based on the PMcomp model, a time complexity of $O(n^4)$ is achieved by simplifying the dynamic programming method utilizing the fact that the base-pairing probability matrix is actually sparse. SPARSE (Will *et al.*, 2015) takes further advantage of this sparsity based on the conditional probabilities of bases and base pairs in the loop region of the RNA secondary structure, achieving a quadratic improvement in the computational time over LocARNA. RAF (Do *et al.*, 2008) achieves the same time complexity as SPARSE by utilizing the sparseness of alignment candidates. DAFS is a state-of-the-art accurate structural alignment program utilizing integer programming technique (Sato *et al.*, 2012) and its time complexity is $O(n^3)$. R-coffee is a multiple RNA alignment package that takes a similar strategy with my dissertation by utilizing an alignment-scoring scheme that incorporates secondary structure information (Wilm *et al.*, 2008) and its time complexity is $O(n^2)$. TOPAS is a network-based scheme for pairwise structural alignment of RNAs that can handle pseudoknots (Chen *et al.*, 2019) and its time complexity is $O(n^4)$ in the worst case. TOPAS employs graph data structures to represent the RNA secondary structure including pseudoknots and designs an efficient algorithm to calculate an alignment of two graph structures by matching two nodes in two different graphs. Finally, MAFFT v7 (Katoh and Standley, 2013), which uses Kimura's two-parameter model (Kimura, 1980) as the score matrix, was adopted as the baseline for RNA sequence alignment. Note that MAFFT is a sequence-based alignment algorithm that does not take RNA structure information into account.

3.2.7 Existing methods for RNA family clustering

The clustering accuracies of the state-of-the-art methods GraphClust (Heyne *et al.*, 2012), EnsembleClust (Saito *et al.*, 2011), and CNNclust (Aoki and Sakakibara, 2018) were compared. CNNclust is a deep learning-based algorithm that performs supervised learning in which the RNA family class is given as a label. CNNclust can classify RNA families that are not used for training by calculating the similarity score matrix for all pairs of input sequences. I performed experiments with CNNclust using different RNA family groups between training and testing. In contrast, GraphClust is an unsupervised learning algorithm that does not require the RNA family class to be a label and achieves alignment-free clustering with some exceptions. GraphClust employs a graph kernel approach to obtain feature vectors that contain both sequence and secondary structure information. These vectors representing RNA

sequences are clustered with a linear time complexity over the number of sequences using a hashing technique. Finally, EnsembleClust calculates the similarity between two ncRNAs using the expected structural alignment and then applies hierarchical clustering based on the similarity.

3.2.8 Sequence motif detection using a self-attention mechanism

I extracted the sequence motifs specific to each RNA family by focusing on the self-attention mechanism, which determines where to focus on the input embedding vectors $X = [x_1, \ldots, x_n]$ of the input RNA sequence $r = r_1, \ldots, r_n$ when generating the output sequence. The attention coefficient sequence $M = [m_1, \ldots, m_n]$, called attention map, that is calculated for the input sequence $r = r_1, \ldots, r_n$ is defined as follows:

$$M = \sum_{h=1}^{H} \sum_{i=1}^{n} softmax\left(\frac{\left(q_i^h\right)(K_h)^\top}{\sqrt{D}}\right). \quad \textbf{(17)}$$

The base r_i at position i with a high m_i value is identified as part of the motif. Thus, the attention map helps discover the sequence motif since it indicates a base that is important for training tasks. (See Supplemental Figure S2 for more detailed explanation about RNA motif detection using self-attention map).

3.2.9 Measures of the accuracies of alignment and clustering

Structural alignment accuracy was measured using sensitivity, positive predictive value (PPV), and F1 score, which are calculated as follows. The number of true positives (TP) (or false positives (FP)) is the number of positions (i, j) in the predicted alignment that belong (or do not belong) to the reference alignment. The sensitivity of the predicted alignment is TP divided by the number of positions in the reference alignment, and the PPV is TP divided by the number of positions in the predicted alignment. The F1 score is the harmonic mean of sensitivity and PPV.

Clustering accuracy was measured with the Rfam family as the true reference class. Three

indices, namely, the adjusted Rand index (ARI), homogeneity, and completeness, were used to evaluate the clustering performance. The ARI is a measure of how well two types of clustering results match. ARI takes a real number from -1 to 1: if the value of ARI is -1, the two clustering results do not match at all, while a value of 1 indicates that they completely match. In this dissertation, the ARI reflects how close the predicted clustering result is to the true reference class composed of the Rfam family.

The ARI is derived from the Rand index (RI), defined as follows:

$$RI = \frac{TP + TN}{TP + TN + FP + FN}$$
$$E = \frac{(TP + FP)(TP + FN) + (TN + FP)(TN + FN)}{TP + TN + FP + FN} \quad \textbf{(18)}$$
$$ARI = \frac{(TP + TN) - E}{(TP + TN + FP + FN) - E}$$

where TP is the number of RNA sequences of the same Rfam family in the same predicted cluster, TN is the number of RNA sequences of a different Rfam family in different predicted clusters, FP is the number of RNA sequences of different Rfam families in the same predicted cluster, and FN is the number of RNA sequences of the same Rfam family in different predicted clusters. Homogeneity is a measure of the proportion of RNA sequences of a single Rfam family that belong to a single predicted cluster, and completeness measures the proportion of RNA sequences of a particular Rfam family that are assigned to the same predicted cluster.

3.2.10 Datasets

For the pre-training of the MLM task, 76,237 human-derived small ncRNAs with lengths ranging from 20 to 440 bases from RNAcentral (The RNAcentral Consortium *et al.*, 2017) were utilized.

In the training of the SAL task, two types of datasets, named TrainSet-A and TrainSet-B, were devised. In both datasets, the pairwise structural alignment extracted from Rfam alignment (Kalvari *et al.*, 2018) was used. TrainSet-A consists of RNA sequences sampled from seed RNA sequences in 36 RNA families (5.8S rRNA, 5S rRNA, Cobalamin, Entero 5

CRE, Entero CRE, Entero OriR, gcvT, Hammerhead 1, Hammerhead 3, HCV SLIV, HCV SLVII, HepC CRE, Histone3, HIV FE, HIV GSL3, HIV PBS, Intron gpII, IRES HCV, IRES Picorna, K chan RES, Lysine, TAR, Retroviral psi, S box, SECIS, sno 14q I II, SRP bact, SRP euk arch, T-box, THI, tRNA, U1, U2, U6, UnaL2, yybP-ykoY) in which all families were overlapped with the following structural alignment benchmark dataset. TrainSet-B consists of RNA sequences from all RNA families (3,983 families) in Rfam database except the RNA families used in the benchmark dataset BRAliBase2.1 k2 dataset (Wilm *et al.*, 2006), that is, the training and test datasets do not overlap with respect to the RNA family.

For the structural alignment benchmark, I utilized the BRAliBase2.1 k2 dataset used in the previous study as the gold standard benchmark dataset. Sequence pairs containing unknown bases were eliminated. A total of 8,587 RNA sequence pairs with an average length of approximately 100 bases were used for the benchmark test dataset. No alignment overlapped between the training dataset of the SAL task and the benchmark test dataset. Note that no alignment overlapped between TrainSet-A and the benchmark test dataset.

To evaluate the clustering accuracy of RNABERT, the test dataset was collected from the BRAliBase2.1 database. The multiple alignment of each ncRNA family provided by the database was treated as a true reference cluster, and each ncRNA sequence in the multiple alignment was treated as a member sequence. All reference clusters with a sequence identity of less than 40% were selected. The dataset contained 37 RNA sequences and 12 RNA families. The RNA sequences used in the RNA family clustering test did not overlap with those used for the pre-training of the SAL task.

3.2.11 Implementation

The RNABERT model was implemented using PyTorch for deep learning. All experiments were run on Red Hat Linux v4.8.5 (GPU: Tesla v100, CPU: Intel(R) Xeon(R) Gold 6148). Optuna (Akiba *et al.*, 2019) was used to find the optimal hyperparameters for the MLM task. The hyperparameters optimized for the transformer layer were the number of attention heads, number of transformer layers, feature size, activation function, and training algorithms, including Adam, AdaGrad, and momentum stochastic gradient descent (SGD). In the MLM task, 5-fold cross-validation was performed, and the hyperparameters were determined to maximize accuracy.

3.3 Results and Discussion

3.3.1 Pre-training of base embedding encodes properties of RNA secondary structure

To investigate whether RNABERT acquired an informative base embedding to encode four RNA bases and secondary structure information, the embedded representations output from the transformer layer for a set of RNA sequences were projected into two-dimensional space using t-distributed stochastic neighbour embedding (t-SNE) (van der Maaten and Hinton, 2008), which is a dimension reduction algorithm for mapping high-dimensional data to low dimensions. Fig. 12 shows the result of mapping the 120-dimensional vector of each base into a two-dimensional space (with the option "n_components=2"). In the dimension reduction by t-SNE, the distance relationship between bases embedded in the original 120-dimensional space is projected in two dimensions so as to be preserved as much as possible. The embedding space adequately represents the clusters for four RNA bases (Fig. 12, left) and the subclusters for characteristic secondary substructures (Fig. 12, right). Fig. 12 shows that the RNA base embedding is globally separated by four RNA bases and locally separated by characteristic secondary substructures (hairpin loop, base pair in stem, and external loop) within each RNA base. This result clearly shows that RNABERT embedding using pre-training with SAL and MLM tasks succeeded in encoding not only base (nucleotide) information but also secondary structure information. (See the Supplemental Figure S3 for t-SNE projection of embedding for all secondary substructures.)

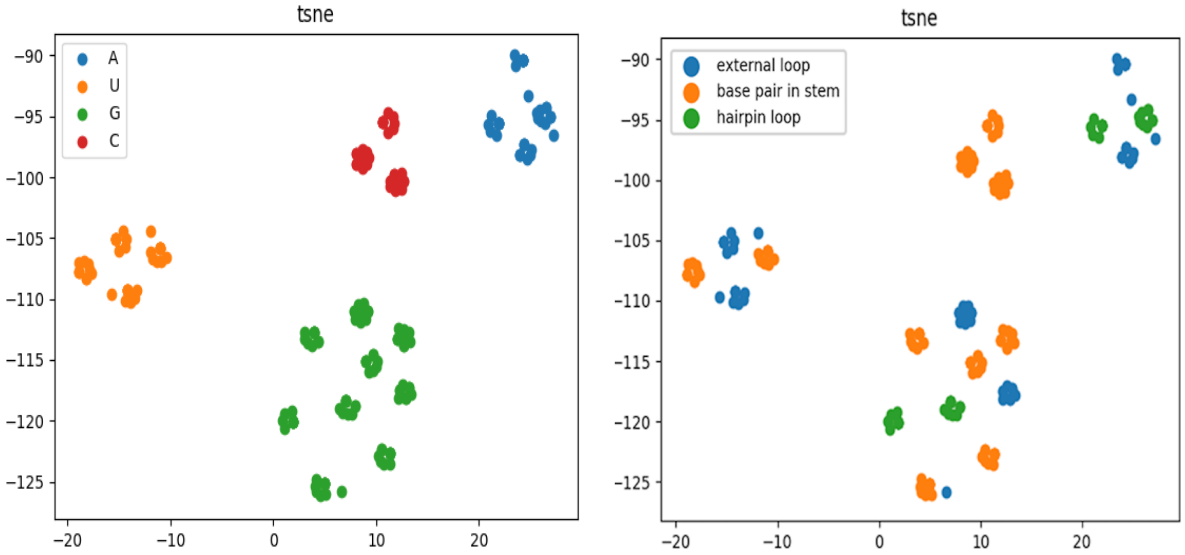


Fig. 12. Visualization of RNA base embedding. Shown is a t-SNE projection from a 120-dimensional embedded space to a two-dimensional space. RNA base embeddings are visualized with colours according to the type of RNA base (left) and the type of characteristic secondary substructure (right). The embedding space adequately represents the clusters for four RNA bases (left) and the subclusters for characteristic secondary structures (right).

3.3.2 RNA structural alignment result

Table 3 summarizes the performance evaluation results based on the BRAliBase2.1 k2 dataset for my RNA structural alignment method, RNABERT trained on TrainSet-A and TrainSet-B, and for the state-of-the-art algorithms for RNA sequence alignment. As shown in Table 3, RNABERT trained on TrainSet-A outperformed the existing state-of-the-art structural alignment algorithms in all three measures of accuracy. On the other hand, the performance of RNABERT trained on TrainSet-B was still sufficiently high and almost same as the one using TrainSet-A. This result indicates that RNABERT has the sufficient generalization ability when trained on a large set of RNA families.

In terms of computation time, RNABERT was faster than the existing state-of-the-art algorithms and even faster than the sequence-based (non-structural) alignment algorithm MAFFT. The alignment computation of RNABERT consists of three sub-procedures: the first procedure (transformer) obtains the embedding of each base; the second procedure calculates the match score between the two input sequences; and the third procedure calculates the alignment by the Needleman-Wunsch algorithm. The first two procedures can

be accelerated by GPU computation, and the Needleman-Wunsch algorithm is a simple algorithm that requires a computation time of $O(n^2)$ for two sequences of length n. I achieved high-speed computation by implementing the deep learning algorithm using Python and PyTorch while implementing the Needleman-Wunsch algorithm in C++. Note that the loading time of the transformer model into the GPU was excluded from the time measurement of pairwise alignment by RNABERT. The typical amount of time needed to load the transformer model onto GPU was around 4.376 seconds. In addition, the maximum "CPU" memory consumption for the RNA structural alignment was around 35.2G bytes in RNABERT.

Table 3. RNA structural alignment accuracies and computational times (shown in seconds) of RNABERT and state-of-the-art algorithms.

	Sensitivity	PPV	F1	Time (sec)
RNABERT (TrainSet-A)	**0.881**	**0.947**	**0.913**	288
RNABERT (TrainSet-B)	0.851	0.932	0.890	**284**
LocaRNA	0.862	0.922	0.891	13,221
SPARSE	0.848	0.931	0.888	4,216
RAF	0.865	0.938	0.900	1,423
PARTS	0.860	0.931	0.894	432,585
Dyalign2	0.706	0.913	0.796	601,104
R-coffee	0.842	0.934	0.886	878
TOPAS	0.879	0.938	0.908	2,103
Foldalign	0.861	0.922	0.890	451,112
DAFS	0.862	0.936	0.897	2,210
MAFFT	0.810	0.901	0.853	1,282

Fig. 13 shows the sensitivity (denoted SEN) and PPV curves calculated for each RNA sequence alignment algorithm. These values were plotted by sequence identity. As shown in Fig. 13, RNABERT yielded very accurate structural alignment results and outperformed the existing state-of-the-art structural alignment algorithms where the sequence identity exceeded 50%. At lower sequence identities, the alignment accuracy of RNABERT was slightly lower than those of LocARNA, SPARSE and Foldalign, which required larger computation times, and was higher than that of RAF, which exhibited the fastest computational time among the existing structural alignment algorithms. All existing Sankoff-style algorithms conduct RNA secondary structure predictions to calculate the distances and similarities between RNA sequences. On the other hand, RNABERT does not explicitly use secondary structure predictions, which implies that the RNA base embedding efficiently captures structural information. In particular, for sequences with very low sequence identities, the accuracy of

the sequence-based alignment MAFFT and R-coffee tend to decrease, while RNABERT and the existing structural alignment algorithms maintain high accuracy.

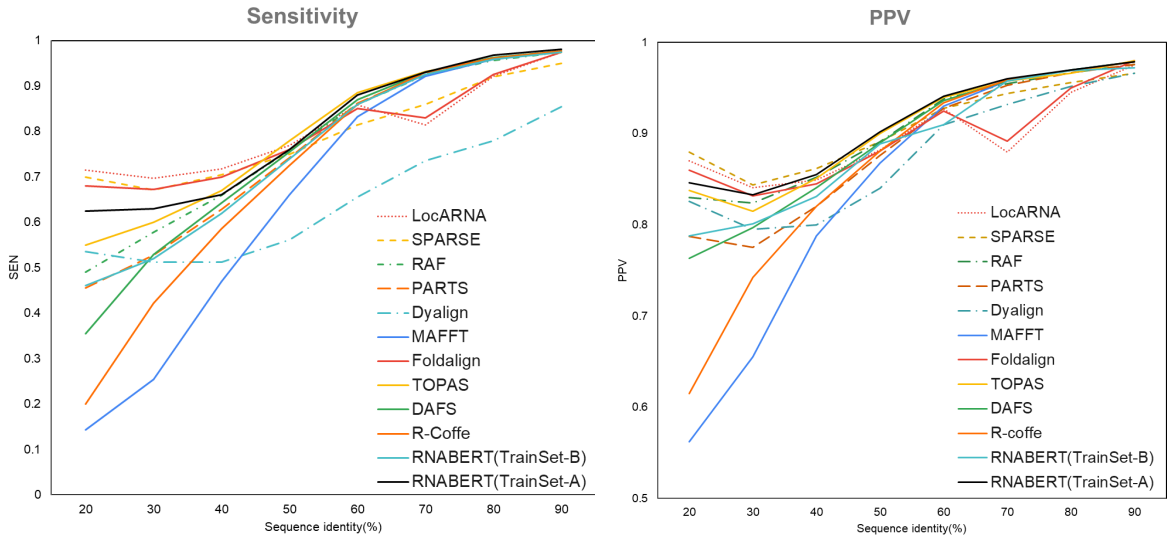


Fig. 13. SEN and PPV score plots for pairwise RNA structural alignments using RNABERT, LocARNA, SPARSE, RAF, PARTS, Dynalign, Foldalign, TOPAS, DAFS, R-Coffee and a sequence-based alignment using MAFFT.

3.3.3 RNA family clustering results

Table 4 shows the ARI, homogeneity and completeness of my RNA clustering method, RNABERT, and those of the state-of-the-art tools for RNA family clustering. RNABERT (TrainSet-A) with the MLM and SAL tasks achieved the highest ARI and completeness among all state-of-the-art tools. The existing methods all utilize RNA secondary structure predictions to calculate the distances and similarities between RNA sequences. This implies that the RNABERT base embedding, which does not explicitly use secondary structure prediction but uses the same RNA family for SAL task, efficiently captures structural information. On the other hand, the performance of RNABERT (TrainSet-B) trained on different RNA families is less accurate compared with GraphClust and similar with CNNclust. This result indicates that the SAL task designed for effective structural alignment, but not for family clustering, is not sufficient for unknown RNA family clustering.

Table 4. RNA family clustering accuracy. The ARI, homogeneity and completeness are shown for RNABERT and the state-of-the-art tools for RNA family clustering.

	ARI	Homogeneity	Completeness	Time (sec)
RNABERT (TrainSet-A) (MLM + SAL)	0.268	0.663	0.758	28.69
RNABERT (TrainSet-B) (MLM + SAL)	0.187	0.568	0.664	27.16
RNABERT (MLM)	0.177	0.556	0.663	27.81
CNNclust	0.189	0.612	0.642	17.45
EnsembleClust	0.200	0.587	0.661	11.32
GraphClust	0.243	0.746	0.666	520.22

3.3.4 RNA motif

Several well-known sequence motifs in the snoRNA and tRNA families were identified by observing the attention maps. Attention maps, which indicate the ratios of contribution to the MLM task, were extracted from the final transformer layer of RNABERT, and sequence motifs were detected from the attention maps.(See the supplementary information and Supplemental Figure S2 for more detailed explanation about the self-attention mechanism.) The "UUCGA" sequence motif shown in Fig. 14a is typical in the T loop of tRNA (Laslett and Canback, 2004). This motif is specifically present in TRT-AGT6-1 (tRNA gene with anticodon AGT), as displayed in the secondary structure in Fig. 14b. The motifs depicted in Fig. 14c are the typical motifs "UGAUGA" and "CUGA" present in the snoRNA C/D box (Ganot *et al.*, 1997; Samarsky *et al.*, 1998). These motifs are specifically present at SNORD113-7, as displayed in the secondary structure in Fig. 14d.

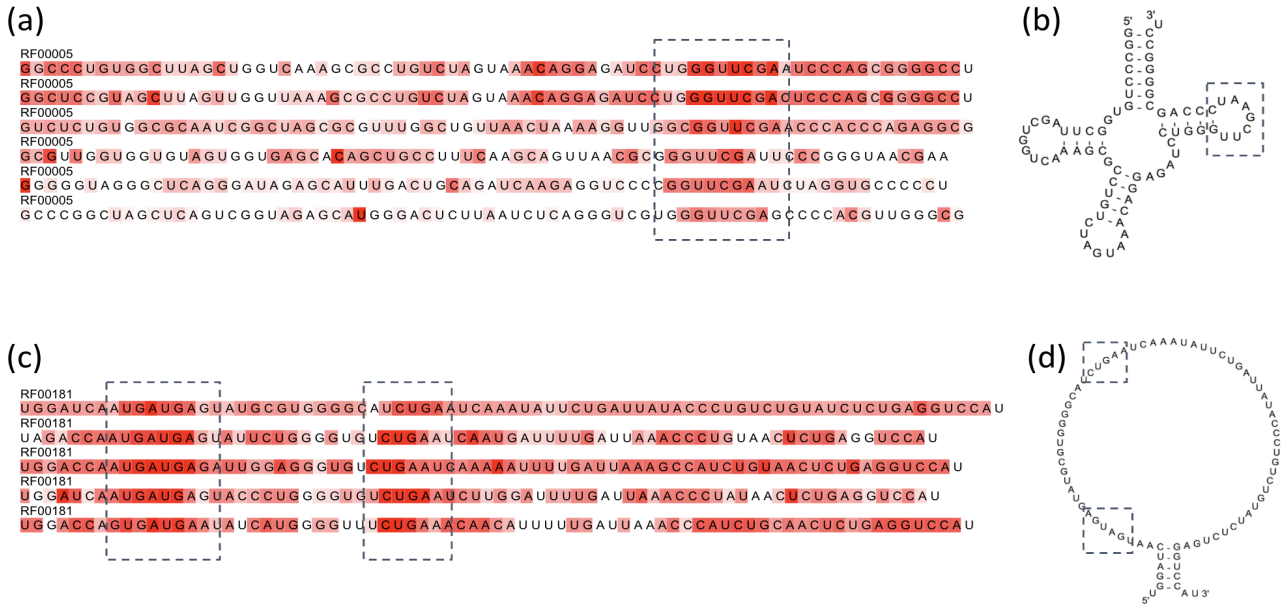


Fig. 14. Extracted sequence motifs of tRNA ((a), (b)) and snoRNA families ((c), (d)). (a) and (c) are visualizations of the attention map at each base. Bases with darker red backgrounds have higher attention map values.

Chapter 4 Conclusion and future work

In this dissertation, I searched for an effective analytical method of ncRNA by two different approaches, RNA secondary structure prediction and informative base embedding.

In the first part of the dissertation, I proposed a novel algorithm for RNA secondary structure prediction that integrates the thermodynamic approach and the machine learning based weighted approach. The fine-grained model combined the experimentally determined thermodynamic parameters with a large number of scoring parameters for detailed contexts of features that were trained by the structured support vector machine (SSVM) with the l_1 regularization to avoid overfitting. Benchmarking analysis showed that the algorithm achieved the best prediction accuracy compared with existing tools, and heavy overfitting as seen in ContextFold cannot be observed. Accurate secondary structure prediction for long RNA sequences has been in demand, since the number of long non-coding RNAs (lncRNAs) have recently been on the rise. To respond to such demand, I would need to implement the sparsification technique (Backofen *et al.*, 2011) to the proposed algorithm with the Viterbi decoding. As shown in Fig. 9, ContextFold that implements the sparsification technique enables us fast structure prediction even for long sequences. The base-pairing probabilities calculated from the posterior distribution are required for various applications for RNA informatics such as family classification (Sato *et al.*, 2008; Morita *et al.*, 2009), pseudoknotted RNA secondary structure prediction (Sato *et al.*, 2011), RNA-RNA interaction prediction (Kato *et al.*, 2010) and simultaneous aligning and folding (Sato *et al.*, 2012). Accurate base-pairing probabilities calculated by my algorithm can improve the quality of such applications.

Next, I performed two tasks to obtain informative base embeddings. While MLM task is a fundamental step in the original BERT algorithm, SAL is a novel RNA sequence-specific task introduced in this dissertation. To determine whether these tasks effectively incorporate RNA secondary structure information into base embeddings, I performed two tests, RNA clustering and sequence alignment.

Sankoff-style algorithm provides high structural alignment accuracy, but these algorithms are usually very complex in both time and space. Unlike many structural alignment algorithms based on the Sankoff algorithm, RNABERT does not explicitly consider RNA

folding and boasts a high structural alignment accuracy. This is considered to be evidence that the base embedding encodes the secondary structure information specific to RNAs. Furthermore, while RNABERT achieves the same accuracy as Sankoff-style algorithms, it is much faster because it uses a simple sequence-based alignment algorithm. In fact, the time complexity of the RNABERT algorithm is only $O(n^2)$ for two sequences of length n.

SPARSE (Will *et al.*, 2015) achieves a quadratic improvement in the computational time of Sankoff-style algorithms for simultaneous alignment and folding by assuming that RNA secondary structures are sparse. Similarly, RNABERT also achieves a quadratic computational time improvement by reducing the RNA structural alignment problem to a sequence alignment problem based on the pre-training of base embeddings. In this way, the computational time of RNABERT was an order of magnitude faster than that of SPARSE, as revealed in this dissertation.

The performance evaluation was done for two types of training datasets, TrainSet-A and TrainSet-B. TrainSet-A contains the same RNA families as the benchmark test dataset while TrainSet-B has no RNA family overlap with the test dataset. When TrainSet-A was used, RNABERT exhibited a superior accuracy than state-of-the-art existing structural alignment methods. When TrainSet-B was used, the performance of RNABERT was still sufficiently high and almost same as the one using TrainSet-A. This result shows that RNABERT has succeeded in proposing a new scoring scheme for sequence-based alignment algorithms to accomplish RNA structural alignment, and has the sufficient generalization ability. In addition, with the development of high-throughput sequencing, hundreds of thousands of ncRNAs have been detected, but many have not been annotated. In fact, 86% (24,972,896) of the 28,895,596 ncRNAs present in RNAcentral do not have gene ontology (GO) annotations. Therefore, fast and accurate structural alignment of unknown sequences of existing RNA families is still practically valuable and RNABERT could contribute to the annotation of such novel transcripts.

The base embeddings obtained by RNABERT are applicable to various fields in RNA informatics. One immediate problem is the multiple structural alignment of RNA sequences. RNABERT can be expected to accomplish this task by combining existing sequence-based multiple alignment algorithms such as MUSCLE (Edgar, 2004) and MAFFT (Katoh and Standley, 2013) with the score matrix Ω and informative base embedding. Another area most likely to improve with the application of RNABERT is the prediction of RNA secondary

structures. Since the base embeddings contain information on secondary structures, RNABERT is expected to contribute to the prediction of RNA secondary structures (Sato *et al.*, 2021; Edgar, 2004; Katoh and Standley, 2013). Similarly, base embeddings can be applied to the RNA interactome (RNA-protein interaction, RNA-RNA interaction), in which the RNA secondary structure acts on the interaction between molecules. In order to accomplish such secondary structure-related problems, it would be a better approach to incorporate the secondary structure prediction as another pre-training task in the pre-training process of RNABERT. Finally, while this dissertation has not addressed RNA modification (e.g., m6A, m1A), these findings may be helpful for utilizing this information for more precise modelling of base embeddings.

Finally, I will illustrate the impact of this dissertation on RNA informatics with a example. Recently, many algorithms for RNA interaction prediction using deep learning have been developed. These methods use experimentally discovered RNA-protein interaction data or RNA-compound interaction data as training data. Although many RNA-protein interactions have been discovered through genome-wide exploration using CLIP-seq, the number is still small. On the other hand, complex functions that have been developed in the field of deep learning require a large amount of training data to avoid overfitting. MXfold and RNABERT can play a role to compensate for such a lack of training data. Since RNA-protein interactions are related to the structure of each molecule, accurate secondary structure prediction by MXfold can strongly assist in predicting the interactions. In the case of RNABERT, base embedding obtained by pre-training using RNA sequences that do not exist in CLIP-seq data can be applied to interaction prediction. iDeepS is an algorithm for predicting protein binding sites in RNA sequences (Pan *et al.*, 2018). In iDeepS, protein binding sites are predicted using a CNN with the one-hot representation of the RNA sequence as input. The one-hot representation of the RNA sequence can be replaced by base embedding using RNABERT. Compared to the simple one-hot representation of RNA sequences, base embedding is richer in structural and contextual information. Therefore, simply adding pre-trained RNABERT to the input of iDeepS is expected to improve the accuracy of interaction prediction. Thus, the contribution of this dissertation to RNA informatics is based on its high versatility to other fields, including RNA interaction. Since vectorization of RNA sequences can be the first step in various areas of RNA informatics, RNABERT will enhance the quality of many studies.

Acknowledgment

First of all, I would like to thank Professor Yasubumi Sakakibara for his support of my research. He has contributed significantly to the promotion of this research and the writing of this dissertation. He was my guide, especially in terms of how to do influential research. Whenever I got stuck in my research, his advice gave me a new breakthrough.

I also would like to thank Full-time lecturer Kengo Sato who has provided a number of technical advices and support for my study as a supervisor of my bachelor, master courses. His advice was accurate and useful in carrying out the study. He also helped me a lot to gain knowledge for machine learning and bioinformatics.

I am very grateful to my colleagues in Sakakibara Laboratory, for making a good academic environment. I would like to thank Dr. Yoshimasa Aoto, Dr. Vasanthan Jayakumar for giving me critical comments on my study as the senior members in the laboratory

Lastly, I would like to express my sincere thanks to Professor Yasubumi Sakakibara, Professor Kotaro Oka, Professor Nobuhide Doi, and Associate Professor Akira Funahashi for examining and judging my doctoral dissertation.

References

Akiba,T. *et al.* (2019) Optuna: A Next-generation Hyperparameter Optimization Framework. In, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Association for Computing Machinery, New York, NY, USA, 2623–2631.

Akiyama,M. *et al.* (2018) A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model. *J. Bioinform. Comput. Biol.*, **16**, 1840025.

Alley,E.C. *et al.* (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods*, **16**, 1315–1322.

Andronescu,M. *et al.* (2010) Computational approaches for RNA energy parameter estimation. *RNA*, **16**, 2304–2318.

Andronescu,M. *et al.* (2007) Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, **23**, i19-28.

Aoki,G. and Sakakibara,Y. (2018) Convolutional neural networks for classification of alignments of non-coding RNA sequences. *Bioinformatics*, **34**, i237–i244.

Asgari,E. *et al.* (2019) Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (DiMotif) and sequence embedding (ProtVecX). *Sci. Rep.*, **9**, 1–11.

Backofen,R. *et al.* (2011) Sparse RNA folding: Time and space efficient algorithms. *J. Discrete Algorithms* , **9**, 12–31.

Baek,J. *et al.* (2018) LncRNAnet: long non-coding RNA identification using deep learning. *Bioinformatics*, **34**, 3889–3897.

Balakrishnan,M. *et al.* (2001) The kissing hairpin sequence promotes recombination within the HIV-I 5' leader region. *J. Biol. Chem.*, **276**, 36482–36492.

Bepler,T. and Berger,B. (2019) Learning protein sequence embeddings using information from structure. In, *International Conference on Learning Representations*.

Bushati,N. and Cohen,S.M. (2007) microRNA functions. *Annu. Rev. Cell Dev. Biol.*, **23**, 175–205.

Carvalho,L.E. and Lawrence,C.E. (2008) Centroid estimation in discrete high-dimensional spaces with applications in biology. *Proc. Natl. Acad. Sci. U. S. A.*, **105**, 3209–3214.

Chang,T.-C. and Mendell,J.T. (2007) microRNAs in vertebrate physiology and human disease. *Annu. Rev. Genomics Hum. Genet.*, **8**, 215–239.

Chen,C.-C. *et al.* (2019) TOPAS: network-based structural alignment of RNA sequences. *Bioinformatics*, **35**, 2941–2948.

Devlin,J. *et al.* (2019) BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*., 4171–4186.

Ding,Y. *et al.* (2005) RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, **11**, 1157–1166.

Do,C.B. *et al.* (2008) A max-margin model for efficient simultaneous alignment and folding of RNA sequences. *Bioinformatics*, **24**, i68-76.

Do,C.B. *et al.* (2006) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**, e90-8.

Dowell,R.D. and Eddy,S.R. (2004) Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, **5**, 1–14.
Durbin,R. *et al.* (1998) Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids Cambridge University Press.
Eddy,S.R. and Durbin,R. (1994) RNA sequence analysis using covariance models. *Nucleic Acids Res.*, **22**, 2079–2088.
Edgar,R.C. (2004) MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, **5**, 1–19.
Flanagan,J.M. and Wild,L. (2007) An epigenetic role for noncoding RNAs and intragenic DNA methylation. *Genome Biol.*, **8**, 1–3.
Fu,Y. *et al.* (2014) Dynalign II: common secondary structure prediction for RNA homologs with domain insertions. *Nucleic Acids Res.*, **42**, 13939–13948.
Ganot,P. *et al.* (1997) The family of box ACA small nucleolar RNAs is defined by an evolutionarily conserved secondary structure and ubiquitous sequence elements essential for RNA accumulation. *Genes Dev.*, **11**, 941–956.
Gardner,P.P. *et al.* (2010) Rfam: Wikipedia, clans and the “decimal” release. *Nucleic Acids Res.*, **39**, D141–D145.
Hamada,M. *et al.* (2009) Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, **25**, 465–473.
Harmanci,A.O. *et al.* (2008) PARTS: probabilistic alignment for RNA joinT secondary structure prediction. *Nucleic Acids Res.*, **36**, 2406–2417.
Heinzinger,M. *et al.* (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 1–17.
Hendrix,D.K. *et al.* (2005) RNA structural motifs: building blocks of a modular biomolecule. *Q. Rev. Biophys.*, **38**, 221–243.
Heyne,S. *et al.* (2012) GraphClust: alignment-free structural clustering of local RNA secondary structures. *Bioinformatics*, **28**, i224-32.
Hirose T. and Tomari Y. (2016) ノンコーディングRNA: RNA分子の全体像を俯瞰する 化学同人.
Hofacker,I.L. *et al.* (2004) Alignment of RNA base pairing probability matrices. *Bioinformatics*, **20**, 2222–2227.
Howe,J.A. *et al.* (2015) Selective small-molecule inhibition of an RNA structural element. *Nature*, **526**, 672–677.
Kalvari,I. *et al.* (2018) Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res.*, **46**, D335–D342.
Kato,Y. *et al.* (2010) RactIP: fast and accurate prediction of RNA-RNA interaction using integer programming. *Bioinformatics*, **26**, i460-6.
Katoh,K. and Standley,D.M. (2013) MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, **30**, 772–780.
Kimura,M. (1980) A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**, 111–120.
Knudsen,B. and Hein,J. (1999) RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. *Bioinformatics*, **15**, 446–454.
Lalwani,S. *et al.* (2014) Sequence-Structure Alignment Techniques for RNA: A Comprehensive Survey. *Advances in Life Sciences*, **4**, 21–35.
Laslett,D. and Canback,B. (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.*, **32**, 11–16.
Lodish H. *et al.* (2019) 分子細胞生物学 東京化学同人.

Lorenz,R. *et al.* (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**, 1–14.
Lu,Z.J. *et al.* (2009) Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, **15**, 1805–1813.
van der Maaten,L. and Hinton,G. (2008) Visualizing Data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.
Mayr,F. and Heinemann,U. (2013) Mechanisms of Lin28-Mediated miRNA and mRNA Regulation—A Structural and Functional Perspective. *Int. J. Mol. Sci.*, **14**, 16532–16553.
McCaskill,J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.
Mikolov,T. *et al.* (2013) Distributed Representations of Words and Phrases and their Compositionality. In, Burges,C.J.C. *et al.* (eds), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 3111–3119.
Min,S. *et al.* (2021) Pre-Training of Deep Bidirectional Protein Sequence Representations With Structural Information. *IEEE Access*, **9**, 123912–123926.
Moore,K.S. and 't Hoen,P.A.C. (2019) Computational approaches for the analysis of RNA-protein interactions: A primer for biologists. *J. Biol. Chem.*, **294**, 1–9.
Morita,K. *et al.* (2009) Genome-wide searching with base-pairing kernel functions for noncoding RNAs: computational and expression analysis of snoRNA families in Caenorhabditis elegans. *Nucleic Acids Res.*, **37**, 999–1009.
Nakamura Y. (2003) RNAがわかる: 多彩な生命現象を司るRNAの機能からRNAi,創薬への応用まで 羊土社.
Nakamura Y. and Siomi H. (2004) 躍進するRNA研究: 進展する構造解析, 機能性RNAの多彩な役割の解明とRNAiなど生命・医工学への応用 羊土社.
Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.
Ng,P. (2017) dna2vec: Consistent vector representations of variable-length k-mers. *arXiv [q-bio.QM]*.
Nussinov,R. and Jacobson,A.B. (1980) Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. U. S. A.*, **77**, 6309–6313.
Pan,X. *et al.* (2018) Prediction of RNA-protein sequence and structure binding preferences using deep convolutional and recurrent neural networks. *BMC Genomics*, **19**, 1–11.
Pennington,J. *et al.* (2014) Glove: Global vectors for word representation. In, *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*., 1532–1543.
Peters,M. *et al.* (2018) Deep Contextualized Word Representations. In, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237.
Reuter,J.S. and Mathews,D.H. (2010) RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**, 1–9.
Rivas,E. *et al.* (2012) A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA*, **18**, 193–212.
Rivas,E. (2013) The four ingredients of single-sequence RNA secondary structure prediction. A unifying perspective. *RNA Biol.*, **10**, 1185–1196.
Rives,A. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U. S. A.*, **118**,

e2016239118.
Saito,Y. *et al.* (2011) Fast and accurate clustering of noncoding RNAs using ensembles of sequence alignments and secondary structures. *BMC Bioinformatics*, **12**, 11–14.
Sakakibara,Y. *et al.* (1994) Stochastic context-free grammers for tRNA modeling. *Nucleic Acids Research*, **22**, 5112–5120.
Samarsky,D.A. *et al.* (1998) The snoRNA box C/D motif directs nucleolar targeting and also couples snoRNA synthesis and localization. *EMBO J.*, **17**, 3747–3757.
Sankoff,D. (1985) Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems. *SIAM J. Appl. Math.*, **45**, 810–825.
Sato,K. *et al.* (2010) A non-parametric Bayesian approach for predicting RNA secondary structures. *J. Bioinform. Comput. Biol.*, **08**, 727–742.
Sato,K. *et al.* (2009) CENTROIDFOLD: a web server for RNA secondary structure prediction. *Nucleic Acids Res.*, **37**, W277-80.
Sato,K. *et al.* (2012) DAFS: simultaneous aligning and folding of RNA sequences via dual decomposition. *Bioinformatics*, **28**, 3218–3224.
Sato,K. *et al.* (2008) Directed acyclic graph kernels for structural RNA analysis. *BMC Bioinformatics*, **9**, 1–12.
Sato,K. *et al.* (2011) IPknot: Fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. *Bioinformatics*, **27**, i85–i93.
Sato,K. *et al.* (2021) RNA secondary structure prediction using deep learning with thermodynamic integration. *Nat. Commun.*, **12**, 1–9.
Schroeder,S.J. and Turner,D.H. (2009) Optical melting measurements of nucleic acid thermodynamics. *Methods Enzymol.*, **468**, 371–387.
Serganov,A. and Nudler,E. (2013) A decade of riboswitches. *Cell*, **152**, 17–24.
Sundfeld,D. *et al.* (2015) Foldalign 2.5: multithreaded implementation for pairwise structural RNA alignment. *Bioinformatics*, **32**, 1238–1240.
The RNAcentral Consortium *et al.* (2017) RNAcentral: a comprehensive database of non-coding RNA sequences. *Nucleic Acids Res.*, **45**, D128–D134.
Tsochantaridis,I. *et al.* (2005) Large Margin Methods for Structured and Interdependent Output Variables. *J. Mach. Learn. Res.*, **6**, 1453–1484.
Turner,D.H. and Mathews,D.H. (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, **38**, D280-2.
Vaswani,A. *et al.* (2017) Attention is all you need. In, *Advances in neural information processing systems*. papers.nips.cc, 5998–6008.
Will,S. *et al.* (2007) Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, e65.
Will,S. *et al.* (2015) SPARSE: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics. *Bioinformatics*, **31**, 2489–2496.
Wilm,A. *et al.* (2006) An enhanced RNA alignment benchmark for sequence alignment programs. *Algorithms Mol. Biol.*, **1**, 1–11.
Wilm,A. *et al.* (2008) R-Coffee: a method for multiple alignment of non-coding RNA. *Nucleic Acids Res.*, **36**, e52.
Zakov,S. *et al.* (2011) Rich parameterization improves RNA structure prediction. *J. Comput. Biol.*, **18**, 1525–1542.
Zuker,M. (1989) On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52.
Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.

Appendix A – List of publications

Journal papers (related to this dissertation)

1. Manato Akiyama, Kengo Sato, Yasubumi Sakakibara: A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model, *Journal of Bioinformatics and Computational Biology*, 16(6), 1840025 (7 pages) (2018).
2. Manato Akiyama, Yasubumi Sakakibara: Informative RNA base embedding for RNA structural alignment and clustering by deep representation learning, *NAR Genomics and Bioinformatics*, (in press).

Journal papers (others)

1. Kengo Sato, Manato Akiyama Yasubumi Sakakibara: RNA secondary structure prediction using deep learning with thermodynamic integration. *Nature Communications*, 12, 941 (2021).

International conferences

1. Manato Akiyama*, Kengo Sato, Yasubumi Sakakibara: A max-margin training of RNA secondary structure prediction integrated with the thermodynamic model, *International Conference on Genome Informatics (GIW2018)*, Kunming, China, December 3-5, 2018.

Appendix B – Supplementary information of genome analysis

Self-Attention Mechanism

Figure S1 illustrates the single-head case of the self-attention mechanism. The transformer is an encoder-decoder type of feed-forward neural network. The self-attention function for an encoder-decoder neural network is a dot-product attention formulated as

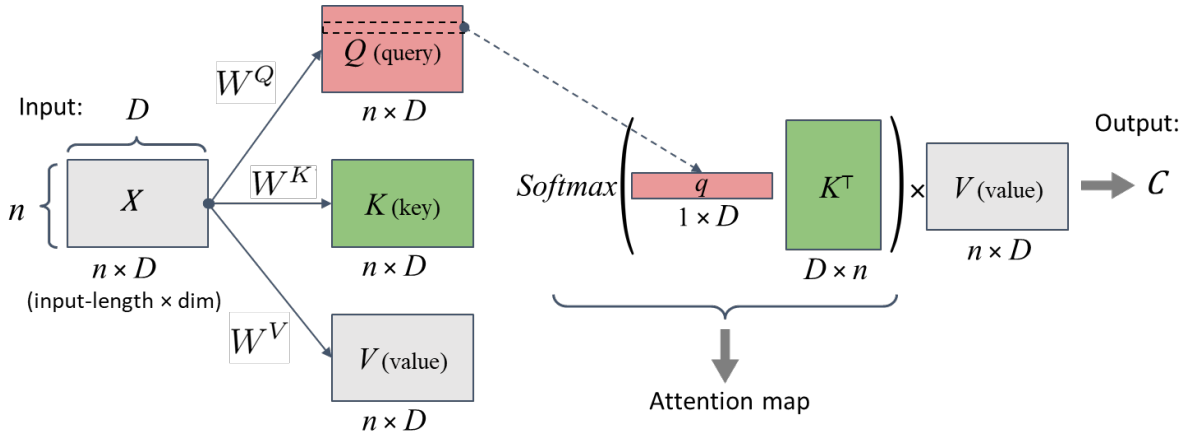
$$attention = softmax(Target \cdot Source^{\top}) \cdot Source\,,$$

where $Souce$ represents the encoder layer and $Target$ represents the decoder layer. The BERT algorithm generalized it by considering $Target$ as (search) query Q and separating $Source$ into key K and value V, formulated as:

$$attention = softmax(Q \cdot K^{\top}) \cdot V\,.$$

In this formulation, the attention function computes an output (attention weight) based on a query (Q) and a set of key-value pairs (K, V). The key-value pairs (K, V) can be considered as a kind of dictionary. By separating $Source$ into key K and value V, the dot-product between query Q and key K plays a role to measure the relevance of the value V for query Q (how much it has an attention). These Q, K and V are calculated by linear projection from the input X with learnable parameters W^Q, W^K and W^V, formulated as:

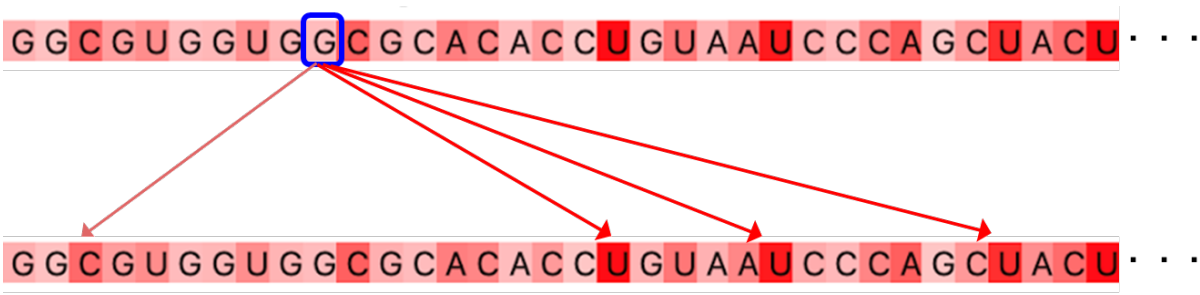
$$Q = XW^Q, K = XW^K, V = XW^V.$$



Supplemental Figure S1. Illustration of the single-head case of the self-attention mechanism.

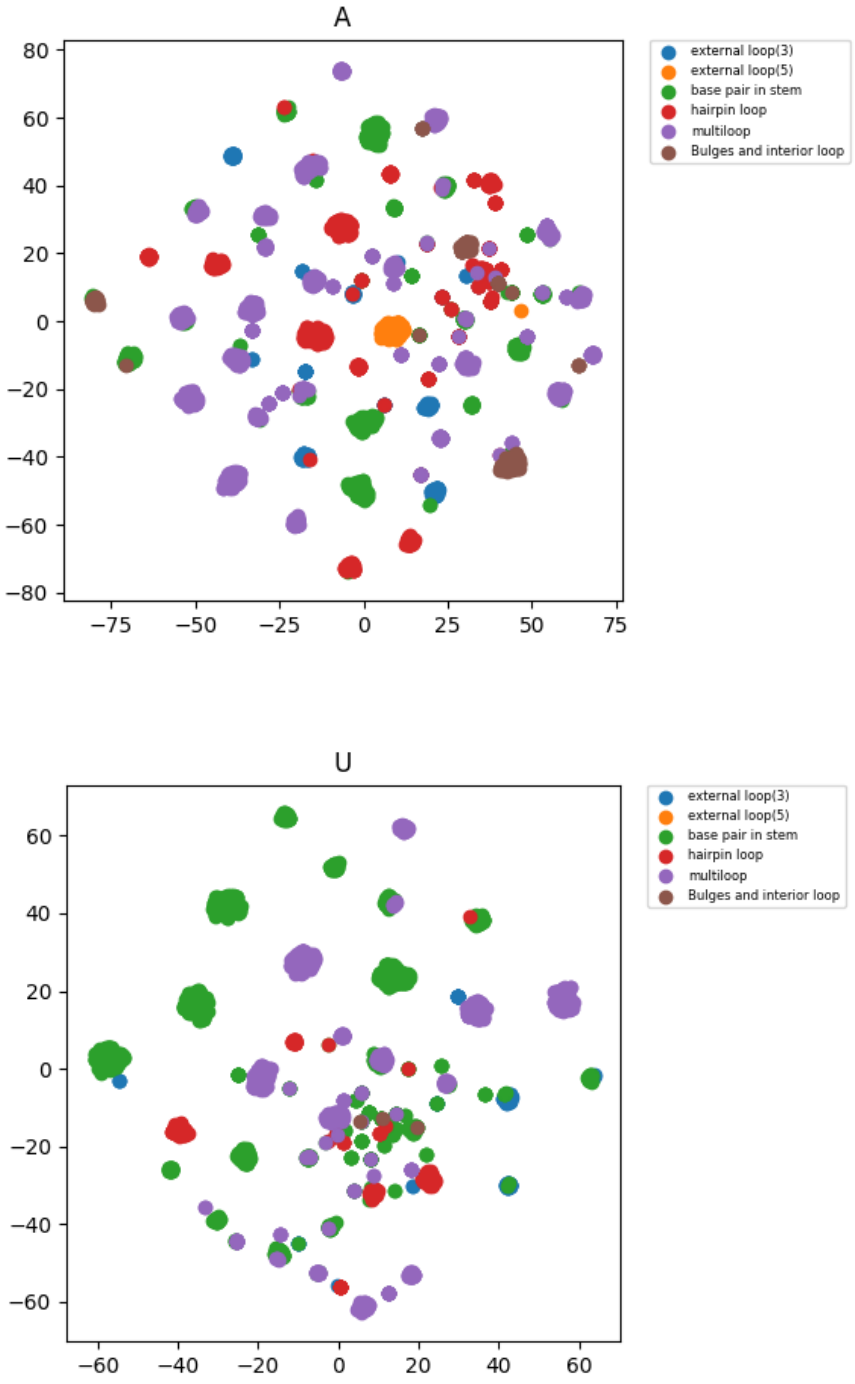
RNA Motif Detection using Self-Attention Map

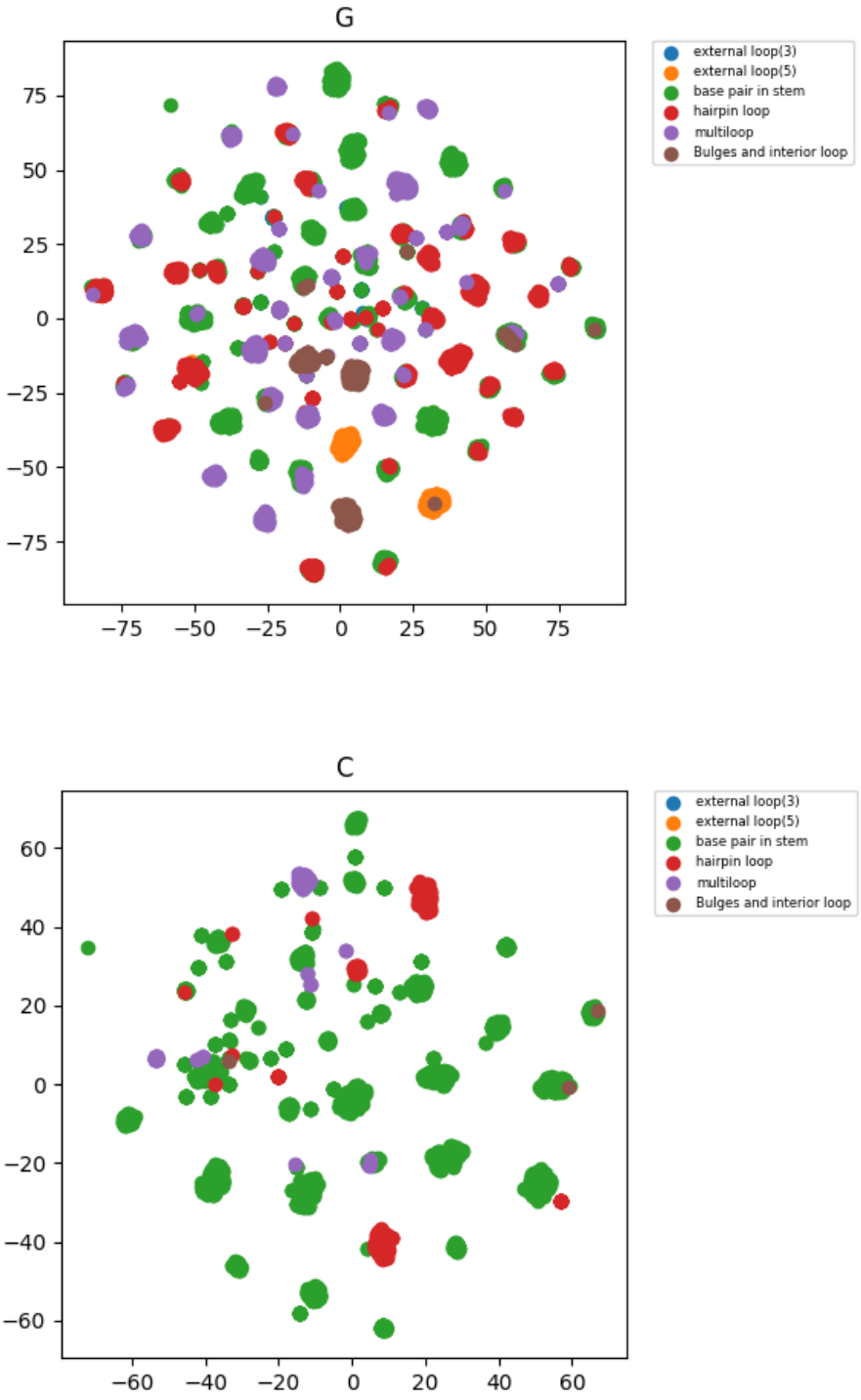
The attention map calculates the inner product between the query vector of each base and the key vector of the other bases in the input RNA sequence, then measures the relevance of the base with the other bases, as illustrated in the supplemental Figure S1. The supplemental Figure S2 shows the strength of the relevance of each base, which is represented by the intensity of red. The sequence below in Figure S2 represents the relevance of the 10th base "G" from the left, which is surrounded by a blue frame. In the Figure S2, arrows are drawn for bases that are particularly relevant to the base "G". The sum of the relevance calculated for each base is finally defined as an attention map. Thus, the attention map is an index showing how much each base contributed to the prediction of the pre-training task. Therefore, in the MLM task, the bases that are important for the prediction of the masked base, and in the SAL task, for the prediction of the structural alignment obtain high values in the attention map. Finally, the bases with high attention values are identified as sequence motif.



Supplemental Figure S2. Example of RNA motif detection using self-attention map

Visualization of base embedding with t-SNE





Supplemental Figure S3. Visualization of embedding of six secondary substructures with t-SNE; hairpin loop, base pair in stem, bulge and internal loop, multibranch loop, external loop at 3', and external loop at 5'. The plot is displayed for each base.

Detail of RNA structural alignment tools

Supplemental Table S1. The list of command, options, package, and link information for each existing method.

Programs	command	Package	URL
LocARNA	locarna fasta_file	LocARNA 1.9.2.1	https://rna.informatik.uni-freiburg.de/LocARNA/Input.jsp
SPARSE	sparse fasta_file	LocARNA 1.9.2.1	http://www.bioinf.uni-freiburg.de/Software/SPARSE/
RAF	raf predict fasta_file	1.0.0	http://contra.stanford.edu/contrafold/raf.html
PARTS	parts configuration_file	RNAstructure version 6.0.1	http://rna.urmc.rochester.edu/RNAstructure.html
Dyalign	dynalign_ii configuration_file	RNAstructure version 6.0.1	http://rna.urmc.rochester.edu/RNAstructure.html
MAFFT	mafft fasta_file	MAFFT version 7	https://mafft.cbrc.jp/alignment/software/
Foldalign	foldalign -global fasta_file	Foldalign version 2.5.0	https://rth.dk/resources/foldalign/
TOPAS	TOPAS[fasta_file, base-pairing and alignment probabilities, default parameters]	TOPAS version 1.3	https://github.com/bjyoontamu/TOPAS
DAFS	dafs fasta_file	0.0.3	https://github.com/satoken/dafs
R-Coffe	t_coffee fasta_file -mode rcoffee	T-COFFEE Version_13.45.0	http://www.tcoffee.org/Projects/rcoffee/index.html