TEL AVIV UNIVERSITY אוניברסיטת תל אביב

Tel-Aviv University
Raymond and Beverly Sackler
Faculty of Exact Sciences
The Blavatnik School of Computer Science

Non-Coding RNA Sequence Alignment by Molecular Sequence and Structure Properties

Thesis submitted in partial fulfillment of the requirements for
M.Sc. degree in the School of Computer Science, Tel-Aviv University

By

Maor Dan

The research work for this thesis has been carried out at Tel-Aviv University
under the supervision of

Prof. Ron Shamir and Dr. Yaron Orenstein

April 2019

ABSTRACT

Non-coding RNAs (ncRNA) play major roles in the cell through their sequence and structure. Identifying functional units within RNA molecules is thus a key challenge. To identify different subsequences of similar function RNA secondary structure analysis can be used in RNA sequence alignment. RNA structure adds information on top of the sequence and allows us to make better alignment and retrieve more significant functional units. Various algorithms for simultaneous alignment and folding of RNA sequences have been developed. These algorithms provide results of variable accuracy that depends on the runtime and most of them are infeasible for large inputs.

We introduce two algorithms: LASSP for local alignment and GASSP for global alignment of ncRNA sequences. The algorithms utilize both sequence and structure information. Both LASSP and GASSP maintain the time complexity of classic sequence alignment algorithms, i.e. they depend quadratically on the input length. They require pre-processing of the data to calculate structural information for the input data. This usually takes more time than the alignment itself, but can be done once in advance for an entire database of RNA sequences in reasonable time. We also extended GASSP to a multiple sequence alignment mode. We show that GASSP significantly outperforms sequence-only alignment tools in alignment quality, while maintaining practical running time. Moreover, the GASSP-generated solution can be used as an initial alignment for the state-of-the-art algorithm LocaRNA to find an optimal alignment and folding in much shorter running times.

ACKNOWLEDGEMENTS

I would like to thanks my advisors, Prof. Ron Shamir and Dr. Yaron Orenstein, for their continued support throughout this way. Their combined knowledge and experience allowed me to dive into a new territory and feel welcome and supported. I also want to thank the members of Ron Shamir's lab with whom I consulted or just had the pleasure of listening to their presentations of fascinating work and research.

1 TABLE OF CONTENTS

1 INTRODUCTION

Sequence alignment is a fundamental problem in computational biology. It is used, for example, for the study of evolution, for comparative genomics, for structural comparison and modeling, for human genetics and for drug design. It allows not only to align a set of sequences but also to model and define what makes one alignment better than another. This model or scoring scheme are a basic means of detecting motifs in biopolymers [1].

Since the function of non-coding RNAs (ncRNA) may depend on structure as well as on sequence, structure may also be conserved through evolution, and structural motifs can be discovered and used for ncRNA detection and classification [2]. This motivates the scientific community to develop sequence and structure-based alignment and motif finding tools.

Structure analysis of RNAs may be very time consuming [3, 4] and speed improvement of accurate structure prediction algorithms is extensively sought after [5]. Faster algorithms for structure prediction tend to be less accurate in general [6]. Therefore, a key challenge is to develop a ncRNA alignment tool with improved accuracy while maintaining a running time that is tractable for genomic scale alignments.

In this work we introduce a local and global sequence and structure alignment algorithms for ncRNA, called LASSP and GASSP, respectively. The two algorithms receive as input two sequences and their one-dimensional vectors of structural information (that will be further explained later) and output an alignment (local or global).

We show, using various benchmarks (some novel and some adopted from previous studies), that LASSP and GASSP improve classic results by using the structural information provided to them. We also demonstrate how GASSP can be used to improve the runtime of a leading alignment tool by narrowing its search space with minimal reduction in accuracy. Lastly, we describe a multiple sequence alignment adaptation of GASSP that is based on progressive alignment.

2 BACKGROUND

2.1 Biological Background

2.1.1 RNA

As with any complex machine, live organisms also use blueprints. The genetic code of an organism fulfills exactly that role. A species' genetic code resides in its Deoxyribonucleic acid (DNA). The process in which this code is read and used involves transcription into RNA. Ribonucleic acids (RNA) are the product of transcription of DNA subsequences of varying lengths. RNA molecules are built from four building blocks: Adenine, Cytosine, Guanine and Uracil, and can be viewed as a sequence over the alphabet $\{A, C, G, U\}$.

RNA molecules have many different roles, many of which may yet be unknown. One role of RNA is to encode proteins. mRNAs (messenger RNA) are RNA molecules that contain recipes for proteins. The protein coding part of an mRNA is made of codons. A codon is an RNA triplet, which encodes for a single amino acid. In a process called translation, the amino acids are assembled in order matching their codons and the result is a chain of amino acids. That chain is folded into the protein that this mRNA held the recipe for.

There are many other types of RNA that are not coding for proteins, but have other roles. Non-coding RNA (ncRNA) is a type of RNA molecules that are not translated to proteins. ncRNAs have many subtypes such as tRNA and rRNA, which have roles in translation of mRNA into proteins. ncRNAs interact with other molecules in the cell, such as proteins, DNA [7] and other RNA molecules. These interactions are mediated through the RNA sequence, the three-dimensional (3D) structure of the RNA molecule, or both. RNA structure may limit the regions in the molecule that are available for interaction and by that is a major factor in determining whether an interaction will occur.

2.1.2 RNA Structure

The RNA structure affects its ability to interact with various molecules. Knowing the 3D structure of an RNA molecule will allow us to deduce which regions in it are more prone to interactions with other molecules based on their accessibility.

The 3D structure of RNA molecules is very complex, but a useful abstraction of the structure is used instead. The secondary structure of an RNA molecule is defined to be a set of base-pairing positions. For an RNA molecule, a pair in the secondary structure, (i, j), implies that the two bases in positions i and j

are *paired* (Figure 2-1). Paired bases in the structure are very close in the three-dimensional space and chemical bonds, called hydrogen bonds, are created between them. These bonds induce a folded structure to a long RNA molecule. The structure enables interactions with other molecules while preventing unwanted interactions. A nucleotide may only pair with one other nucleotide. Pairs can be formed between A and U, between G and C and G and U only.

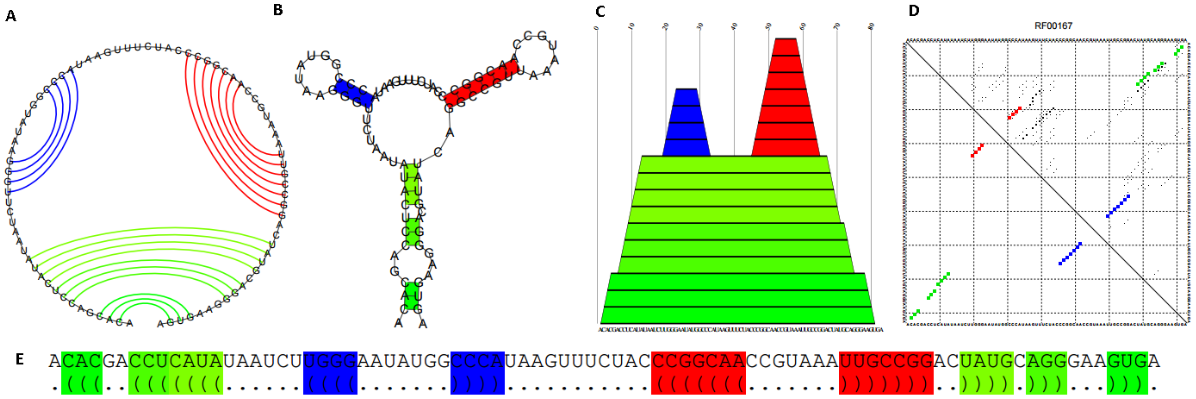

Figure 2-1 Different representations of RNA secondary structure. Figure taken from [8].
A. Circle Plot – the sequence is arranged as a circle and arcs are drawn between paired bases.
B. Conventional visualization of the secondary structure.
C. Mountain plot – each level begins and ends at positions of paired bases starting with the most external pairs.
D. Dot plot – a dot marks an (x,y) position where x and y are positions of paired bases. The dot plot also shows (upper right half) calculated probabilities of base pairing that are not part of the specific secondary structure shown in the example.
E. Bracket notation shows the sequence with an additional line of brackets and dots. The dots represent unpaired positions and every pair of matching brackets (open close) represent a base pair.

Although RNA secondary structure is limited in its ability to describe the actual structure of the molecule, it allows us to identify certain properties of the actual structure. Such properties are adjacent paired bases that form a stem-like structure (red, green and blue colored segments in Figure 2-1), or the existence of loops of single-stranded sequence of bases (non-colored segments in Figure 2-1). Using these properties, we can learn about the function of the RNA molecule.

2.1.3 Computational Prediction of RNA Secondary Structure

Fortunately, *in silico* RNA secondary structure prediction is a relatively tractable problem. Under the secondary structure model, along with a simplifying assumption that prohibits a certain substructure, efficient calculation of RNA secondary structure is possible.

Structure prediction is generally done by building a model of the physical forces acting between particles. Usually a simplified energy-based model is used since it allows quantifying a complex 3D physical problem with a scalar number. For a single sequence, the most common method for predicting a secondary structure is by minimizing a quantity called *free energy*. Amongst algorithms employing free energy minimization, the most popular method is using dynamic programming while considering only tree-like structures. This is done by minimizing free energy of sub-sequences and finding the best global structure by combining these sub-structures and optimizing the total free energy [9].

Using these models, it is also possible to estimate the probability that two specific bases in the sequence would be paired in the secondary structure. This can be accomplished, for example, by computing the local secondary structure of all subsequences of a certain length containing the two bases in question. The total probability of an RNA to reside in structures in which the pair is a structural base pair is the probability of this base pair to occur in the global secondary structure [10].

Pseudoknots*:* An RNA structure for sequence S of length n can be represented by a set of pairs (i,j) showing the paired positions in S. A pseudoknot in S is two pairs $(i,j) \in S$ and $(k,l) \in S$ that are overlapping, namely the intervals $i..j$ and $k..l$ are neither disjoint nor one is contained in the other.

The common assumption in structure prediction is that pseudoknots are not allowed in the secondary structure. When pseudoknots are allowed, the problem of RNA secondary structure prediction, e.g. finding the minimum free energy structure, becomes NP-hard [11]. The operation of calculating the structure of an RNA molecule is termed the *folding* of the molecule. Basic substructures that may be induced by RNA secondary structures are stems, loops and bulges, but more complex structures, such as multi-loops are also possible (definition of multi-loops can be found at [8]).

2.1.4 RNA-Protein Interactions

One of the molecules with which ncRNAs interact are proteins. Proteins that chemically bind RNAs are called RNA-binding proteins (RBPs). Each RBP binds different RNA molecules. Both proteins and RNA molecules have linear chemical structure (comprised of a chain of sub-elements). Protein-RNA binding occurs at specific positions of the RNA and protein molecules. RBPs usually distinguish specific sequences as binding sites, in which case we say that the protein has a sequence preference. The binding preference can also be structural. Most RBPs are known to bind to single-stranded RNA sequences, while few are known to interact with paired RNA nucleotides[12].

2.1.5 Experimental Methods for Measuring Protein-RNA binding

CLIP experiments

CLIP experiments measure protein-RNA binding *in vivo* in a high-throughput manner. The experimental protocol consists of a few stages. First a tissue, an organism or a cell are irradiated with UV-B radiation. The radiation forms covalent bonds between RNA atoms and protein atoms that are in close proximity. In an enzyme cleaving process, the RNA molecules are shortened to around 40nt long. Using a protein-specific antibody, the copies of a specific protein together with the shortened bound RNA are purified. When sufficient purification is achieved, the complex of RNA and protein is dissolved resulting in a large set of short RNA molecules that were all bound by the tested protein [13]. Sequencing and mapping the resulting RNA yields a map of locations where these RNA sequences are most probably located on the organism's reference genome. In a process of peak calling, locations in the genome with sufficient RNA sequences mapped to them are identified. These are considered as sequences that were actually bound by the protein.

RNAcompete

The RNAcompete technology measures protein-RNA binding *in vitro* in high-throughput. In each experiment a pool of synthetic 29-38nt long RNA sequences is generated. These sequences are designed such that every possible RNA 9-mer appears as a subsequence at least 16 times. The target RNA binding protein (RBP) is incubated in the RNA pool. After isolating the proteins with bound RNA from the rest of the pool, the relative occurrence of each RNA sequence is measured using microarray hybridization. The ratio between this measurement in the array with the RBP and in the original pool provides an estimate of the protein binding intensity to each sequence [14]. The output of one RNAcompete experiment is a list of binding intensities of a specific RBP to more than 240,000 sequences. Post-processing of these data gives a Z-score for the binding of every possible RNA 7-mer by the RBP.

Currently, RNAcompete data comprises of experiments done on 205 different RBPs in 244 different experiments. These RBPs include 85 RBPs from human, 61 from Drosophila and 61 RBPs from 18 other species [15].

2.2 Computational Background

2.2.1 Formal Notations and Definitions

Alignment related definitions:

Alignment – an alignment of two sequences, $S = (S_1 \dots S_n)$ and $T = (T_1 \dots T_m)$ is a set of pairs $(i, j) \;\; 1 \leq i \leq n, 1 \leq j \leq m$ specifying aligned bases. This set defines a unique way to align the two sequences allowing character replacement and insertions/deletions. In the textual representation of a pairwise sequence alignment, dashes (also called spaces) are used to denote gaps (a missing base or an inserted base in the other sequence).

Example:

```
AC-GTTAC--TAG
AAAGT-AGGGTAG
```

This alignment is represented as the set

$$A \ = \{ \ (1,1), (2,2), (3,4), (4,5), (6,6), (7,7), (8,10), (9,11), (10,12)\}$$

Note that pairs cannot cross: $(i, j), (k, l)$ with $i < k$ and $j > l$ is impossible, and that no column in the textual representation can contain two dashes.

Multiple sequence alignment (MSA) – an MSA is an extension of pairwise alignment to multiple sequences.

An MSA of a set of N sequences, $S_1, S_2, \dots, S_N$ is another set of N sequences $S'_1, S'_2, \dots, S'_N$ such that $|S'_1| = |S'_2| = \cdots = |S'_N|$ and S'_i is the sequence S_i with spaces inserted at 0 or more positions.

We represent MSA textually. Each sequence's characters are put in a separate line so that characters aligned together in the MSA are all in one column. Gaps are represented by dashes.

Example:

```
AC-GTTAC--TAG
AAAGT-AGGGTAG
AA-GTTAGGC-AG
```

Hierarchical clustering and MSA – a clustering of a set of sequences (or items) is a partition of the set of sequences into subsets called *clusters***.** The partition can be done in different ways using different clustering methods. A hierarchical clustering is a data structure that describes how the initial set is recursively split into subsets until single items (1-sized clusters) remain. The process could also be done in the opposite direction, starting with single items and joining subsets to larger and larger clusters until achieving the original set. Such hierarchy does not specify clusters per se, but can be used to cluster the data quickly by various properties or constraints (such as the number of clusters wanted, or the maximal distance between items in the cluster).

Hierarchical clustering can be used to perform MSA heuristically very fast. Instead of finding an optimal MSA (which is an NP-Hard problem [16]), we can start from individual sequences as clusters and repeatedly align the best pair of clusters. Each alignment forms a new MSA. Each time two clusters are aligned a single cluster replaces them and can be aligned with other clusters. This method can align N sequences of length m in $O(N(N-1)m^2)$ time [17].

Notations

Sequence notations – Throughout the thesis, while discussing an alignment of two sequences:

$\boldsymbol{S}$ marks the first sequence

$\boldsymbol{T}$ marks the second sequence

$\boldsymbol{n}$ marks the length of S

$\boldsymbol{m}$ marks the length of T

$S[i]$ denotes the base at position i in the sequence S.

$S[i:j]$ denotes the subsequence of S from position i to j (inclusive).

RNA structure

Secondary Structure –

A secondary structure of a sequence S, is a set F, of pairs (i,j) specifying paired bases in the molecule structure. The pair (i,j) in F, indicates the pairing of $S[i]$ with $S[j]$.

Example:

The secondary structure in Figure 2-1 is represented as:

$$F = \begin{Bmatrix} (2{,}82),(3{,}81),(4{,}80),(6{,}76),(7{,}75),(8{,}74),(9{,}72), \\ (10{,}71),(11{,}70),(12{,}69),(19{,}34),(20{,}33),(21{,}32),(22{,}31), \\ (45{,}66),(46{,}65),(47{,}64),(48{,}63),(49{,}62),(50{,}61),(51{,}60) \end{Bmatrix}$$

Base pairing probabilities – For a given sequence S, the base pairing probabilities are, for every pair (i,j) of positions in the sequence, the probability that the bases $S[i]$ and $S[j]$ are paired in the RNA secondary structure. Note that by definition the probability is symmetric.

Example:

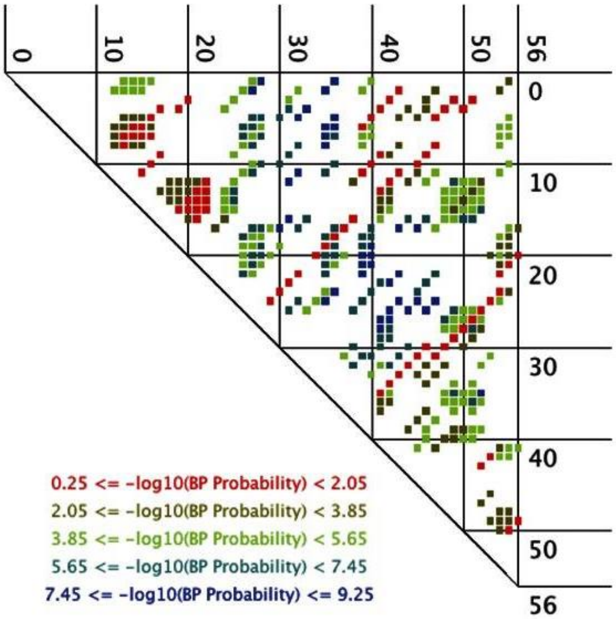

Figure 2-2 A color coded base pairs probabilities matrix. [35]

2.2.2 Sequence Alignment

Perhaps the most basic and important computational tool in the field of RNA and DNA sequences is *sequence alignment*. The purpose of sequence alignment is simple: Given two (or more) sequences over some alphabet (representing DNA, RNA, or even amino acids), find an optimal alignment. For example, maximize the number of matching characters across the sequences while minimizing the number of gaps and mis-matching characters. The objective function is chosen to best fit the biological model describing the relation between the sequences. For example, considering two DNA sequences from two similar species sharing a common ancestor, gaps in the alignment represent insertions or deletions (*indels*), i.e. some DNA bases disappeared over the generations, or a nucleotide was inserted into one of the sequences. A substitution refers to replacement of one nucleotide by another, represented as a mismatch in the alignment. In that case, the objective function of the alignment would express how likely a substitution and indel is. If indels are less probable than substitutions of existing nucleotides, the objective function would penalize for indels more than for substitutions.

The concept of sequence alignment was vastly extended over the years [18]. Among the common alignment methods used today are:

- *Global alignment* - refers to an alignment between two whole sequences
- *Local alignment* - used to find the best aligned subsequences of two sequences
- *Multiple sequence alignment* of three or more sequences

Of those alignment problems, optimal global and local alignment of two sequences can be calculated in $O(mn)$ time and $O(m+n)$ space. The multiple sequence alignment problem is NP-hard [16] and thus can only be solved heuristically when the number of sequences is large.

2.2.3 RNA Folding

Computational prediction of RNA structure is cheaper and easier to obtain than laborious experiments aimed to measure the molecular structure.

RNA structure can be efficiently predicted *in silico*. Prediction of RNA secondary structure (RNA folding) is done computationally through various methods under different simplifying assumptions. The number of possible secondary structures is exponential in the sequence length, but polynomial time dynamic programming algorithms can find an optimal structure. Excluding pseudo-knots allows for such algorithms to benefit from the recursive nature of the non-looped secondary structures. If pseudo-knots are prohibited, every subsequence can be folded regardless of the rest of the sequence. A plethora of algorithms were developed for calculating the minimum free energy structure, and ensemble of representative structures. It is also possible to calculate a vector of pairing probabilities, where each nucleotide is assigned the probability of being paired. Some of these tools will be discussed in the next chapter.

2.2.4 Combined alignment and folding

Combining RNA secondary structure prediction with RNA sequence in alignment helps in finding similarity between RNA molecules that have only limited sequence similarity. Some families of ncRNA have more noticeable structural features in common, while not much is common in terms of subsequences. Using structural information in combination with sequence data provides more accurate alignments and allows discovery of novel types of RNA families and regulatory elements in them [4].

2.2.5 Algorithms and Tools

This subsection presents the key principles applied in algorithms for aligning and folding of RNA sequences, with specific emphasis on the tools used and compared to in this thesis.

Sequence Alignment Algorithms and Tools

The standard method for performing pairwise sequence alignment is through dynamic programing. *Global alignment* can be performed using the Needleman-Wunch algorithm [19] in quadratic running time and linear space (using a modification proposed by Hirschberg [20]). *Local alignment* can be computed using the Smith-Waterman algorithm [21] and with similar Hirschberg's modification in quadratic time and linear space. Those methods were improved and extended over the years to allow applications such as multiple sequences alignment or identification of short repeating motifs in one long sequence. Running any of those algorithms to align two sequences takes little time. Optimally aligning multiple sequences or sequence profiles however is intractable since MSA is NP-Hard under the conventional scoring schemes [16]. *Needle* [22] is a popular implementation of the Needleman-Wunch algorithm for global alignment and is used in this thesis for comparison. There are other alignment tools available such as the popular ClustalW and BlastZ (for a comparison of some of the more popular alignment tools, see [23]).

UPGMA

***U**nweighted **P**air **G**roup **M**ethod with **A**rithmetic mean* [24] is an algorithm for generating a hierarchical clustering of a set of samples given a pairwise similarity matrix.

Starting with the original N samples, at each iteration a new cluster is created by joining the two most similar existing clusters and discarding the two source clusters. A single sample is considered a cluster of size 1.

When a new cluster is created, the similarity matrix is also extended to measure its similarity to all the existing clusters. In UPGMA, when joining two clusters A and B, the similarity of all other clusters and the new cluster C is calculated by the following formula:

$$For\ some\ existing\ cluster\ x:$$

$$d_{C,x} = \frac{|A| \cdot d_{A,x} + |B| \cdot d_{B,x}}{|A| + |B|}$$

Where $d_{A,x}$ and $d_{B,x}$, the similarity of clusters A and B to x, respectively, were calculated the same way and are already in the matrix. For clusters of size 1, d is the similarity of the appropriate samples.

The algorithm terminates when there is only one cluster left.

The output is a binary tree structure where the original samples are its leaves and each internal node is a cluster of two or more samples. The tree's root is a cluster containing all the samples.

Sankoff's Algorithm for RNA sequence and structure pairwise alignment

The first method to find optimal simultaneous alignment and folding of RNA is Sankoff's algorithm [3]. The algorithm is based on dynamic programing whose objective function aims to maximize both sequence similarity and predicted secondary structure of the input sequences. The complexity of the algorithm is $O(n^3m^3)$.

The score used in the target function is based on chemical principles and sequence similarity. A quantity called *free energy* is used to score a structure of RNA (based on structure stability) and together with sequential differences a combined score is calculated. We do not describe the full method here in order to avoid providing the chemistry background needed for it. We shall describe instead a simplified version.

PMComp

One simplified variant of Sankoff's algorithm is called *PMComp* [25] and is using McCaskill's algorithm [26] to score different secondary structures Instead of using a scoring system that favors thermodynamically stable structures.

McCaskill's algorithm produces a matrix of pairing probabilities for every pair in a sequence. Those probabilities are then evaluated into a scoring matrix that is used by PMComp.

PMComp uses the following formulas for finding the best alignment and folding [27]:

For sequence A, define

$$\Psi_{\mathrm{ij}}^{\mathrm{A}} = \log \frac{P_{ij}^{A}}{p_{min}}$$

Where P_{ij} is the probability for bases in positions i and j to be paired, and p_{min} is the minimal probability for a pairing that is deemed significant. Ψ_{ij}^{A} is the score the pairing of $A[i]$ and $A[j]$ in a secondary structure.

Recall that $S[i{:}j]$ is the substring $S_i, S_{i+1}, \dots S_j$. The algorithm computes recursively $M_{ij;kl}$, the optimal score of the sub-alignments $S[i{:}j]$ and $T[k,l]$ as follows:

$$M_{ij;kl} = \max \begin{cases} M_{ij-1;kl-1} + \sigma(S_j, T_l) \\ M_{ij-1;kl} + \gamma \\ M_{ij;kl-1} + \gamma \\ \max\limits_{j'l'} \left\{ M_{ij'-1;kl'-1} + D_{j'j;l'l} \middle| \begin{matrix} i \le j' < j \\ k \le l' < l \end{matrix} \right\} \end{cases} \quad \forall \begin{matrix} 1 \le i < j \le n \\ 1 \le k < l \le n \end{matrix}$$

σ is the sequence alignment score for the two given bases and γ is the gap penalty.

$$D_{ij;kl} = M_{ij-1;kl-1} + \Psi_{ij}^{S} + \Psi_{kl}^{T} + \tau\,(S_i, S_j\,; T_k, T_l)$$

$\tau(S_i, S_j\,; T_k, T_l)$ is the match score for a base pair $ij; kl$. Defining τ separately allows us to score the matching of bases differently when those are structurally paired and unpaired.

$D_{ij;kl}$ is an auxiliary matrix used to evaluate matrix M. It contains the scores of the respective sub-alignment with the addition of the structural and sequential score for the outermost base pair. To populate the entire matrices D and M, four-dimensional matrices, O(n^2m^2) cell calculations are required.

Each cell is calculated after examining $O(nm)$ previous values of M and D. This amounts to a time complexity of $O(n^3m^3)$ and requires $O(n^2m^2)$ space.

The final result is computed at $M_{1n;1m}$ and standard backtracking can be used to derive the underlying alignment.

LocaRNA

LocaRNA (*Local Alignment of RNA)* [4] is a tool for simultaneous alignment and folding of RNA sequences implementing a Sankoff-style algorithm. LocaRNA is based on a dynamic programing as a mean of finding an optimal solution, but introduces a new assumption allowing it to ignore some possible structures a priori. Eliminating most structures before trying to find the optimal solution enables the algorithm to decrease its running time by a factor of $O(nm)$ compared to the original Sankoff's algorithm.

Just as with PMComp, Prior to dynamic programming, pairwise base-pairing probabilities are calculated for each sequence. LocaRNA disregards any pairing with probability below a given threshold, making the number of possible structures much smaller. The downside is that LocaRNA is a heuristic and it may exclude the optimal solution.

LocaRNA uses the following formulas for finding the best alignment and folding:

For sequence A, define

$$\Psi_{ij}^{\mathrm{A}} = \begin{cases} \dfrac{\log \dfrac{P_{ij}}{p_0}}{\log \dfrac{1}{p_0}} & P_{ij} \geq p^* \\ -\infty & else \end{cases}$$

Where p^* is a minimal pairing probability threshold, P_{ij} is the probability for bases in positions i and j to be paired, and p_0 is the expected probability for a pairing to occur at random. Ψ_{ij}^{A} is a scoring matrix used for scoring the pairing of $A[i]$ and $A[j]$ in a secondary structure.

Using a very similar recursion formula, but extended to allow local alignments, LocaRNA defines:

$$M_{ij;kl} = \max \begin{cases} M_{ij-1;kl-1} + \sigma(S_j, T_l) \\ M_{ij-1;kl} + \gamma \\ M_{ij;kl-1} + \gamma \\ \max\limits_{j'l'}\{M_{ij'-1;kl'-1} + D_{j'j;l'l}\} \end{cases} \quad i > 0 \; or \; k > 0$$

σ is the sequence alignment score for the two given bases and γ is the gap penalty.

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

$M_{0j;0l}$ is a special slice of matrix M, as 0 is the minimal value of any cell in it.

$$D_{ij;kl} = M_{ij-1;kl-1} + \Psi_{ij}^{S} + \Psi_{lk}^{T}$$

$D_{ij;kl}$ is an auxiliary matrix used to evaluate matrix M. It contains the scores of the respective sub-alignment with the addition of the structural score for the last base pair.

$\max_{jl} M_{0j;0l}$ is the optimal local alignment.

To populate the entire matrix D, a four-dimensional matrix, O(n^2m^2) cell calculations are required. At a first glance it seems that each cell calculation itself requires testing of $O(nm)$ entries from M and D. However, due to the assumption of sparsity of D it can be shown that over the entire calculation of M, no more than $O(n^2m^2)$ entries of M are tested. As a result, LocaRNA calculates a near-optimal alignment while reducing the total time required by a factor of $O(nm)$.

There is a hidden assumption in the previous statement that p^* is not too small. For small p^* the time complexity of one matrix cell calculation can be $O(nm)$ on average, rendering the entire LocaRNA model useless. The reason that p^* is omitted from the time complexity calculation is the assumption that it will not be dependent on the input length in any way.

Unfortunately, LocaRNA's running time is still too long to be useful for large inputs. For a single alignment of two 100nt long RNA sequences, it could take half a second for a system dedicating one core of 3.3GHz CPU to the task and 8 seconds for 200nt long sequences. For comparison, the same system can calculate a classical global alignment of the same 100nt long sequences in 0.05 seconds. It takes marginally the same time for 200nt long sequences as most of this time is overhead. This ratio, of course, gets worse for longer sequences and also become significant when trying to perform an MSA, which requires performing all pairwise alignments ($O(N^2)$), where N is the number of sequences.

One way for LocaRNA to shorten running time is to rely on some reference alignment. Limiting the algorithm to test only alignments (or sub-alignments) that are close enough to the reference is known as 'banding' in dynamic programming [28]. Doing so rules out most cases LocaRNA would usually test, thus cutting down running time profoundly, but may produce less accurate results.

SPARSE

SPARSE or "*sparsified prediction and alignment of RNAs based on their structure ensembles"* [5] is another Sankoff-style algorithm for alignment and folding or RNA sequences. SPARSE uses a different energy model than LocaRNA and introduces a stronger sparsity assumption. SPRASE assumes that most structural constellations have a probability that is lower than some fixed threshold. (In fact, it uses three probability thresholds.) This assumption leads to a running time complexity of $O(nm)$. However, this complexity actually depends on the choice of the thresholds and choosing them to be very small may lead to a very long running times in practice. This may explain why even though the running time appears to be faster by a quadratic factor compared to LocaRNA, it only cuts the practical running time by a small factor (less than 4) [5].

BEAR

BEAR or "*Brand nEw Alphabet for RNA*" [6] attempts to apply the concept of sequence alignment to the matching of two structures. BEAR aligns two RNA sequences over the "BEAR Alphabet", which represents the structural elements in the various positions.

Using the input sequence, an optimal secondary structure is first calculated for each sequence. This structure is then translated to the new alphabet. Using the two sequences written in the new alphabet, Needleman-Wunsch algorithm can be used almost directly to find an optimal alignment. The different structural modifications within known RNA families were used to build the transition matrix for scoring the alignment. On top of the structural score of alignment, the corresponding regular sequence alignment information is considered while computing the overall score of the alignment. For this purpose, the classical global alignment recursion formula is modified to include a bonus score for aligning similar nucleotides on top of the score for aligning similar structural elements.

The time and space complexities for BEAR are the same as those for Needleman-Wunch, $O(nm)$ time and $O(n + m)$ space. This does not include the calculation of the secondary structure, which is a costly operation, but for each sequence the secondary structure is only calculated once.

RNAplfold

RNAplfold is part of the Vienna Package 2.0 [29], a collection of RNA secondary structure related computational tools. These tools use concepts from thermodynamics, such as minimum free energy, in order to predict the structure of RNA sequences. RNAplfold is a tool for base-pairing probabilities prediction. An important output of RNAplfold are probability estimates of each subsequence of length u

to be single stranded in the RNA secondary structure. By setting $u = 1$, the probability of each nucleotide to be unpaired is calculated. Figure 2-3 demonstrates a result of such use.

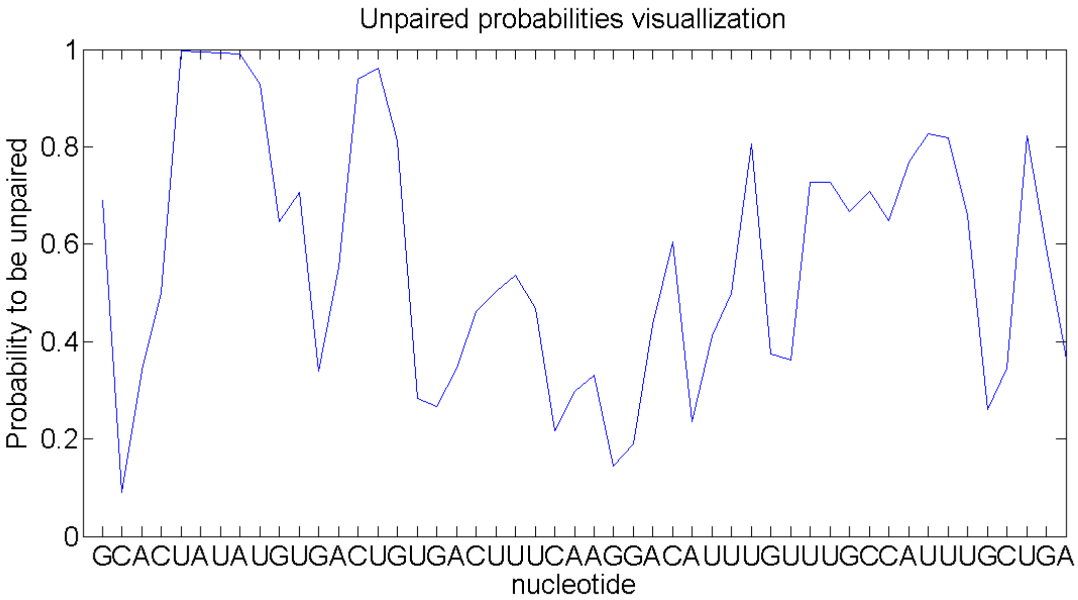

Figure 2-3 Visualization of the unpaired base probabilities vector for u=1 calculated by RNAplfold on the sequence shown at the bottom.

Compalignp

Compalignp [30] scores how similar a given alignment is to a reference alignment. Given a reference alignment of two sequences and another alignment of them, Compalignp measures what percentage of aligned characters and gaps from the reference alignment exists in the tested alignment.

The Compalignp score is calculated in the following manner: The input is a pair of sequences T, S of lengths m, n respectively, A_r, a reference alignment for T and S, and A_t another alignment of T and S that we wish to score.

Let $P_{ref} = \{(i,j) \mid S[i]\ aligned\ with\ T[j]\ in\ A_r\}_{1\le i\le n, 1\le j\le m}$

$G_{ref} = \{(i,-1) \mid S[i]\ is\ an\ insertion\ in\ A_r\}_{1\le i\le n} \cup \{(-1,j) \mid T[j]\ is\ an\ insertion\ in\ A_r\}_{1\le j\le m}$

and $P_{test} = \{(i,j) \mid T[i]\ aligned\ with\ S[j]\ in\ A_t\}_{1\le i\le n, 1\le j\le m}$

$G_{test} = \{(i,-1) \mid T[i]\ is\ an\ insertion\ in\ A_t\}_{1\le i\le n} \cup \{(-1,j) \mid S[j]\ is\ an\ insertion\ in\ A_t\}_{1\le j\le m}$

The Complalignp score is defined as:

$$\frac{2\cdot|\{P_{ref}\cap P_{test}\}|+|\{G_{ref}\cap G_{test}\}|}{2\cdot|P_{ref}|+|G_{ref}|}.$$

2.2.6 Rfam Database

Rfam is public collection of multiple sequence alignments of ncRNA families. Each family entry in Rfam consists of the seed alignment, a model of that seed alignment used for finding additional candidates for that family, and an extended multiple alignment. The seed alignment is a manually curated alignment of known RNA sequences in that family. The seed alignment can be used to extend the alignment of the family to a larger multiple alignment that includes the newly found RNA sequences. [31] Figure 2-4 shows an example of an Rfam entry.

```
# STOCKHOLM 1.0
#=GF AC   RF01382
#=GF ID   HIV-1_SL4
#=GF DE   HIV-1 stem-loop 4 packaging signal
#=GF AU   Chen A, Brown C, Daub J
#=GF SE   Chen A
#=GF SS   Published; PMID:18713870
#=GF GA   31.00
#=GF TC   31.10
#=GF NC   30.90
#=GF TP   Cis-reg;
#=GF BM   cmbuild -F CM SEED
#=GF CB   cmcalibrate --mpi CM
#=GF SM   cmsearch --cpu 4 --verbose --nohmmonly -T 21.20 -Z 549862.597050
CM SEQDB
#=GF DR   SO; 0005836; regulatory_region;
#=GF RN   [1]
#=GF RM   18713870
#=GF RT   MS3D structural elucidation of the HIV-1 packaging signal.
#=GF RA   Yu ET, Hawkins A, Eaton J, Fabris D
#=GF RL   Proc Natl Acad Sci U S A. 2008;105:12248-12253.
#=GF WK   Retroviral_Psi_packaging_element
#=GF SQ   16

X04415.1/351-370     UGGGUGCGAGAGCGUCAGUA
K03454.1/337-356     UGGGUGCGAGAGCGUCAGUA
M22639.1/791-810     UGGGUGCGAGAGCGUCAGUA
M62320.1/258-277     UGGGUGCGAGAGCGUCAGUA
K03455.1/791-810     UGGGUGCGAGAGCGUCAGUA
M38429.1/791-810     UGGGUGCGAGAGCGUCAGUA
DQ396394.1/226-245   UGGGUGCGAGAGCGUCAAUA
AY169803.1/272-291   UGGGUGCGAGAGCGUCUGUG
AF418366.1/2-21      UGGGUGCGAGAGCGUCAGUU
AY134942.1/2-21      UGGGUGCGAGAGCGUCAGAA
EU047601.1/21-40     UAGGUGCGAGAGCGUCAGUA
AJ251056.1/3-22      AGGGUGCGAGAGCGUCAGUG
AY169805.1/239-258   UGGGUGCGAGUGCGUCAGUG
EF394357.1/353-372   UGGGUGCGAGAGCGUCAGUG
AF382828.1/9328-9347 UGGGUGCGAGAGCGUCUAUA
DQ373064.1/337-356   UGGGUGCGAGAGCGUCAAUC
#=GC SS_cons         ::<<<<<____>>>>>::::
#=GC RF              UGGGUGCGAGAGCGUCAGUA
//
```

Figure 2-4 A sample Rfam seed that was used in this thesis.

3 Methods

The goal of this thesis is to develop a fast and efficient tool for multiple alignment of ncRNAs based on both sequence and structure. This chapter will formally define the problems as well as our proposed solutions.

3.1 Problem Input and Output

Input:

a. Set of sequences, $S = \{S_1, S_2, \dots, S_N\}$ over alphabet $\Sigma = \{A, C, G, U\}$
b. Set of probability vectors, $P = \{P_1, P_2, \dots, P_N\}$ where P_i is the base unpairing probability vector for sequence S_i

Output: An MSA of the input sequences S optimizing a target function that considers both sequence and structure similarities. The concrete function will be described later.

Most of the thesis will deal with the case of $N = 2$, i.e., pairwise sequence and structure alignment.

3.2 Structural Information Calculation

We used RNAplfold (see chapter 2.2.5) to calculate unpairing probability vectors for every input sequence. The calculation is performed once for each sequence in time complexity $O(n^3)$, where n is the length of the sequence. The results are stored as vectors of real numbers using $O(n)$ space.

Since RNAplfold uses a sliding window of fixed size when calculating its output, it can be argued that it is possible to run these calculations once for the entire genome (or subsequences annotated as ncRNA sequences) at a reasonable time. This long calculation will count as pre-processing and will later allow us to run our algorithm for any given set of sequences from the genome. Thus, in the algorithm running time analysis we exclude the structure prediction run time, and assume the probability vectors are given as input.

3.3 Local Alignment

3.3.1 Objective

We define a target function intended to measure the quality of alignment of two sequences. The function has four parameters. m$atch, mismatch,$ gap and $length\_penalty$. Given sequences S and T with unpairing probability vectors S_p and T_p where

$$|S| = |S_p| = n \text{ and } |T| = |T_p| = m$$

An alignment is obtained by adding spaces to the vectors forming l-long vectors such that

S^* = RNA sequence S of size n with $(l - n)$ spaces inserted ('-')
T^* = RNA sequence T of size m with $(l - m)$ spaces inserted ('-')
S_p^* = Unpaired probabilities for sequence S with $(l - n)$ spaces inserted ('-')
T_p^* = Unpaired probabilities for sequence T with $(l - m)$ spaces inserted ('-')

The spaces in S^* and S_p^* are in the same positions, and the spaces in T^* and T_p^* are in the same positions. There are no positions i with $S^*[i] = T^*[i] =' - '$.
The Score of the alignment is defined as:

$$F_t = \sum_{i=0}^{l} F\,(S^*[i], T^*[i])$$

With

$$F(S[i], T[i]) = \begin{cases} match \cdot \text{geometric\_}mean(S_p[i], T_p[i]) & S[i] = T[i] \\ mismatch \cdot \text{geometric\_}mean(S_p[i], T_p[i]) & S[i] \neq T[i] \\ gap \cdot S_p[i] & T[i] =' - ' \\ gap \cdot T_p[i] & S[i] =' - ' \end{cases}$$

In order to adjust the target function to be higher for short similar unpaired sub-sequences we introduce a length penalty which will be explained in the following section.

$$F_t^* = \sum_{i=0}^{l} F\,(S^*[i], T^*[i]) - l \cdot length\_penalty$$

Our goal is to find an alignment of maximum score.

3.3.2 Modified Smith-Waterman

We modified the classical Smith-Waterman algorithm for local alignment to incorporate structural data.

LASSP (***L**ocal **A**lignment using **S**equence and **S**tructure **P**robabilities)*

Definitions:

S = RNA sequence 1 (size n)
T = RNA sequence 2 (size m)

S_p = RNAplfold probabilities for sequence 1 (size n)
T_p = RNAplfold probabilities for sequence 2 (size m)
(probabilities that a single base is unpaired in the secondary structure of the RNA)

$length\_penalty,\ match, mismatch$ and gap are parameters used to calculate alignment score.

Initialization:

$M \coloneqq (n+1) \times (m+1) matrix$

$$M[0,i] = 0 \qquad \forall\ 0 \le i \le m$$

$$M[i,0] = 0 \qquad \forall\ 0 \le i \le n$$

Recursive calculation of matrix M:

$$M[i,j]\ =\ max \begin{cases} 0 \\ M[i-1,j-1] + \sigma(i,j) - length\_penalty \\ M[i,j-1] + g_S(j) - length\_penalty \\ M[i-1,j] + g_T(i) - length\_penalty \end{cases}$$

Match and mis-match scores:

$$\sigma(i,j) =\ \text{geometric\_}mean(S_p[i], T_p[j]) \cdot \begin{cases} match\text{h} & S[i] = T[j] \\ mismatc\text{h} & else \end{cases}$$

$$g_S(j) = gap \cdot T_p[j]$$

$$g_T(i) = gap \cdot S_p[i]$$

Optimal solution: $\max_{i,j} M[i,j]$

Backtracking can be used to find the optimal alignment.

It can be easily shown that this algorithm's output alignment optimizes F_t^*, defined in the previous chapter.

A mis/match score of a pair is multiplied by the geometric mean of the bases unpairing probabilities, thus making alignment pairs with high probabilities to be structurally unpaired receive higher scores. The $length\_penalty$ parameter is used to make the algorithm prefer shorter results and score them higher.

In our algorithm, the longer the alignment is the more penalty it suffers due to *length_penalty*. This modification prevents the algorithm from adding pairs to the alignment unless the matching score for these pairs, or sets of pairs is above a certain threshold. As a result, the higher *length*_penalty is, the more the algorithm will prefer short sequences with few structural base pairs. The other three parameters are used in the same way they are used in the original Smith-Waterman algorithm.

3.3.3 Differences from Local Sequence Alignment

The most significant difference from the original algorithm is that the contribution to the alignment score by each position in the alignment is multiplied by the probability that the bases in that position are structurally accessible (i.e. unpaired). The two different probabilities in the case of a match or mismatch of two bases are summarized by their geometric mean. Geometric mean was selected in order to prefer matches with high availability (unpaired) probabilities, and to penalize very different probabilities.

Another difference is the addition of a negative factor that grows linearly with the alignment length in order to prefer short alignments

3.3.4 Runtime and Space Analysis

The time complexity of LASSP is the same as the original Smith-Waterman. Each cell of the matrix M is calculated based on previously calculated values of M in $O(1)$ time. Since M is an $(n+1) \times (m+1)$ matrix, the time needed for filling the entirety of M is $O(nm)$.

As with classical local alignment, there are various methods that allow linear space complexity [20]. These methods can be applied since the basic assumption that for calculating the optimal alignment score we only need to store $O(n)$ matrix cells (2 rows/columns) is preserved.

3.4 Global Alignment

3.4.1 Objective

We define a target function intended to measure the quality of alignment of two sequences. The function has four parameters. m$atch, mismatch,$ $structural\_score$ and gap. Given sequences S and T with unpairing probability vectors S_p and T_p where

$$|S| = |S_p| = n \text{ and } |T| = |T_p| = m$$

An alignment is obtained by adding spaces to the vectors forming l-long vectors such that

S^* = RNA sequence S of size n with $(l-n)$ spaces inserted ('-')
T^* = RNA sequence T of size m with $(l-m)$ spaces inserted ('-')
S_p^* = Unpaired probabilities for sequence S with $(l-n)$ spaces inserted ('-')
T_p^* = Unpaired probabilities for sequence T with $(l-m)$ spaces inserted ('-')

The spaces in S^* and S_p^* are in the same positions, and the spaces in T^* and T_p^* are in the same positions.
There are no positions i with $S^*[i] = T^*[i] = '-'$.
The Score of the alignment is defined as:

$$F_t = \sum_{i=0}^{l} F\,(S^*[i], T^*[i])$$

With

$$F(S[i],T[i]) = \begin{cases} match + structural\_score \cdot (1 - |S_p[i] - T_p[i]|) & S[i] = T[i] \\ mismatch + structural\_score \cdot (1 - |S_p[i] - T_p[i]|) & S[i] \neq T[i] \\ gap & T[i] = '-' \\ gap & S[i] = '-' \end{cases}$$

Our goal is to find an alignment of maximum score.

3.4.2 Modified Needlman-Wunch

Needleman-Wunch algorithm solves global alignment of two sequences. As we did with our version of Smith-Waterman, we adjusted Needleman-Wunch to take structural properties of the RNA sequence into account.

GASSP (***G**lobal **A**lignment using **S**equence and **S**tructure **P**robabilities)*

Definitions:

S = RNA sequence 1 (size n)
T = RNA sequence 2 (size m)
S_p = RNAplfold probabilities for sequence 1 (size n)
T_p = RNAplfold probabilities for sequence 2 (size m)
$match, mismatch$, gap and $ratio$ are parameters used to define the preference of the algorithm.

Initialization:

$M \coloneqq (n+1) \times (m+1)\ matrix$

$$M[0,i] = i * gap$$
$$M[i,0] = i * gap$$

Recursive calculation of matrix M:

$$M[i,j] \;=\; max \begin{cases} M[i-1,j-1] + \sigma(i,j) \\ \quad M[i,j-1] + gap \\ \quad M[i-1,j] + gap \end{cases}$$

Match and mis-match scores:

$$\sigma(i,j) = \; match\mathrm{h} \cdot ratio \;\cdot \left(1 - \left|S_p[i] - T_p[j]\right|\right) + \begin{cases} \;\; matc\mathrm{h} & S[i] = T[j] \\ mismatc\mathrm{h} & \quad else \end{cases}$$

Optimal solution: $M[n,m]$

Backtracking can be used to find the optimal alignment.

It can be easily shown that this algorithm's output alignment optimizes F_t, defined in the previous chapter.

3.4.3 Difference from Global Sequence Alignment

The only difference we introduced to the Needleman-Wunch original algorithm was that in addition to the match/mismatch score for each non-gapped position in the alignment, a structural similarity score is also given. The similarity is measured by the difference between the pairing probabilities of matched bases.

$$s(P_a, P_b) = 1 - |P_a - P_b|$$

If the probabilities are similar, the similarity score would be high. If the probabilities are further apart, the score can be as low as zero. Since $P_a, P_b \in [0,1]$, the difference $|P_a - P_b|$ is also within this range, and $1 - |P_a - P_b|$ is also in the range $[0,1]$.

3.4.4 Differences between GASSP and LASSP

One obvious difference is the absence of the *length_penalty* which was used in LASSP to allow biasing the result to shorter alignments of mostly unpaired structure. The way we take the pairing probabilities into account was also changed. Geometric mean (as was used in LASSP), which was fitting for biasing towards substructures with higher unpairing probability, was replaced by a simple difference measurement which is neutral (with respect to biasing towards specific structure) and linear.

3.4.5 Runtime and Space Analysis

The time complexity of GASSP is the same as that of the original algorithm. Each cell of the matrix M is calculated based on previously calculated values of M in $O(1)$ time. Since M is an $n \times m$ matrix, the time needed for filling the entirety of M is $O(nm)$.

On the matter of space complexity, as with LASSP, the basic assumptions of Hirschberg's method [20] can be applied to our implementation to achieve a linear space complexity.

3.5 Improving LocaRNA speed using a reference alignment

As described in Chapter 2, LocaRNA can be limited to run on a very small subset of alignments and structures. In order to enable this, a prior alignment can be provided. The better the provided prior alignment is, the higher the chance that the non-banded result is close enough to be within the narrowed search space of LocaRNA. This will allow LocaRNA to find its solution much faster.

3.6 Multiple Sequence Alignment

We implemented a UPGMA based progressive MSA using GASSP. The metric used for the UPGMA is the pairwise alignment score. For cells in the dynamic programming matrix the score computation is as follows: Aligning two groups of sequences, the score is taken to be the mean score for all possible combinations of two sequences (one from each group).

4 RESULTS

4.1 Local Alignment Results

4.1.1 Data Source

We used protein-RNA binding data to test and validate our algorithm. As input sequences we used protein-RNA bindings, as measured by CLIP experiments [32]. Each dataset comprised of a large set (a few tens of thousands) of about 40nt long RNA sequences, all derived from a single CLIP experiment. The bound peaks were extended by 150nt downstream and upstream for more accurate structure prediction. These flanking sequences were removed following the structure prediction.

We concentrated our efforts on data from one CLIP experiment for the protein ELAVL1 which produced 23,455 peaks.

4.1.2 Benchmark

We tested LASSP in finding binding sites in CLIP data for an RBP that also had RNAcompete data. The local alignments are predictions of binding sites of specific RBPs. We used RNAcompete 7-mer binding scores of the same RBP to evaluate our predictions. Since the length of the alignment is bounded only by the length of the sequences, we developed a way to use 7-mer scores on arbitrary length sequences (see **Error! Reference source not found.**).

For each sequence in the alignment, we removed all gaps. Then, we calculated the average 7-mer score of all 7-mers that appear in the sequence. If a sequence is shorter than 7 bases, the average of the scores of all 7-mers that contain the sequence is taken as its score. The alignment score is the sum of the two scores (one for each sequence). All RNAcompete 7-mer scores (which are Z-scores) were normalized per experiment by dividing by the maximum 7-mer score's absolute value, so all scores are between -1 to 1.

```
Function RNAcompeteScore(Sequence $S$):
If ($|S|$ == 7):
    Return RNAcompeteZScore[$S$]
Else If ($|S|$ > 7):
    Sum ← 0
    For i = 1 to $|S| - 7 + 1$:
        Sum ← Sum + RNAcompeteScore( $S$[i:i+7-1] )
    Return Sum/($|S|$– 7+1)
Else:
    Sum ← 0
    For c in $\{'A','C','G','U'\}$:
        Sum ← Sum + RNAcompeteScore( Concat( $S$,c) ) + RNAcompeteScore( Concat(c,$S$) )
    Return Sum / (2 · $|\{'A','C','G','U'\}|$)
```

Algorithm 4-1 Arbitrary length RNAcompete based scoring algorithm.

4.1.3 RNAcompete Score

Using the above scoring method for one sequence, each alignment of two sequences is scored by summing the scores of its two projected subsequences for a score between -2 and 2. When benchmarking a set of N sequences, all possible pairs $\binom{N}{2}$ are aligned and scored. We denote the sum of all scores the *RNAcompete total score* for the sequence set.

4.1.4 Parameter Optimization

There are four parameters in the formula, and the resulting score depends on the values of these parameters. Since the final score has no specific scale, there are actually only three degrees of freedom in modifying the algorithm using these parameters. For this reason, we set the $length\_penalty$ parameter arbitrarily to 3. The other three parameters were chosen from the following ranges:

$match \in [0,100]$
$mismatch \in [-1000,0]$
$gap \in [-1000,0]$

We tested many different combinations (20 values from $mismatch$ and gap for several $match$ values totaling in a few hundred combinations). We implemented our algorithm and ran it to pairwise align thousands of RNA sequences. For each parameter set, millions of alignments were made and scored. The sum of the scores of each alignment represents the score for the set of parameters.

4.1.5 Results

Figure 4-1 shows the scores obtained for alignments of CLIP sequences for the RBP ELAVL1 for different parameter combinations. Using these tests, we were able to find the parameters most suitable for the task of locating ELAVL1 binding site in multiple sequences of RNA.

After testing many combinations of all parameters, we chose to use match score of 40. For values far from 40, we were not able to locate a local optimum for the other parameters. With a match score of 40, the optimal mismatch and gap score were -100 and -60, respectively. We did not test other proteins thus we state the optimal parameters for ELAVL1 only.

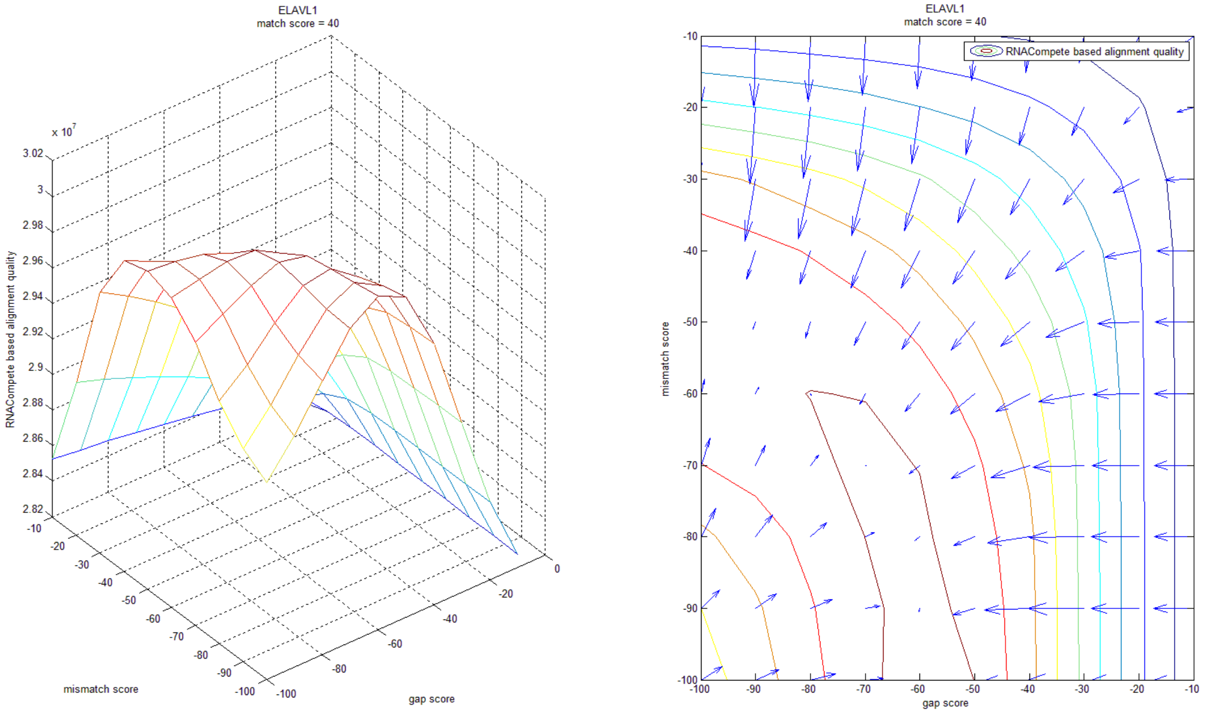

Figure 4-1
RNAcompete total score for pairwise alignment of CLIP results for RBP ELAVL1 [32].
The graphs show how the total score is affected by changing the gap and mismatch parameters values while maintaining match parameter value of 40 constant. In the right figure gradient markers were also added to better visualize parameter optimization landscape.

4.2 Global Alignment Results

4.2.1 Data Source

BRAliBase study includes an extensive comparison of RNA alignment algorithms [33]. The dataset of BRAliBase contains the sets of sequences and the 'ground truth' of their alignment. The dataset comprises of a few subsets, where each one is a collection of MSAs for similar RNA sequences. One of the subsets for example contains 89 MSAs of different groups of rRNA sequences. Each MSA contains on average around 5 sequences.

Each MSA can be converted back to a set of unaligned sequences and used as input for GASSP. For pairwise alignment validation (as opposed to MSA) we generated all possible pairs of unaligned sequences, and used the projection of the MSA on them as the ground truth. Note that this may create bias as two sequences may be aligned better in isolation than in their projected alignment in the MSA. The number of tested pairs is listed in Table 4-1.

Group	Pairs count
g2intron	920
rRNA	890
tRNA	980
U5	1,080

Table 4-1 Sequence pair count for different groups generated from BRAliBase.

4.2.2 Benchmark

In BRAliBase study the authors used Compalignp to compare the alignments produced by various tools to manually curated alignments. We used the same tool to give each of our alignments a score between 0 and 1, allowing us to compare GASSP's results to those published for other tools.

For each subset from BRAliBase, the mean Compalignp score was calculated for GASSP's results and then compared to different tools and to different parameter sets.

4.2.3 Parameter Optimization

We tested parameters in the following ranges:

$$match = 100$$

$$\text{ratio} \in [0.1,2]\cup\{0\}$$

$$mismatch \in [-1000,1000]$$

$$gap \in [-1000,1000]$$

Every position (that is not a gap) in the alignment is assigned a score that is the sum of a sequence similarity score and a structure similarity score, balanced by the ratio parameter. This results in a structure similarity score cap in the range $[0,200]$ or $[0, 2 \cdot match]$

This cap is multiplied by the structural score $\boldsymbol{s}$ described in chapter 3.4.3 which is in the range $[0,1]$.

Setting this cap to 0 ($ratio = 0$) is equivalent to using a simple Needleman-Wunch and discarding the structural score completely.

Though there are four parameters in the formula, there are actually only three degrees of freedom in modifying the algorithm using these parameters. For this reason, we decided to set the $match$ parameter arbitrarily to 100. The other three parameters were chosen from the ranges stated above.

After testing roughly in the ranges above, we refined the parameters using the following ranges for finding a local optimum of our parameters:

$$match = 100$$

$$ratio \in [0.1,2.0] \cup \{0\}$$

$$mismatch \in [-100,100]$$

$$gap \in [-200,0]$$

We tested 20 different values (uniformly distributed within the above ranges) for each parameter for a total of 8,000 sets of parameters.

4.2.4 Results

Our goal is to find a robust set of parameters based on the data from BRAliBase that will work well on different RNA families from different sources. Our results indicate that for different types of RNA sequences (e.g. rRNA, g2intron, tRNA) the optimal parameters are different. Figure 4-2 shows our results for taking an entire dataset from BRAliBase, called BRAliBase II dataset 1 containing 3870 pairs (all the pairs in Table 4-1), and finding the best parameters for GASSP. (limited to ratios between 0.1 and 0.9)

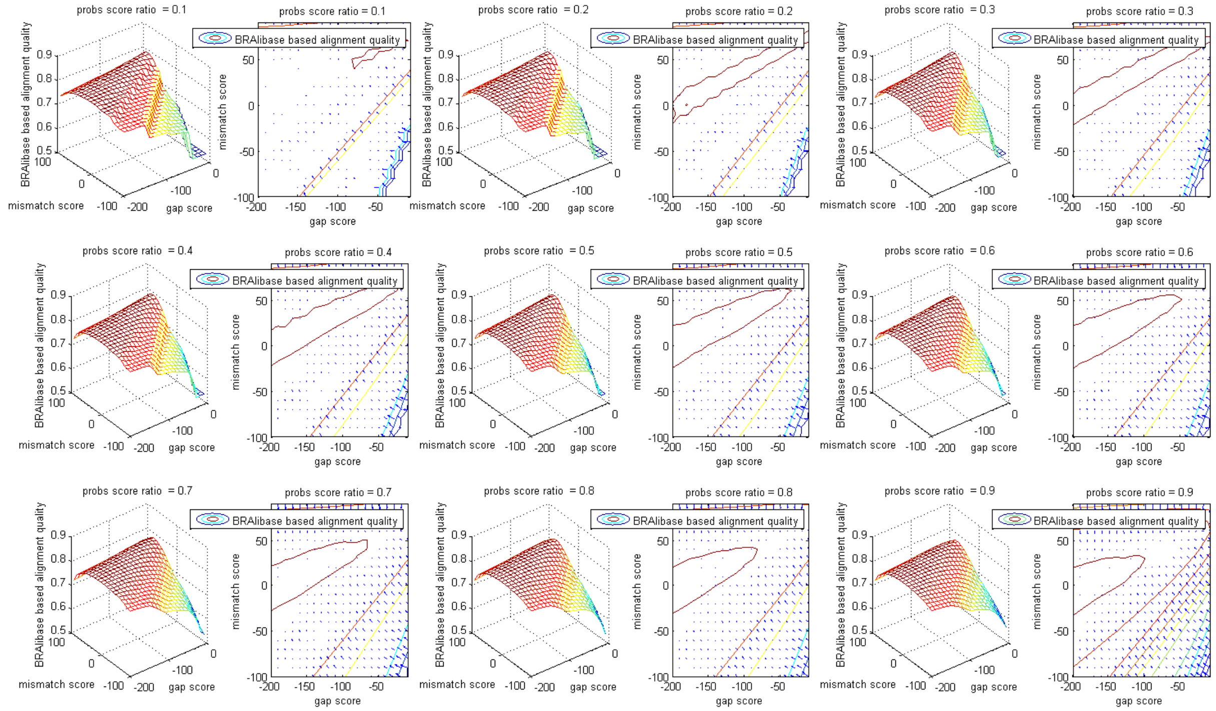

Figure 4-2 Compalignp scores for GASSP tested on BRAliBase II dataset 1 (all groups) for various parameters. Best Compalignp score – 0.8026. Best parameters – ratio = 0.6, mismatch = 10, gap = -180. Subplots correspond to different values of ratio parameter between 0.1 to 0.9 inclusive.

4.2.5 The Rfam Test

We used Rfam database to compare GASSP's performance to a classical global alignment tool (Needle). Using Rfam database has several advantages over BRAliBase:

1. Rfam is larger, so broader conclusions can be drawn.
2. Rfam is more versatile and therefore less prone to bias our results.

The Rfam seeds contain 2450 manually curated MSAs, containing between 19 to 8,395 sequences each (average of 143). Each MSA corresponds to a different RNA family.

Each MSA was split into all possible sequence pairs, as we did with BRAliBase. We then removed all gap characters, resulting in sets of unaligned sequence pairs. Each sequence pair in every set was aligned twice. Once with the Needle tool using default parameters, and once with GASSP (with the optimal parameters computed on BRAliBase). The aligned pairs were compared to the original alignment taken from Rfam MSA and were graded using the Compalignp tool. We then calculated the average score for each Rfam family and compared the needle score to GASSP's score. We also compared our score to that of our algorithm with the *ratio* parameter set to 0. This ignores structure information, making it a classical simple Needleman-Wunch tool but with parameters trained on BRAliBase.

Figure 4-3 and Figure 4-4 shows the performance of GASSP and Needle for different families. It highlights subsets (longest and shortest by alignment length) of the results. It can be seen that in most cases our results are better but there are RNA families where Needle is more accurate in aligning the input sequences. The influence of the sequence length is clear from this figure. For shorter sequences GASSP's performance is better than for longer sequences. Figure 4-4 shows all the differences between the GASSP and classic scores plotted against the family's seed MSA length. Overall, GASSP scores on average 0.036 higher than needle on Rfam seeds with a p-value of 1.71×10^{-73} as measured by a paired sample T-test.

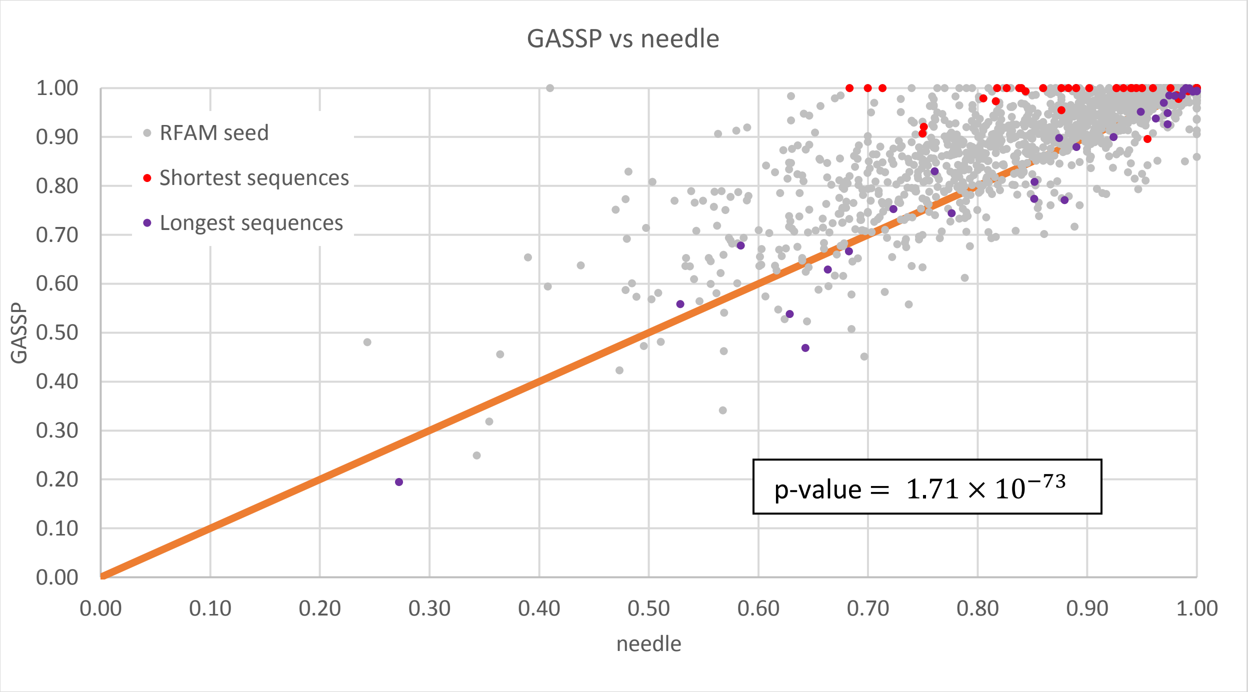

Figure 4-3: GASSP performance compared to Needle. Each spot represents the average score computed for one Rfam family as measured by Compalignp. Shortest (≤40nt) and longest (≥400nt) seeds are colored. The orange line describes the identity function.

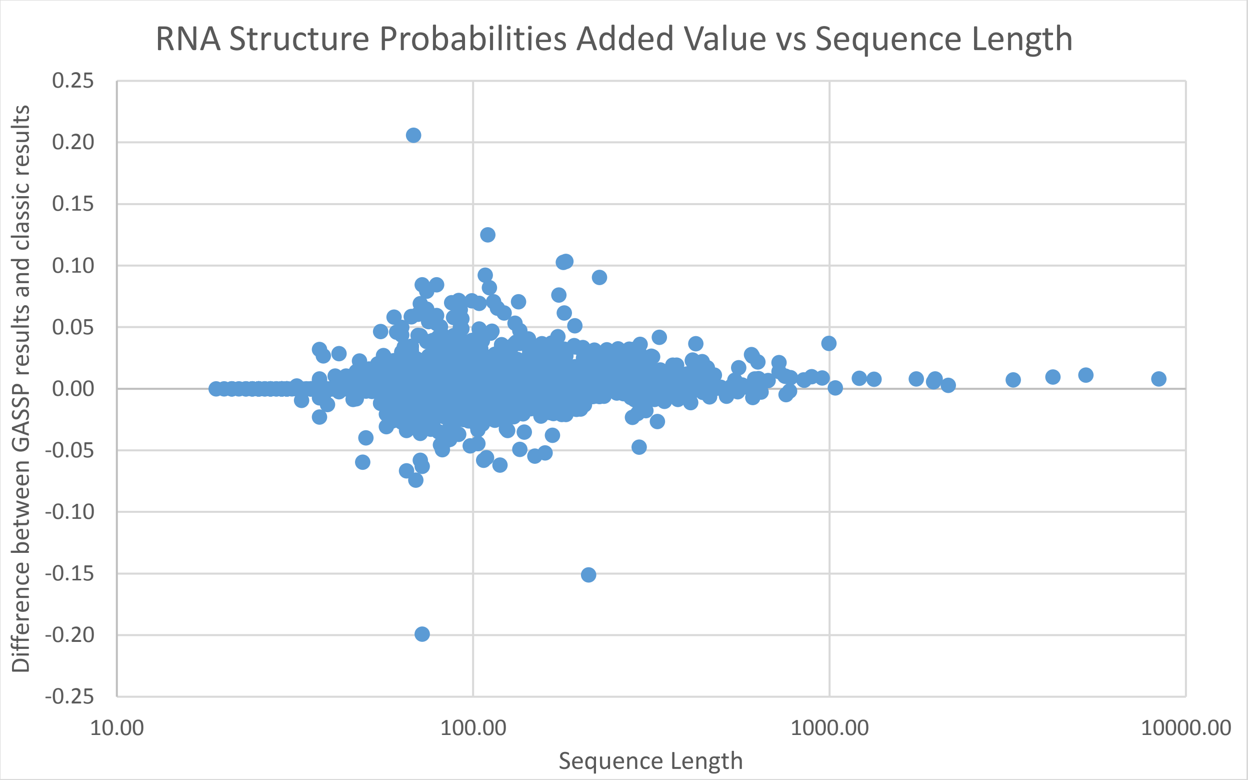

Figure 4-4 Difference between GASSP and Needle scores across Rfam families. Each spot shows the difference for a single Rfam family between the GASSP score and the Needle score (GASSP with $ratio = 0$). The scores are Compalignp values for the match between the computed and the reference alignment. X axis: the family's seed MSA length in logarithmic scale.

Rfam seeds are also categorized to super families. We calculated the mean difference between GASSP's score and the score of a classical sequence alignment for different super families. Figure 4-5 shows which Rfam super families showed meaningful difference in terms of p-value (using paired T-test and Bonferroni correction). For each super family, two scores where generated. One vector of the results for classical sequence-based alignment, and another one that includes structural features analysis. These results suggest that some super families consist of RNAs with functions that are more dependent on secondary structure than others.

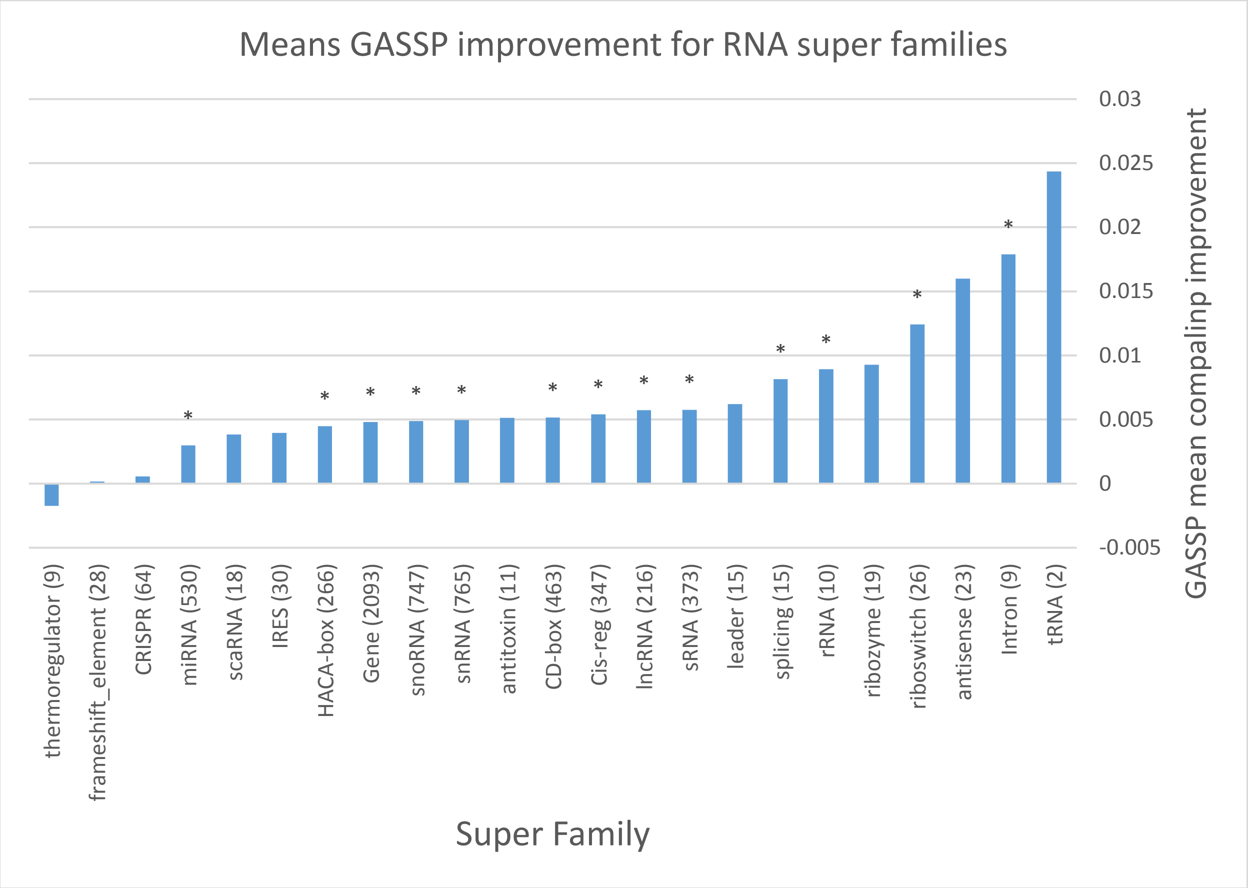

*Figure 4-5 Mean difference between GASSP's Compalignp scores and those of a classical sequence alignment for different RNA super families. Significant results (corrected p-value smaller than 0.05) are marked with *. The size of each family is shown in parentheses.*

4.3 Using GASSP to improve LocaRNA

4.3.1 Benchmark

LocaRNA can start computation from an input alignment (a "seed" solution). By starting from a seed and banding the computation (using the "max-diff" parameter) it can run faster, but the final solution may differ from that obtained without banding and an initial solution. We wanted to test how running LocaRNA using as the seed the alignment obtained by GASSP improves the results (1) compared to running without a seed, and (2) compared to running with a seed of the regular global alignment solution, which uses sequences only and no structural information.

4.3.2 Results

We ran LocaRNA on 1657 RFAM families' seed alignments data (See Chapter 4.2.5) twice: using GASSP alignments as seeds, and using the classic sequence alignment solution as seeds. For each family we aligned all possible pairs of seed sequences. We measured the quality of each resulting solution by comparing it to the pairwise alignments induced by Rfam seed alignments. The results are summarized in Figure 4-6. There were more cases where GASSP reference improved LocaRNA performance over classic (Needle) alignments than the other way around (173 vs 107; for the remaining families the two seeds produced identical scores). The mean score for GASSP based LocaRNA was 0.0004736 higher, with p-value 0.0233 calculated using a paired sample T-test. Hence, overall, GASSP was significantly (but mildly) better than classic alignment as a seed.

As for the running time, aligning 15 pairs of sequences from Rfam family "RF00224" with seed alignment length of 507nt took LocaRNA 44 minutes without reference. Aligning the same sequences with a reference and max-diff=50 took less than 4 minutes (see Figure 4-7). The alignment quality was the same. In fact, the same quality was observed for max-diff=10 for this family (see Figure 4-8). The figures also show the results for two other Rfam families, showing similar speed-ups. In those cases too, alignment quality was not harmed by using higher max-diff.

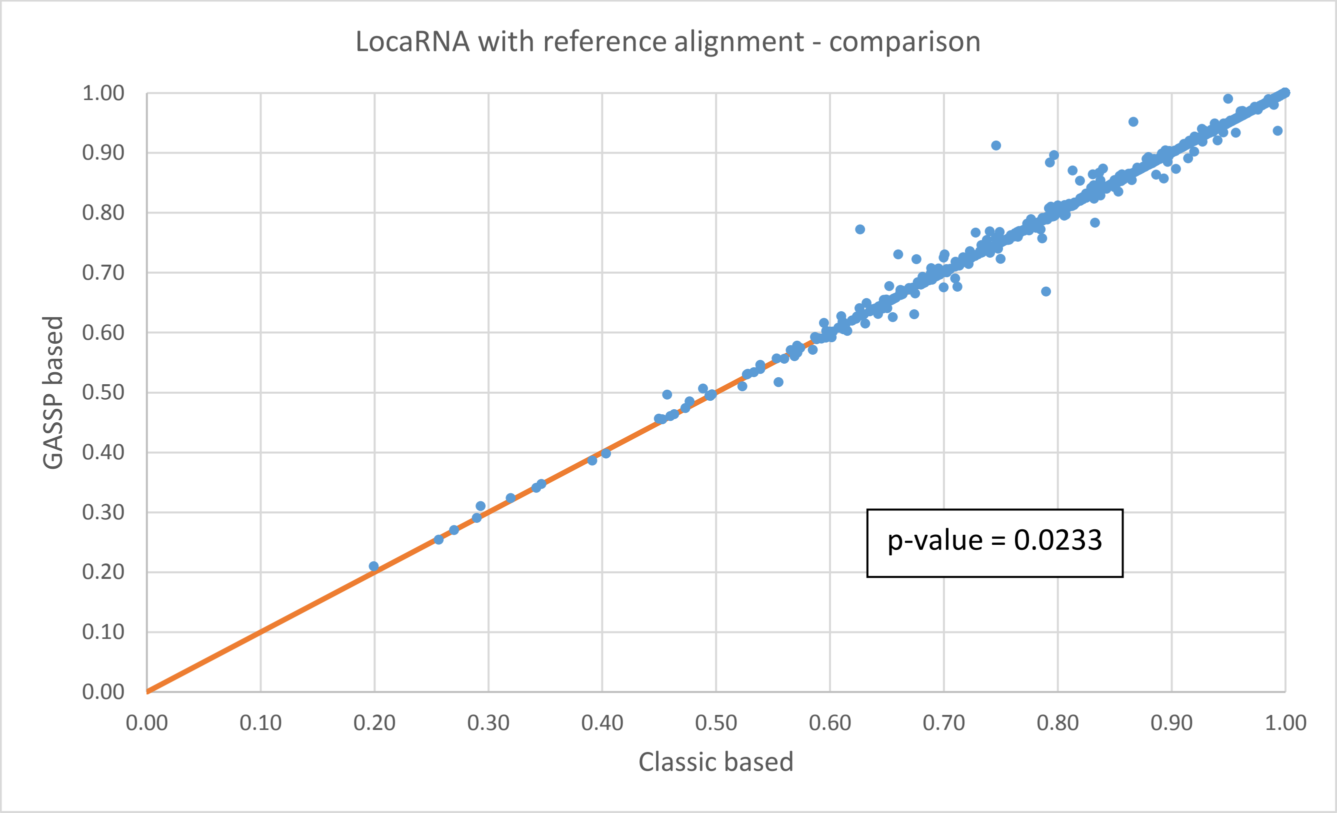

Figure 4-6 Each point on the graph shows, the mean Compalignp scores for LocaRNA when using GASSP solutions as a reference to align all pairs in a RFAM family, and that of LocaRNA with a needle results as a reference. The orange line is the identity function.

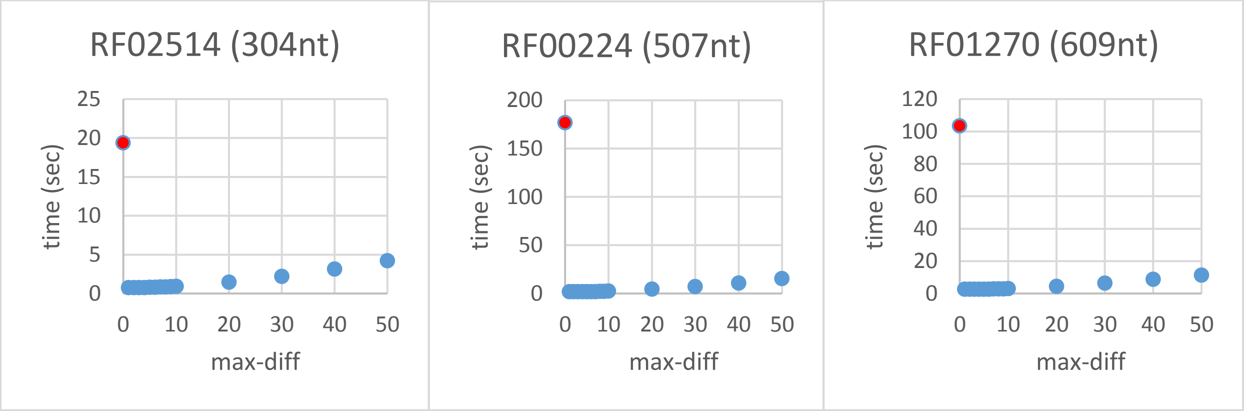

Figure 4-7 Running time of LocaRNA using a seed reference for different 'max-diff' values, on three Rfam families. The red marker on the left is the mean alignment time for regular LocaRNA. Aligned sequences are the seed alignment sequences of each Rfam family.

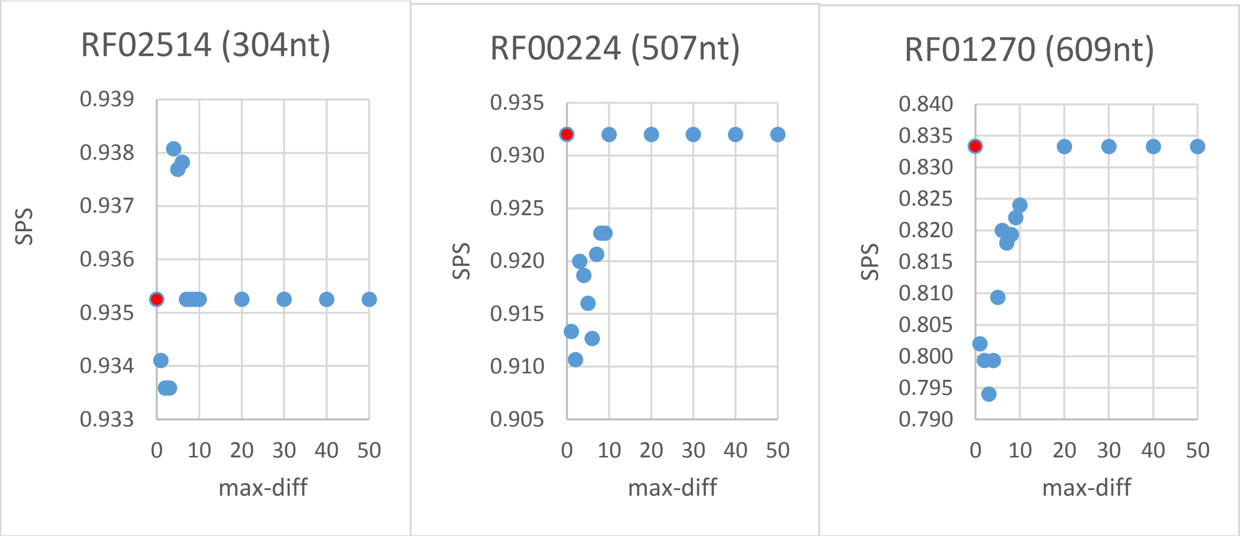

Figure 4-8 Sum-of-pairs score (SPS) of LocaRNA solutions using a seed reference for different 'max-diff' values, on the three Rfam families shown in the previous figure. The red marker on the left is the mean alignment score for regular LocaRNA. Aligned sequences are the seed alignment sequences of each Rfam family.

4.4 MSA

Our MSA solution was tested against Rfam seeds. We measured its quality by comparing results to the seeds reference MSA with Compalignp. In our MSA implementation we used GASSP while performing progressive multiple sequence alignment. Figure 4-9 shows the results for all Rfam families. The results show a slight degradation in alignment quality, but most GASSP-MSA results are close to those of the pairwise version. Figure 4-10 shows a histogram of the differences between GASSP-MSA and GASSP-pairwise scores for same seeds. The mean difference is -0.0205 with a p-value of 8.27×10^{-67} calculated using a paired sample T-test.

Comparing these results to the results of a standard Smith-Waterman based progressive MSA (i.e. GASSP-MSA with ratio parameter set to 0), we discovered that our extension does not contribute to the accuracy of the results. There was no significant difference between the results using a positive ratio value and using a zero value.

We also tried to use the GASSP-MSA solution as reference alignment for mLocaRNA (MSA version of LocaRNA). GASSP-MSA shows no significant improvement compared to the classic approach (neither in accuracy nor in running time of mLocaRNA).

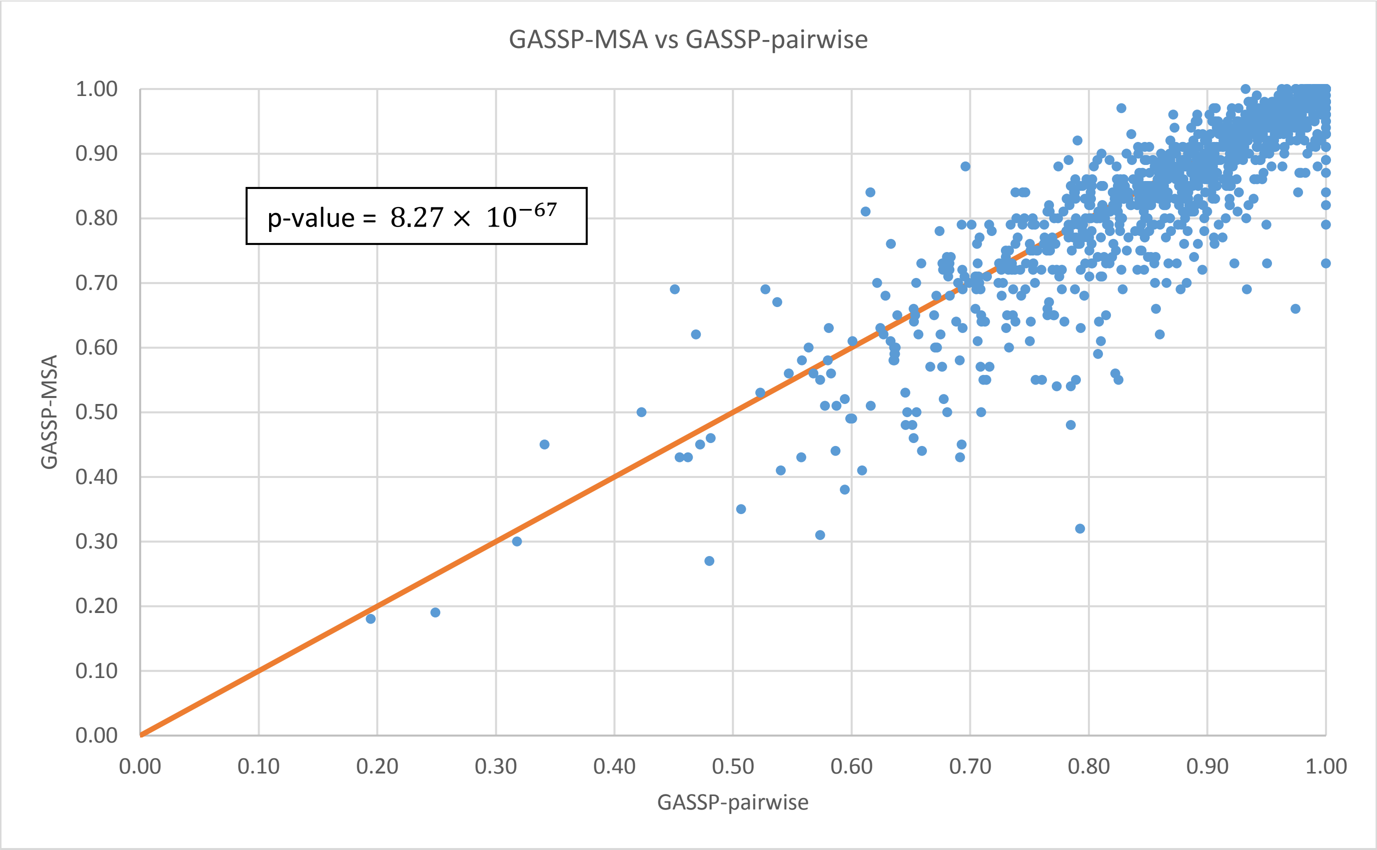

Figure 4-9 A scatter plot comparing GASSP-pairwise to GASSP-MSA. Each dot is an Rfam family. X axis: average Compalignp scores of GASSP-pairwise alignment. Y axis: average score for pairwise alignments based on GASSP-MSA. Orange line is the identity function.

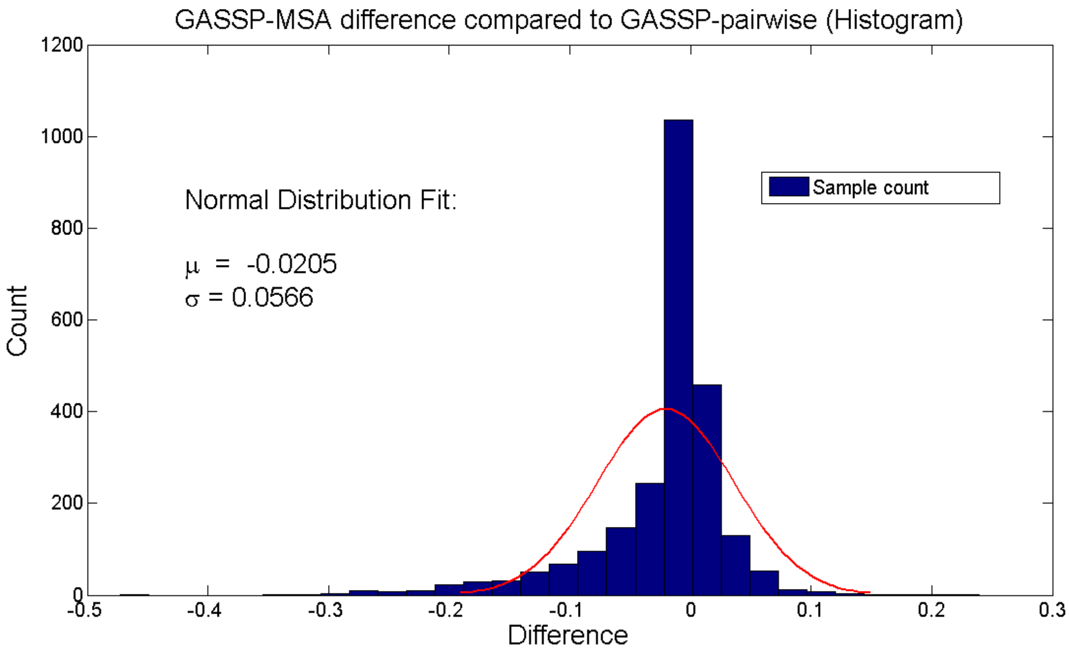

Figure 4-10 A histogram of the differences between GASSP-MSA and GASSP-pairwise scores.

5 CONCLUSION

The objective of this research was to develop an efficient method for sequence and structure-based alignment in ncRNA sequences. Classical motif finding methods take advantage of the fact that functional segments in the sequence are conserved through evolution. A recurring sequence motif in the DNA may therefore represent a functional element. Since RNA functionality depends on both its sequence and structure, we expect RNA molecules of similar functionality to contain a recurring sequence and structure motif.

We set out to develop an algorithm to align multiple sequences of ncRNA molecules based on their sequence and structure. We hoped that such multiple alignment tool will help improving motif finding in the large amounts of RNA data.

We first developed LASSP, an RNA sequence-structure local alignment tool with quadratic time complexity. We adapted it for aligning certain types of RBPs. While developing LASSP we realized it was hard to compare it to other algorithms in the field. To compare LASSP to different local alignment tools we needed a benchmark, but it was novel and therefore not yet reliable. Instead of pursuing other benchmarks, we decided to leave local alignment for future research. For global alignments and even multiple alignments, we did find a reliable benchmark that was already applied to various tools. We decided to adapt our algorithm's key ideas to global alignment and attempted to utilize the extra structural information to produce a fast and more accurate alignment tool for ncRNA. That effort yielded GASSP.

GASSP is an RNA sequence-structure global alignment tool that runs in quadratic time complexity. Comparing our results with classical global alignment, GASSP, having more degrees of freedom, could be adapted better to perform the alignment of BRAliBase pairs. The still open question is whether we succeeded in "training" our parameters and will achieve superior results over more data sets. We got a partial answer to that question when we used GASSP for aligning Rfam seeds. In this benchmark, using parameters learned on BRAliBase, GASSP outperformed sequence-only based alignments. GASSP proved to be superior to the popular alignment tool Needle in most cases when aligning RNAs of various types. GASSP's results also proved to be better than Needle in providing a starting reference alignment for LocaRNA, providing a small but significant improvement in results.

Finally, we based a Progressive MSA tool on GASSP. Results were slightly poorer in comparison to a pairwise (sequence-only) alignments of the same set of RNA sequences. Using our MSA as a reference for mLocaRNA proved to be unsuccessful.

As the wider goal of this thesis was to lay a foundation for secondary structure assisted motif finding in ncRNA, the research of LASSP should be extended. A reliable benchmark should be chosen and LASSP should be put to the test. A fast and accurate local alignment algorithm can be extended to a motif finding method [34] and we hope LASSP or an extension of it could be that algorithm.

To conclude, these methods have a long way to go before they can be reliably used. But it is our belief that the one-dimensional structural information of a RNA can and should be used instead of methods only applying sequence based comparison and alignment.

6 REFERENCES

[1] M. C. Frith, U. Hansen, J. L. Spouge, et al., “Finding functional sequence elements by multiple local alignment.,” *Nucleic Acids Res.*, vol. 32, no. 1, pp. 189–200, 2004.

[2] G. Badr, I. Al-Turaiki, and H. Mathkour, “Classification and assessment tools for structural motif discovery algorithms.,” *BMC Bioinformatics*, vol. 14 Suppl 9, no. Suppl 9, p. S4, 2013.

[3] D. Sankoff, “Simultaneous Solution of the RNA Folding, Alignment and Protosequence Problems,” *SIAM J. Appl. Math.*, vol. 45, no. 5, pp. 810–825, Oct. 1985.

[4] S. Will, K. Reiche, I. L. Hofacker, et al., “Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering.,” *PLoS Comput. Biol.*, vol. 3, no. 4, p. e65, Apr. 2007.

[5] S. Will, C. Otto, M. Miladi, et al., “SPARSE: quadratic time simultaneous alignment and folding of RNAs without sequence-based heuristics.,” *Bioinformatics*, vol. 31, no. 15, pp. 2489–96, Aug. 2015.

[6] E. Mattei, G. Ausiello, F. Ferrè, et al., “A novel approach to represent and compare RNA secondary structures.,” *Nucleic Acids Res.*, vol. 42, no. 10, pp. 6146–57, Jun. 2014.

[7] M. D. Simon, C. I. Wang, P. V. Kharchenko, et al., “The genomic binding sites of a noncoding RNA,” *Proc. Natl. Acad. Sci.*, vol. 108, no. 51, pp. 20497–20502, Dec. 2011.

[8] “RNA Secondary Structures.” [Online]. Available: http://www.bioinf.uni-leipzig.de/Leere/SS15/Bioinf2/lengauer.pdf. [Accessed: 30-Dec-2015].

[9] D. H. Mathews and D. H. Turner, “Prediction of RNA secondary structure by free energy minimization,” *Curr. Opin. Struct. Biol.*, vol. 16, no. 3, pp. 270–278, Jun. 2006.

[10] S. H. Bernhart, I. L. Hofacker, and P. F. Stadler, “Local RNA base pairing probabilities in large sequences,” *Bioinformatics*, vol. 22, no. 5, pp. 614–615, Mar. 2006.

[11] K.-K. Tong, K.-Y. Cheung, K.-H. Lee, et al., “GAknot: RNA secondary structures prediction with pseudoknots using genetic algorithm,” in *2013 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2013, pp. 136–142.

[12] D. Dominguez, P. Freese, M. S. Alexis, et al., “Sequence, Structure, and Context Preferences of Human RNA Binding Proteins,” *Mol. Cell*, vol. 70, pp. 854–867, 2018.

[13] R. B. Darnell, “HITS-CLIP: panoramic views of protein-RNA regulation in living cells.,” *Wiley Interdiscip. Rev. RNA*, vol. 1, no. 2, pp. 266–86, Jan. .

[14] D. Ray, H. Kazan, E. T. Chan, et al., “Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins.,” *Nat. Biotechnol.*, vol. 27, no. 7, pp. 667–70, Jul. 2009.

[15] D. Ray, H. Kazan, K. B. Cook, et al., “A compendium of RNA-binding motifs for decoding gene regulation,” *Nature*, vol. 499, no. 7457, pp. 172–177, Jul. 2013.

[16] W. Just, “Computational Complexity of Multiple Sequence Alignment with SP-Score,” *http://dx.doi.org/10.1089/106652701753307511*, vol. 8, no. 6, pp. 615–623, 2004.

[17] F. Corpet, “Multiple sequence alignment with hierarchical clustering.,” *Nucleic Acids Res.*, vol. 16, no. 22, pp. 10881–90, Nov. 1988.

[18] M. S. Rosenberg, *Sequence alignment : methods, models, concepts, and strategies*. University of California Press, 2009.

[19] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, Mar. 1970.

[20] D. S. Hirschberg, “A linear space algorithm for computing maximal common subsequences,” *Commun. ACM*, vol. 18, no. 6, pp. 341–343, Jun. 1975.

[21] T. F. Smith and M. S. Waterman, “Identification of common molecular subsequences.,” *J. Mol. Biol.*, vol. 147, no. 1, pp. 195–7, Mar. 1981.

[22] P. Rice, I. Longden, and A. Bleasby, “EMBOSS: The European Molecular Biology Open Software Suite,” *Trends Genet.*, vol. 16, no. 6, pp. 276–277, Jun. 2000.

[23] D. A. Pollard, C. M. Bergman, J. Stoye, et al., “Benchmarking tools for the alignment of functional noncoding DNA,” *BMC Bioinformatics*, vol. 5, no. 1, p. 6, 2004.

[24] P. H. A. Sneath and R. R. Sokal, “Numerical Taxonomy: The Principles and Practice of Numerical Classification.” 1973.

[25] I. L. Hofacker, S. H. F. Bernhart, and P. F. Stadler, “Alignment of RNA base pairing probability matrices,” *Bioinformatics*, vol. 20, no. 14, pp. 2222–2227, Sep. 2004.

[26] J. S. McCaskill, “The equilibrium partition function and base pair binding probabilities for RNA

secondary structure," *Biopolymers*, vol. 29, no. 6–7, pp. 1105–1119, May 1990.

[27] "Sequence-Structure Alignment-A General Formulation."

[28] K. M. Chao, W. R. Pearson, and W. Miller, "Aligning two sequences within a specified diagonal band.," *Comput. Appl. Biosci.*, vol. 8, no. 5, pp. 481–7, Oct. 1992.

[29] R. Lorenz, S. H. Bernhart, C. Höner Zu Siederdissen, et al., "ViennaRNA Package 2.0.," *Algorithms Mol. Biol.*, vol. 6, p. 26, Jan. 2011.

[30] A. Wilm, I. Mainz, G. Steger, et al., "An enhanced RNA alignment benchmark for sequence alignment programs," *Algorithms Mol. Biol.*, vol. 1, no. 1, p. 19, 2006.

[31] S. Griffiths-Jones, A. Bateman, M. Marshall, et al., "Rfam: An RNA family database," *Nucleic Acids Res.*, vol. 31, no. 1, pp. 439–441, Jan. 2003.

[32] K. Blin, C. Dieterich, R. Wurmus, et al., "DoRiNA 2.0--upgrading the doRiNA database of RNA interactions in post-transcriptional regulation.," *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D160-7, Jan. 2015.

[33] P. P. Gardner, "A benchmark of multiple sequence alignment programs upon structural RNAs," *Nucleic Acids Res.*, vol. 33, no. 8, pp. 2433–2439, Apr. 2005.

[34] M. C. Frith, U. Hansen, J. L. Spouge, et al., "Finding functional sequence elements by multiple local alignment.," *Nucleic Acids Res.*, vol. 32, no. 1, pp. 189–200, 2004.

[35] D. P. Aalberts and W. K. Jannen, "Visualizing RNA base-pairing probabilities with RNAbow diagrams.," *RNA*, vol. 19, no. 4, pp. 475–8, Apr. 2013.