



universität
wien

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Computational prediction of structural non-coding RNAs“

verfasst von / submitted by

Lorenz Perschy BSc.

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 834

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Molekulare Biologie / Molecular Biology

Betreut von / Supervisor:

Univ.-Prof. Dr. Robert Konrat

Mitbetreut von / Co-Supervisor:

Dipl.-Biol. Dr. Tanja Gesell

Abstract

While the “true” number of non-coding ribonucleic acids (ncRNA) in the human genome is still heavily debated, their biological role, which ranges from control of transcription and translation to splicing and epigenetic silencing, is undisputed. Due to their diverse regulatory function, which can be found in all domains of life, abnormal levels of ncRNAs have been linked to the development of cancer, Alzheimer and many other diseases. With the advances in sequencing technology and the ever growing amount of sequence data, *in silico* methods for the reliable identification of ncRNAs are a crucial prerequisite to find new ncRNA candidates and to decipher their role in the development of diseases; however the relatively high false positive rate (FPR) remains until today a big challenge in the prediction of ncRNAs. Therefore, the aim of this master’s thesis is to improve and update **SISSIZ**, an already established software for the prediction of ncRNAs and simulation of alignments: One of the main objectives was to facilitate the use of experimental probing data and to update new energy parameters by incorporating the latest version of the ViennaRNA package. As a result, it could be shown that experimental SHAPE (Selective 2'-hydroxyl acylation analyzed by primer extension) data possess great potential to aid the search for ncRNAs, most of all with the advent of high throughput probing methods. Additionally, the SCI (Structural Conservation Index), which is used by the competitor ncRNA gene finder **RNAz**, appeared as a promising extension to **SISSIZ** as indicated in the benchmark tests. Furthermore, a large part of the work focused on sequence and alignment simulation. In this context, **SISSIZ** could also be improved by adding new functions that allow simulations to be parallelized and reproduced. Last but not least, **SISSIZ** was applied in two genomic screens on the *Drosophila* X chromosome and the HIV (Human Immunodeficiency Virus). In the later case SHAPE data was available which could be demonstrated to improve the prediction of ncRNAs. The C source code of the new **SISSIZ** version will be available under the free GNU General public license alike the current version of the program.

Zusammenfassung

Während die "wahre" Anzahl nicht kodierender Ribonukleinsäure (Englisch: ncRNA) im menschlichen Genom noch immer stark umstritten ist, ist ihre biologische Rolle, die von Transkriptions- und Translationskontrolle zu Splicing und epigenetischem Silencing reicht, unumstritten. Als Folge ihrer vielfältigen regulatorischen Funktionen, die in allen Domänen des Lebens anzutreffen sind, stehen veränderte ncRNA Level im Zusammenhang mit der Entstehung von Krebs, Alzheimer und vielen anderen Krankheiten. Mit den Fortschritten in der Sequenzierungstechnologie und der stetig wachsenden Anzahl an Sequenzierungsdaten, sind *in silico* Methoden für die verlässliche Identifizierung von ncRNA eine entscheidende Voraussetzung, um neue ncRNA Kandidaten zu finden und ihre Rolle in der Entstehung von Krankheiten zu entschlüsseln. Allerdings stellt die relativ hohe Falsch Positiv Rate (FPR) bis heute eine große Herausforderung in der Vorhersage von ncRNA dar. Deshalb ist das Ziel dieser Masterarbeit **SISSIz**, ein bereits etabliertes Programm zur Vorhersage von ncRNA und zur Simulation von Alignments, zu verbessern und auf den neuesten Stand zu bringen: Eines der Hauptziele war es, die Verwendung von experimentellen Daten und neuer Engegieparameter durch die Integration der neusten Version des ViennaRNA packages zu ermöglichen. So konnte gezeigt werden, dass experimentelle SHAPE (Englisch: Selective 2'-hydroxyl acylation analyzed by primer extension) Daten ein großes Potential besitzen, die Suche nach ncRNA zu unterstützen. Dies ist vor allem dem Aufkommen von probing Methoden mit hohen Durchsatz zu verdanken ist. Zusätzlich erscheint die Erweiterung **SISSIz** mit dem SCI (Englisch: Structural Conservation Index), der vom konkurrierenden ncRNA gene finder RNAz verwendet wird, in den Benchmark Tests als vielversprechend. Des Weiteren war ein großer Teil dieser Arbeit der Sequenz und Alignment Simulation zugeschrieben. In diesem Zusammenhang konnte **SISSIz** ebenfalls durch neue Funktionen, die die Parallelisierung und Reproduktion von Simulationen erlauben, verbessert werden. Zu guter Letzt, wurde **SISSIz** an zwei genomischen Untersuchungen am Drosophila X Chromosome and am HIV (Englisch: Human Immunodeficiency Virus) angewendet. Im letzteren Fall waren SHAPE Daten verfügbar, die die Vorhersage von ncRNA verbessern konnten. Der Quelltext der neuen **SISSIz** Version wird, genauso wie die derzeitige Version des Programms, unter der freien GNU General public license verfügbar sein.

Acknowledgements

I would like to thank my supervisor PhD Tanja Gesell, for encouraging me to write my master's thesis in the field of bioinformatics and giving me the chance to work on her program. Coming from the field of molecular biology, the endeavor appeared challenging in the beginning, but thanks to her support everything worked out nicely. I will be forever grateful to her to have had the opportunity to work on this master's thesis.

Furthermore, I want to address my gratitude to Professor Robert Konrat, who let me join his group and allowed me to work on a project seemingly so far away from NMR.

To Professor Thomas Rattei, who initially sparked my interest in bioinformatics during the bachelor thesis.

I am also thankful for PhD Martin Smith's advice on SISSIZ during his visit to Vienna and for PhD Marco Sealey's biological view on the topic.

Last but not least, I would like to thank my girlfriend Isabella, for her support during the good and hard times of my thesis and my family for enabling me to pursue my dreams.

Contents

1	Introduction	11
1.1	Goal and Structure	11
1.2	SISSIZ	11
1.3	nRNAs	12
1.3.1	Definition	12
1.3.2	RNA structure	12
1.3.3	ncRNA types	13
1.3.4	Abundance	13
1.3.5	Prediction	14
2	Simulation	15
2.1	Introduction	15
2.1.1	Single sequence shuffling	17
2.1.2	Alignment shuffling and simulation	19
2.2	Upgrading SISSIZ's Pseudo Random Number Generator	23
2.2.1	Motivations and Background for updating the PRNG	23
2.2.2	Methods	24
2.3	Benchmark of simulation runtime	25
2.3.1	Setup	25
2.3.2	Results	26
2.3.3	Conclusion	29
2.4	Benchmark of simulation performance	29
2.4.1	Methods	29
2.4.2	Results	29
2.4.3	Discussion	30

3	RNA gene finding	33
3.1	Introduction	33
3.1.1	SISSIZ as an RNA genefinder	33
3.1.2	RNAz	35
3.1.3	RNAalifold	36
3.1.4	SHAPE	39
3.1.5	ROC curves	41
3.2	Upgrading Vienna RNA package's RNAalifold	45
3.2.1	Motivations and Background	45
3.2.2	Runtime behavior	45
3.3	Benchmarking	49
3.3.1	SISSIZ old VS new	49
3.3.2	The SCI and SISSIZ	55
3.3.3	The impact of alignment programs	59
3.3.4	SISSIZ with SHAPE	61
4	Application: Screening for non-coding RNAs	65
4.1	Drosophila Screen	65
4.1.1	Introduction	65
4.1.2	Methods	66
4.1.3	Results	68
4.1.4	Conclusion	72
4.2	HIV Screen	73
4.2.1	Introduction	73
4.2.2	Methods	75
4.2.3	Results and Discussion	76
5	Outlook	81
5.1	Simulation	81
5.2	RNA gene finding	82
5.3	Application: Screening for non-coding RNAs	82
6	Appendix	83

List of Figures

2.1	Swapping algorithm	18
2.2	Eulerian Walk	19
2.3	SISSIZ simulation flowchart	22
2.4	Shuffling performance in dependence of alignment length	28
2.5	Shuffling performance in dependence of alignment size	28
2.6	Alignment statistics after simulation - RNA input alignments	30
2.7	Alignment statistics after simulation - DNA input alignments for multiperm . . .	31
3.1	SISSIZ RNA gene finder workflow	34
3.2	SHAPE reaction	40
3.3	The ROC curve	43
3.4	RNAalifold runtime	47
3.5	SISSIZ runtime	48
3.6	Benchmark: SISSIZ old vs. new - multiperm as negative control	52
3.7	Benchmark: SISSIZ old vs. new - SISSIZ simulation as negative control	53
3.8	Benchmark: SISSIZ old vs. new - AUC	54
3.9	Relationship between z-score and SCI	55
3.10	Benchmark: SISSIZ with SCI	58
3.11	Benchmark: The impact of alignment programs on SISSIZ	60
3.12	Benchmark: SISSIZ with SHAPE data	64
4.1	Pipeline of ncRNA screen	67
4.2	Drosophila screen - Venn Diagrams	69
4.3	Drosophila screen - Statistics of common annotated predictions	70
4.4	Drosophila screen - Recovered annotation types	71
4.5	Statistics of annotation types	71
4.6	RRE Secondary Structure	74
4.7	HIV frameshift stimulating signal	75

4.8	HIV-screen 1	78
4.9	HIV-screen 2	79

List of Tables

2.1	Substitution Models	20
2.2	Overview of added command line options	25
2.3	DI-simulation runtime	27
2.4	MONO-simulation runtime	27
3.1	Confusion Matrix	42
3.2	RNAalifold changelog	46
3.3	Sensitivity at fixed FPR	56
3.4	Predictor thresholds at fixed FPR	57
3.5	RFAM families used in SHAPE benchmark	61
4.1	Screen Thresholds	68
6.1	RFAM families used in Benchmark-Part 1	84
6.2	RFAM families used in Benchmark-Part 2	85
6.3	RFAM families used in Benchmark-Part 3	86

Chapter 1

Introduction

1.1 Goal and Structure

The aim of this master's thesis is to update and improve the program **SISSIZ** (Gesell and Washietl (2008), Smith et al. (2013)) by adding new functions to its simulation module and by incorporating the newest version of the Vienna RNA package (Lorenz et al., 2011) in order to facilitate the use of experimental probing data. The improvements are anticipated to reduce the False Positive Rate which constitutes until today a big challenge in the prediction of ncRNAs. Eventually, the updated version of **SISSIZ** is evaluated in two genomic screens, one of which is aided by experimental data. To this end, the thesis is partitioned into the main chapters of **Simulation** (2), **RNA gene finding** (3) and **Application: Screening for non-coding RNAs** (4). The last chapter (5) provides an outlook on the topic.

1.2 SISSIZ

To start with, an introduction to **SISSIZ** will be given. The program is based on two components: the simulation module and the thermodynamic module. While the simulation module constitutes a special case of the **SISSI** (Simulating Site-Specific Interactions, Gesell and von Haeseler (2006)) framework, the thermodynamic module is represented by **RNAalifold** (Hofacker et al. (2002), Bernhart et al. (2008)). Both modules, which will be explained more thoroughly in the following chapters, combined turn **SISSIZ** into a tool for the detection of functional ncRNAs (non coding ribonucleic acids) in alignments. The underlying hypothesis for the prediction is made by defining structural conservation of the consensus secondary structure on the level of Minimum Free Folding Energy (MFE) as the hallmark of functional ncRNAs. Therefore, a z-score is calculated based on the MFE of a given input alignment and simulated null alignments in which the sequences evolved independently from structural constraints. To this end, the rel-

evance of randomized sequences and alignments arises and is dedicated the entire chapter 2. A background on the thermodynamic module of SISSIZ will be given in chapter 3.

1.3 nRNAs

1.3.1 Definition

But what are actually ncRNAs? The most general answer is that ncRNAs are defined as RNAs that are not translated into proteins but still have a function. The central dogma of molecular biology dictates the DNA is first transcribed to mRNA (messenger RNA) which is then (usually) translated to a chain of aminoacids referred to as proteins (Crick, 1958). The chains of aminoacids (proteins) fold into (mostly) well defined three dimensional structures that account for their function. However, if the translation process is absent, it is still possible for these (nc)RNA to serve a function. As a result, the genomic region, harboring a ncRNA, is often referred to as RNA gene, since the most general definition of a gene is given as a functional DNA element. In analogy to proteins, the structure of the RNA is greatly responsible for its function too.

1.3.2 RNA structure

RNAs are polymers of nucleotides with the four bases Adenine, Guanine, Cytosine and Uracil. While Adenine and Uracil can form base pairs via two hydrogen bonds, Guanine and Cytosine base pairing is established via three hydrogen bonds. Additionally, so called Wobble base pairing is possible between Guanine and Uracil. Via these base pairing interactions it is possible for RNA to fold into a defined structure. In this context the secondary structure is often used as it enables the representation of the intricate three dimensional molecular RNA structures (tertiary structure) in a two dimensional space. The secondary structure allows the description of stem loops, interior loops, hairpin loops, multiloops and pseudo knots, from which functional inferences can be made. Nussinov et al. (1978) was among the first to develop an algorithm for the prediction of RNA secondary structures by calculating the maximum number of matching base pairs. Zuker and Sankoff (1984) extended this approach to find the best secondary structure by energy minimization. While the first algorithms were based on finding the optimum secondary structure for a single sequences more recent programs use alignment based approaches to predict a consensus structure and MFE, respectively. For example: *RNAalifold*, which is also used by SISSIZ for the calculation of the MFE. This topic will be discribed in more detail in chapter 3.1.3.

1.3.3 ncRNA types

The probably best known and most abundant ncRNAs by mass and number comprise ribosomal RNAs (rRNA) and transfer RNAs (tRNAs). In the ribosome rRNA catalyzes the peptidyl transfer reaction adding a new aminoacid to the growing polypeptid chain. Thus, RNA can catalyze reactions like an enzyme which is why catalytically active rRNAs are also called (ribozymes). But also the class of snRNAs (small nuclear RNAs) are at the active site in spliceosomes which are crucial for the maturation of pre-mRNA (precursor messenger RNA). The discovery of these and other catalytically active RNAs has supported the hypothesis of the RNA world (Gilbert, 1986) according to which the first forms of life utilized RNA to store information and to enable biochemical reactions. The typical cloverleaf secondary structure of the tRNA, on the other hand, is required for the correct interpretation of the genetic code. However, many more classes of ncRNA exist: For example miRNA (microRNA) (Bartel, 2004), siRNA (small interfering RNA) (Agrawal et al., 2003) snoRNA (small nucleolar RNA) (Tollervey and Kiss, 1997) and lncRNAs (long non-coding RNAs) (Rinn and Chang, 2012). Both miRNA and siRNA form hairpin structures in their primary transcripts which are processed to approximately 20 nt long single stranded RNAs that are complementary to specific mRNAs. Therefore, they can regulate gene expression by binding to mRNAs, which can either lead to translational inhibition or degradation of the mRNA. snoRNA are involved in the modification of rRNAs and tRNAs (pseudouridylation) and methylation. The role of lncRNAs is with few exceptions still largely illusive. For example, the XIST long non coding RNA is necessary to inactivate one of the two X chromosomes during the female development as a measure of dosage compensation. Managadze et al. (2011) could find a weak correlation between the level expression and the conservation in lincRNA (long intergenic non coding RNA) which raised the question to what extent ncRNAs are functional. Given their broad biological role it is not surprising that ncRNAs are involved in the development of many diseases, like cancer, Alzheimer's disease (Esteller, 2011).

1.3.4 Abundance

As a result, there is still a great debate on how much of the genome, especially the human genome, is functional and how much is simply "junk" (Palazzo and Lee, 2015). In the sense of ncRNA prediction the fundamental question on which the community appears to be divided regards the definition of functional ncRNAs. According to the ENCODE (Encyclopedia of DNA Elements) project consortium (Dunham et al., 2012) about 62% of the human genome is covered by various RNA types and 80.4% of the genome is considered functional. Mattick and Dinger (2013) argues that level of transcription is a better indicator for functional ncRNAs

than sequence conservation, thus supporting the findings of the ENCODE project. Smith et al. (2013) predicted over 4 million evolutionary conserved RNA structures in alignments of 35 mammals using structural conservation as measure to identify novel ncRNAs. This approach has been criticized by Eddy (2014) for the high FDR (False Discovery Rate). On the other hand, Eddy (2012) opposes the conclusion drawn by the ENCODE project even more strongly and argues that the C-value paradox and mutational load contradict the hypothesis that most of the human genome is functional. The fact that the complexity of organisms does not correlate with genome size (C-Value paradox) is seen as an indicator that most of the genome has no function. Additionally, the number of mutations is deemed deleterious if indeed such a great proportion of the genome was functional. Graur et al. (2013) is especially criticizing ENCODE's low specificity and literally calls it a hype for mass media. The argument about the true number of ncRNAs will likely continue for a long time until a consensus on the question of functionality is reached.

1.3.5 Prediction

Now that the biological relevance has been illustrated, the most commonly used programs for the prediction of ncRNA are listed in the following:

- **Alifoldz** (Washietl and Hofacker, 2004) uses, similar to **SISSIZ**, a z-score as a predictor for ncRNAs in alignments but with a different background model which does not maintain the di-nucleotide content. Chapter 2 will discuss the relevance of the background models in more detail.
- **RNAz** (Gruber et al. (2010), Washietl et al. (2005)) is an alignment based SVM (Support Vector Machine) that identifies ncRNAs based on a structural conservation index (SCI), a base-composition derived z-score and the Shannon entropy of the alignment. In chapter 3 a more thorough introduction to **RNAz** will be given.
- **EvoFold** (Pedersen et al., 2006) uses an alignment and a phylogentic tree as an input. In contrast to **SISSIZ** it is based on two SCFG instead of a thermodynamic model. It can be used for both, the prediction of secondary structure and the predictions of ncRNAs where the fps (folding potential score) serves as a predictor for ncRNAs.
- **CMfinder** (Yao et al., 2006) uses a Bayesian network to combine a covariance and a thermodynamic model which can be used for the identification of conserved RNA motifs.

Chapter 2

Simulation

2.1 Introduction

In every lab experiment it is crucial to conduct negative controls to check the validity of the results. The same applies in bioinformatic screens in order to estimate the "trustworthiness" of a prediction. To be more precise negative controls are needed to determine the False Positive Rate (FPR) or Type 1 error, which is the probability of incorrectly rejecting H_0 (the null hypothesis) even though H_0 is true. In the context of ncRNA gene finding it is necessary to estimate whether the properties of the input sequence/alignment differs from a randomized control (null distribution). Chapter 3.1.5 will describe the FPR in context of the Receiver Operating Characteristic (ROC) curve more precisely. For now, the chapter focuses on the generation of such negative control datasets for single sequences and alignments which should ideally possess the same properties but not the signal in question. At first glance it may seem trivial to create these datasets by simple permutation, but in the end the complexity of the problem becomes substantial if the following sequence statistics are to be maintained:

- The **mono-nucleotide content** as the relative frequency of every nucleotide in a sequence or alignment.
- The **di-nucleotide content** is especially important in RNA secondary prediction since neighboring bases contribute stacking energies when paired (Crothers and Zimm (1964), Workman and Krogh (1999) and Washietl and Hofacker (2004)). Since the order of base pairs influences the MFE (Minimum Free Energy) it is necessary to retain the di-nucleotide frequencies in the simulated datasets for thermodynamic approaches. It could be shown by Gesell and Washietl (2008) that simulating alignments with di-nucleotide control lowered the FPR in their thermodynamic based ncRNA gene-finder. Let x_k be the site in a sequence x with length l at $k = (1, \dots, l)$. Then the di-nucleotide content is

defined as the relative frequency of every doublet $\{x_k, x_{k+1}\}$ with $k \in \{1, l-1\}$ (overlapping counts). In the case of alignments the relative doublet frequencies are counted with respect to every sequence in the alignment. Note that this is the general definition of the di-nucleotide content. In **SISSIZ** (Gesell and Washietl, 2008) only doublets (di-nucleotides) without gaps are counted.

- The **tri-nucleotide content** is defined as the relative frequency of every triplet $\{x_k, x_{k+1}, x_{k+2}\}$ with $k \in \{1, l-2\}$ (in general this can be extended to any k-mer size). It becomes relevant when the codon usage must be preserved as not to destroy the codon bias which can vary quite significantly from organism to organism (Hershberg and Petrov, 2008). Thus it could be potentially useful to search for ncRNAs in coding regions.
- In an alignment A the **Mean Pairwise Identity** (MPI) is described as in Equation 2.1

$$\frac{1}{l} \cdot \sum_{\substack{x, y \in A \\ x \neq y}} (l - h(x, y)) \cdot \frac{2}{n \cdot (n-1)} \quad (2.1)$$

with n sequences, length l and $h(x, y)$ as the Hamming distance between two sequences x and $y \in A$. The factor $\frac{2}{n \cdot (n-1)}$ stems from the Gaussian formula and corresponds to the number of Hamming sequence comparisons. Thus the MPI ranges from 0 to 1: While a MPI of 0 would theoretically indicate that all sequences in the alignment do not share a single base in any alignment column, a MPI of 1 would denote an alignment with 100% identical sequences.

- The **(Local) Conservation pattern** which is defined as the frequency of the most abundant nucleotide (=consensus nucleotide) in every column of an alignment divided by the number of sequences in the alignment. As a result, every column in the alignment is associated with a certain degree of conservation.
- The **GC content**: The energy contribution of $G \equiv C$ base pairs is higher than for $A = U$ base pairs, as the first is formed by three hydrogen bonds and the latter by only two. The GC content is defined as the relative number of G and C nucleotides in the sequence or alignment.

As the simulation of sequences and alignments is often not a simple task, sampling from genomic data seems like a viable option. However, this approach is limited by the lack of "negative" annotations as it is rather difficult to prove that a genomic region, for example, truly has no function (Varadarajan et al., 2008). And even if such annotation existed extracting sequences

with the same statistics as in the positive control dataset would be difficult. Therefore, simulation appears to be the more favorable option after all. But also in the context of homology search negative controls are important, as pointed in Paul Gardener’s online article ”On the importance of controls, even when the tools are computational”¹ The next sections will discuss methods to create randomized sequences and alignments while maintaining the mentioned statistics.

2.1.1 Single sequence shuffling

Maintaining mono-nucleotide content

When no higher statistics but the mono-nucleotide content are to be preserved, the **Fisher-Yates/Knuth shuffle** (Fisher and Yates (1948) and Knuth (1969)) offers a simple method for unbiased sequence permutation: The algorithm can be described as a recursion as follows: Let x be a sequence of length l . Pick a random number $r \in \{1, l\}$. Swap x_r with x_l and recurse on $x_l - 1$ until $l = 1$.

```
shuffle(x) {
  l=length(x)
  r=1+random()%(l+1)
  temp=x[l]
  x[l]=x[r]
  x[r]=temp
  if (l>1)
    shuffle(x[1:l-1])
}
```

For this and in all the other following methods, it is of course crucial to use a ”good” random number generator in order to obtain reasonably permuted sequences. This holds true for all simulation programs and will be discussed more thoroughly in the course of chapter 2.2.

Maintaining di-nucleotide content

First order **Markov chains** can be used to simulate sequences with approximately the same di-nucleotide content (Naturally, Markov chains can also be used to model sequences with the same mono-nucleotide content). To this end, the transition probabilities $t_{a,b}$ that a nucleotide a is followed by nucleotide b , are derived from the doublet frequencies of the input sequence. Then a new sequence x is created from scratch. The likelihood at which a new nucleotide x_i is added depends only on the probability of the current nucleotide x_i . This transition probability $t_{x_i, x_{i+1}}$

¹<http://rnainformation.blogspot.co.at/2017/07/on-importance-of-controls-even-when.html?t=1&cn=ZmxleGlibGVfcw==&iid=e4cfedc606b745288b3d1d05bff64a22&uid=2787671532&nid=244+281546760> (14.9.2017)

can be written as the conditional probability $P(x_i + 1, x_i) = P(x_i + 1 | x_i)$. By this method a plethora of unique sequences can be simulated. However, the di-nucleotide content remains only approximately the same.

Another method to generate sequences with the same di-nucleotide content was presented by Kandel et al. (1996) as the **swapping algorithm**. In essence, it states that two nucleotides in a sequence can be exchanged if they are flanked by the same nucleotides. To be more concise: Pick two random positions $i, j \in \{1, l\} | i \neq j$ in a sequence x and swap x_i and x_j if $x_{i-1} = x_{j-1}$ and $x_{i+1} = x_{j+1}$. Repeat the process n times. Figure 2.1 shows a graphical representation of the process. The algorithm can easily be extended to maintain triplet or generally any k -let frequency by comparing all $x_{i-h} = x_{j-h}$ and $x_{i+h} = x_{j+h}$ with $h \in \{1, \dots, k\}$. However, a drawback of the algorithm is that the permuted sequences are uniform only approximately and that there appears to be no general rule to how many cycles n are necessary to obtain a sufficient degree of permutation. The algorithm implemented by Workman and Krogh (1999) uses $n = l \cdot 10$

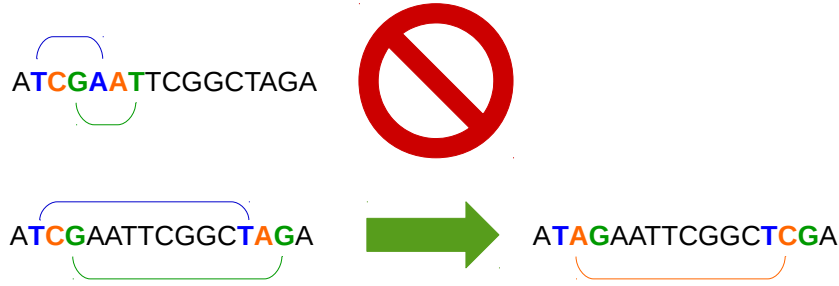


Figure 2.1: **Swapping algorithm:** The swapping algorithm randomly picks two positions x_i and x_j (orange) in a sequence x and swaps them if $x_{i-1} = x_{j-1}$ (blue) and $x_{i+1} = x_{j+1}$ (green). In the first case the condition that the flanking nucleotides are the same is not true; thus no position exchange takes place. In the second case the neighboring sequences are identical and the nucleotides at sites i and j are swapped. As a result, the di-nucleotide content is preserved.

The most elegant solution to the shuffling problem constitutes the **Euler algorithm** which constructs all sequences uniformly. Therein, the di-nucleotide content of the input sequence is represent in a directed multigraph with one vertex per nucleotide. To create a new sequence with the same di-nucleotide content it is now necessary to find a path (Eulerian walk) through the multigraph that uses all edges exactly once. While Altschul and Erickson (1985) were among the first to implement an algorithm to find such an Eulerian walk to maintain the doublet and triplet frequencies, Jiang et al. (2008) created a more efficient algorithm that preserves any given k -let count and implemented it in the program **ushuffle**. Figure 2.2 illustrates how the di-nucleotide content of a sequence can be represented in directed multigraph.

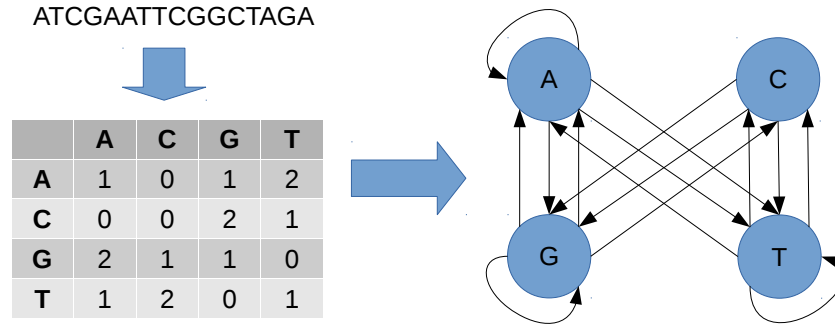


Figure 2.2: **Eulerian Walk:** The di-nucleotide counts of the sequence are represented both in a matrix and a directed multigraph. In order to create a new sequence with the same di-nucleotide a path (Eulerian walk) through the multigraph, that crosses every edge exactly once, has to be found.

2.1.2 Alignment shuffling and simulation

Maintaining mono-nucleotide content

With alignments the complexity of the simulation increases as the aligned sequences cannot simply be shuffled independently without causing complete destruction of the alignment. One solution to the problem can be found by shuffling the order of the alignment columns. **RNAz** 's perl script `rnazRandomizeAln.pl` Washietl et al. (2005) extends this approach by also maintaining local conservation and gap patterns. Even though, this method maintains the MPI and mono nucleotide frequencies exactly, it still fails to preserve the di nucleotide content.

Maintaining di-nucleotide content

As an extension of the swapping algorithm for single sequences Babak et al. (2007) implemented `shuffle-pair.pl` for the randomization of **pairwise sequence alignments** maintaining the di-nucleotide frequencies. In this implementation the swapping algorithm's permutation space is further reduced since sequence identity and gap patterns are also conserved. Most of all the limitation to randomize pairwise alignments only renders it impractical for most applications.

multiperm

The program `multiperm` (Anandam et al., 2009) takes advantage of methods established for single sequence shuffling, i.e. Eulerian algorithm, and employs them on alignments. In brief, the algorithm shuffles the consensus sequence of the alignment and randomly reintroduces the columns with matching conservation classes. As a result, the MPI and mono-nucleotide frequencies are maintained exactly while the local conservation patters and the di-nucleotide content are conserved approximately. Since the method relies on shuffling columns based on their consensus

sequence, it heavily depends on the degree of conservation in each column. Therefore **multiPerm** may perform well at high MPIs to conserve the di-nucleotide content but worse at low MPIs. Moreover, since only alignment columns are shuffled, possible column correlations arising due to covariation are not destroyed. Gruber et al. (2010) have therefore set a threshold for the shannon entropy of an alignment (see equation 3.5) according to which **multiPerm** is to be used at entropies below 0.5 and SISSIZ, which will be described in the following, at entropies above 0.5.

Sequence Evolution

Phylogenetic approaches for the simulation of sequences are commonly used to model the evolution of DNA/RNA (**Seq-Gen** Rambaut and Grassly (1997)) and proteins (**PSeq-Gen** Grassly et al. (1997)). **Seq-Gen** (Sequence Generator) has implemented three Markov substitution models which describe the process of sequence evolution: the HKY (Hasegawa et al., 1985), the F84 (Felsenstein and Churchill, 1996) and REV (general reversible process model, Yang (1994)). As visualized in table 2.1 the JC69 model (Jukes and Cantor, 1969) can be derived from the HKY model if the stationary nucleotide distributions π are set to 0.25 and $\alpha = \beta$, i.e. transitions and transversions are equally likely. Transitions are regarded as mutations from purines to purines and pyrimidines to pyrimidines. Transversions, on the other hand are changes from pyrimidines to purines and vice versa. The K80 model (Kimura, 1980) makes this distinction and corresponds to the HKY model with equal stationary distributions. The F81 model (Felsenstein, 1981) does not distinguish between transitions and transversions but allows to set specific stationary distributions. The TN93 (Tamura and Nei, 1993) model tweaks the HKY model by distinguishing by the kind of transition: α_1 refer to $A \leftrightarrow G$ and α_2 to $C \leftrightarrow T$ changes. Finally,

	A	C	G	T	A	C	G	T	A	C	G	T
	JC69				K80				HKY			
A	*	α	α	α	*	β	α	β	*	$\beta\pi_C$	$\alpha\pi_G$	$\beta\pi_T$
C	α	*	α	α	β	*	β	α	$\beta\pi_A$	*	$\beta\pi_G$	$\alpha\pi_T$
G	α	α	*	α	α	β	*	β	$\alpha\pi_A$	$\beta\pi_C$	*	$\beta\pi_T$
T	α	α	α	*	β	α	β	*	$\beta\pi_A$	$\alpha\pi_C$	$\beta\pi_G$	*
	TN93				F81				GTR			
A	*	$\beta\pi_C$	$\alpha_1\pi_G$	$\beta\pi_T$	*	π_C	π_G	π_T	*	$a\pi_C$	$b\pi_G$	$c\pi_T$
C	$\beta\pi_A$	*	$\beta\pi_G$	$\alpha_2\pi_T$	π_A	*	π_G	π_T	$a\pi_A$	*	$d\pi_G$	$e\pi_T$
G	$\alpha_1\pi_A$	$\beta\pi_C$	*	$\beta\pi_T$	π_A	π_C	*	π_T	$b\pi_A$	$d\pi_C$	*	$f\pi_T$
T	$\beta\pi_A$	$\alpha_2\pi_C$	$\beta\pi_G$	*	π_A	π_C	π_G	*	$c\pi_A$	$e\pi_C$	$f\pi_G$	*

Table 2.1: Substitution Models: The table (by Gesell (2009)) shows 6 substitution models: JC69 Jukes and Cantor (1969), K80 (Kimura, 1980), HKY (Hasegawa et al., 1985), TN93 (Tamura and Nei, 1993), F81 Felsenstein (1981) and the GTR (General Time Reversible) model (Rodriguez et al., 1990). The models are defined by the equilibrium parameters π and/or the substitution rate parameter(s) $\alpha, \beta, a, b, c, d, e, f$

the general time reversible model (GTR, Rodriguez et al. (1990)) has four equilibrium parameters and 6 substitution rate parameters which allow to adjust the rate for any kind of mutation while maintaining time reversibility.

SISSIZ

SISSIZ (Gesell and Washietl, 2008) extends on the phylogenetic approach to generate alignments with approximately the same mono and di-nucleotide content by employing an adapted version of the independent program **SISSI** (Simulating Site Specific Interactions, Gesell and von Haeseler (2006)). Furthermore, local conservation and gap patterns are retained. In order to maintain the di-nucleotide frequencies, a "neighborhood system" is used; therein the substitution probability Q_k of a site x_k in a sequence $x = (x_1, \dots, x_k, \dots, x_l)$ depends on its neighboring positions x_{k-1} and x_{k+1} . Thus Q_k has the size $|\check{A}|^3 \times |\check{A}|^3 = 64 \times 64$ with $\check{A} = \{A, U, G, C\}$. Since a substitution is only permissible at site x_k , Q_k contains only $|\check{A}|^4 = 256$ non-zero entries as defined by equation 2.2, where s_k is the triplet $\{x_{k-1}, x_k, x_{k+1}\}$ at site k in sequence x and y an arbitrary triplet $\{y_1, y_2, y_3\} \in \check{A}^3$.

$$Q_k(s_k, y) = f_k \cdot \begin{cases} r(s_k, y) \cdot \pi(y) & \text{if } h(s_k, y) = 1 \text{ and } x_k \neq y_2 \\ -\sum_{\substack{z \in \check{A}^3 \\ z \neq s_k}} Q_k(s_k, z) & \text{if } h(s_k, y) = 0 \\ 0 & \text{else} \end{cases} \quad (2.2)$$

The parameter $r(s_k, y)$ is set to the transition/transversion rate ratio κ which is determined by maximum likelihood on an independent HKY (Hasegawa et al., 1985) mono-nucleotide model. Additionally, site specific rates f_k are derived from the column-wise identities in order to maintain local conservation patterns. Q_k can be viewed as 16 4×4 sub-matrices as denoted in equation 2.3.

$$\begin{array}{cccc} & y_1 A y_3 & y_1 G y_3 & y_1 C y_3 & y_1 U y_3 \\ \begin{array}{c} y_1 A y_3 \\ y_1 G y_3 \\ y_1 C y_3 \\ y_1 U y_3 \end{array} & \begin{pmatrix} * & \pi_{y_1 C y_3} & \kappa \pi_{y_1 G y_3} & \pi_{y_1 U y_3} \\ \pi_{y_1 A y_3} & * & \pi_{y_1 G y_3} & \kappa \pi_{y_1 U y_3} \\ \kappa \pi_{y_1 A y_3} & \pi_{y_1 C y_3} & * & \pi_{y_1 U y_3} \\ \pi_{y_1 A y_3} & \kappa \pi_{y_1 C y_3} & \pi_{y_1 G y_3} & * \end{pmatrix} \end{array} \quad (2.3)$$

The stationary triplet frequency $\pi(y) = \pi(\alpha\beta\gamma)$ is derived from the counted average mono $\pi(\alpha)$ and doublet $\pi(\alpha\beta)$ and $\pi(\beta\gamma)$ frequencies of all sequences in the input alignment according to

equation 2.4 with $\alpha, \beta, \gamma \in \check{A} \wedge \pi(\beta) = \sum_{\alpha} \pi(\alpha\beta) = \sum_{\alpha} \pi(\beta\alpha)$.

$$\pi(\alpha\beta\gamma) = \frac{\pi(\alpha\beta) \cdot \pi(\beta\gamma)}{\pi(\beta)} \quad (2.4)$$

Then a distance matrix from all pairwise distances is used to construct a tree by applying a BIONJ (Gascuel, 1997) algorithm. Finally, a new sequences are simulated along the tree starting from an ancestral 1st order Markov sampled sequence yielding the simulated alignment. Thereby, both local and global MPIs are preserved. The sequences are simulated without gaps which are reintroduced in the same pattern as in the native alignment in the final step of the simulation. The algorithm is summarized in a flowchart in figure 2.3 by Gesell and Washietl (2008).

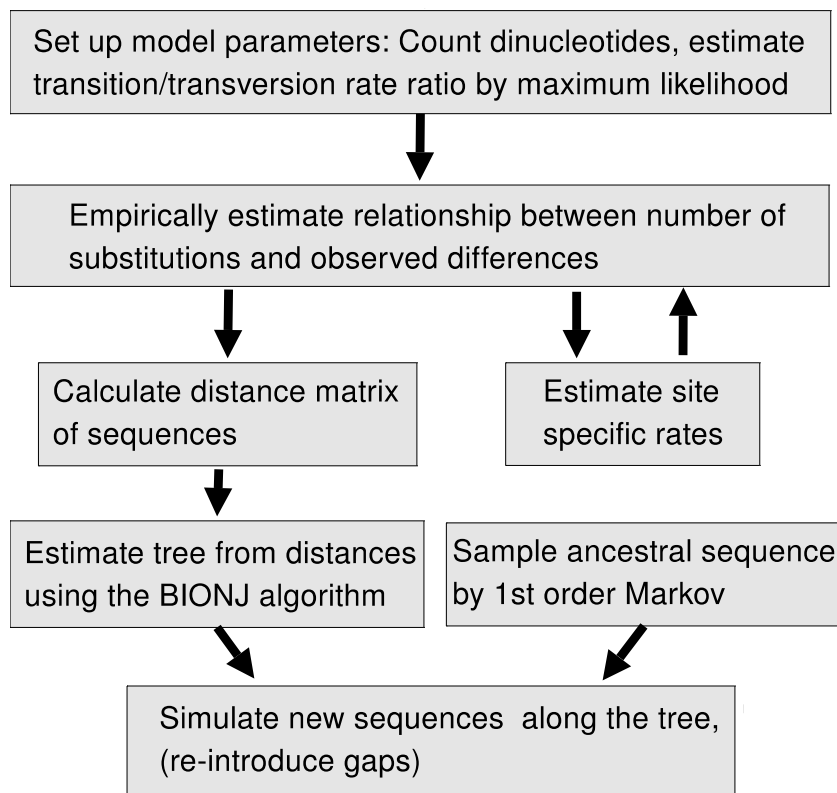


Figure 2.3: **SSISSi simulation flow chart** by Gesell and Washietl (2008): In the first step the di-nucleotide frequencies and the transition transversion rate ratio are derived from the input alignment. A distance matrix is then calculated from the site-specific rates by exponential regression and a tree calculated using the BIONJ algorithm (Gascuel, 1997). Eventually, the sequences for the alignment are simulated along the tree from an ancestral 1st order Markov sampled sequence.

2.2 Upgrading SSIz's Pseudo Random Number Generator

As trivial as the demand for random numbers may sound, as hard they are to generate. Most computer applications use PRNGs (Pseudo Random Number Generator) which produce a series of random numbers. However, the sequence of those numbers is anything but random since it is determined by certain seed values, which initialize the PRNG, thus the name Pseudo Random Number Generator. "True" random numbers can be generated by measuring radioactive decay, thermal noise or any other non-deterministic process. The disadvantage of "true" random number generators lies in their inherent slowness and difficult implementation requiring specialized hardware which renders them unsuitable for many applications. Furthermore, the properties of numbers generated by PRNGs are usually sufficient for most purposes. There are several tests available to analyze the quality of random number generators, for example the **DIEHARD** ² test suite which was developed by George Marsaglia or its successor the **Dieharder** ³ test suite by Robert G. Brown. Many system random number generators have been shown to be defective, including C's and Java's random number generator (L' Ecuyer and Simard, 2007). Therefore, it is always advisable to use a separate PRNG.

2.2.1 Motivations and Background for updating the PRNG

Random number generators play a crucial role in computational biology as they are fundamental to many applications, be it complex simulations in molecular dynamics or basic sequence shuffling (Click et al., 2011). The latest version of SSIz uses the 32 bit DKISS (Double Keep It Simple Stupid) based PRNG proposed by George Marsaglia⁴. As the current version of the DKISS was not optimized for simultaneous runs and its period length was just about 2^{124} SSIz which depends on a random number generator for alignment simulation was granted an PRNG upgrade.

The updated SSIz version now uses a 64 bit KISS version, with an increased period length of 2^{191} . The process of seeding the PRNG was also improved so that it no longer depends on the time of execution only, thus enabling simultaneous runs of SSIz. Furthermore, it is now possible to save used seed values and to use them for the reproduction of results. Despite the higher complexity of the new PRNG, SSIz's run time has even been slightly improved, though not significantly. In comparison with the original SSIz version, similar results on identical data sets could be obtained.

²<https://wayback.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/> and <http://www.jstatsoft.org/v07/i03/paper>

³<http://www.phy.duke.edu/~rgb/General/dieharder.php>

⁴<https://programmingpraxis.com/2010/10/05/george-marsaglias-random-number-generators/> (02.03.2017)

2.2.2 Methods

SISSIZ uses a 32 bit DkISS algorithm which takes advantage of three independent random number generators:

- Multiply-With-Carry (MWC)
- Xorshift (XSH)
- Congruential (CNG)

The period length (length of repetition free sequence) of this 32 bit DkISS is about 2^{124} . The algorithm passes all tests of the DIEHARD test suite and also the BigCrunch Test (L'Ecuyer and Simard, 2007), as the combination of the three PRNGs cancels out their individual weaknesses. The result is a fast high quality PRNG with fair period length. DkISS source code as implemented in SISSIZ (v.2.0):

```
1 /* Seed variables */
2 unsigned int x=seed;y=102;z=12;w=34535;
3
4 unsigned int kiss(void){
5     x = x * 69069 + 1;
6     y ^= y << 13; y ^= y >> 17; y ^= y << 5;
7     k = (z >> 2) + (w >> 3) + (carry >> 2);
8     m = w + w + z + carry; z = w; w = m; carry = k >> 30;
9     return x+y+z;
10 }
```

David Jones developed a new version of the KISS algorithm, the JLKISS, to support 64 bit integers as seed variables (Jones, 2010). It has an increased period length of 2^{191} and passes all of the above statistical tests for random number generators as well. JLKISS source code as implemented in the new SISSIZ version:

```
1 /* Seed variables */
2 unsigned long long x = 123456789123ULL,y = 987654321987ULL;
3 unsigned int z = 43219876,c = 6543217;
4
5 unsigned int JLKISS(void){
6     unsigned long long t;
7     x = 1490024343005336237ULL * x + 123456789;
8     y ^= y << 21; y ^= y >> 17; y ^= y << 30;
9     t = 4294584393ULL * z + c; c = t >> 32; z = t;
10    return (unsigned int)(x>>32) + (unsigned int)y + z;
11 }
```

Additionally, the flaw in seeding the KISS PRNG in **SISSIZ** was taken care of. Previously, just one seed variable was initialized with every program call using the system time. This can be problematic if multiple processes of **SISSIZ** were started simultaneously, potentially causing identical runs to be submitted. The new JLKISS PRNG offers the option to not only seed with the system time but also the process ID which is stored to a second, independent seed variable (command line flag **-y**). (Table 2.2 gives an overview on all new command line options.) Other seeding alternatives were not exploited to ensure compatibility over all platforms. Furthermore, the command line flag **-z** was added to allow printing of the seed values used by **SISSIZ** to a *seed.txt* file. The file is automatically appended with every call of **SISSIZ**. Conversely, (a) seed value(s) can be read by the command line flag **--read_seeds={seed1(,seed2)}**. As a result, the reproduction of results is now possible.

Command line parameter	Description
-y, --seed_with_pID	Seed with process ID. Default: OFF
-z, --print_seeds	Print seed values to seed.txt . Default: OFF
--read_seeds	set seed value(s) = {seed1(,seed2)}

Table 2.2: **Overview of added command line options**

Protocol

SISSIZ's DKISS was replaced by David Jones' JLKISS. Additionally, the initialization function **start_kiss()** was changed to allow seeding with system time and process identity (**time()** and **getpid()**, respectively). The obsolete functions **restart_kiss()** and **kiss_state()** were discarded as they are not used in any other part of the **SISSIZ** code.

2.3 Benchmark of simulation runtime

In order to evaluate the impact of the new PRNG four benchmark tests were performed. Therefore, the runtime was measured in dependence of alignment length and size.

2.3.1 Setup

In the first benchmark test, **SISSIZ**'s *simulation only* (command line parameter: **-s**) mode was used in order to measure mono and di nucleotide alignment simulation performance. An HIV alignment (Lavender et al., 2015a) consisting of 3 sequences and a length of about 10,000 nucleotides served as an input file. Number of repetitions per alignment file: 100.

In the second benchmark test, **SISSIZ** was used on the files `genomic.maf` and `rRNA.aln`⁵ to compare the resulting z-scores in both mono and di nucleotide mode. Number of repetitions per alignment file: 100.

In the third benchmark test, the dependence of alignment length on shuffling performance was analyzed using *simulation only* (command line parameter: `-s`) mode. The HIV alignment of the first benchmark test, was divided into sub-alignments, resulting in alignment lengths from 60 to 2400 nucleotides in 60 nt steps. Number of repetitions per alignment file: 100.

In the fourth benchmark test, the dependence of alignment size, in terms of number of sequences, on shuffling performance was analyzed using *simulation only* (command line parameter: `-s`) mode. Therefore, alignments of differing numbers of HIV sequences were created using `clustalo` (Sievers et al., 2011). The resulting alignment files consisted of 5, 10, 15, 20, 25 and 30 sequences, with sequence lengths of about 9000 nucleotides. Number of repetitions per alignment file: 100.

Scheme for command line calls in the first, third and fourth benchmark test:

```
SISSIZ <alignment_file> -s (--mono)
SISSIZ_beta <alignment_file> -s (--mono)
SISSIZ_beta <alignment_file> -s (--mono) -y
```

Scheme for command line calls in the second benchmark test:

```
SISSIZ <alignment_file> (--mono)
SISSIZ_beta <alignment_file> (--mono)
SISSIZ_beta <alignment_file> (--mono) -y
```

2.3.2 Results

The first benchmark showed that **SISSIZ** with `JLKISS` runs slightly faster than **SISSIZ** with `DKISS`, but not significantly. Interestingly, there is no significant difference in runtime when the process ID was additionally used for seeding with **SISSIZ** `JLKISS` (Table:2.3). Similar results could be obtained when **SISSIZ** was used in the mono-shuffling mode (Table:2.4). Naturally, the runtime is faster than in the di-shuffling mode. Therefore, no performance drawbacks are to be expected from using **SISSIZ** with `JLKISS`, even with enhanced seeding.

The second benchmark was indented to compare the calculated z-scores of **SISSIZ** for two small alignment files. The first alignment file comprised 6 genomic sequences. All three **SISSIZ**

⁵from **SISSIZ**'s source code folder `examples/`

SISSIZ	DKISS	JLKISS	JLKISS - wo process ID
Period length	2^{124}	2^{191}	2^{191}
Run time (s)	8.186 ± 0.191	8.085 ± 0.192	8.079 ± 0.193
GENOMIC run (z-score)	-0.05 ± 0.10	-0.04 ± 0.11	-0.06 ± 0.10
rRNA run (z-score)	-2.60 ± 0.25	-2.56 ± 0.26	-2.57 ± 0.26

Table 2.3: **DI-simulation:** The table compares three **SISSIZ** configurations (DKISS, JLKISS with and without seeding with process ID). The average runtime was calculated from **SISSIZ**'s simulation speed on a 3 sequence HIV alignment of about 9000 nucleotide length from 100 runs. The average z-scores were calculated from running **SISSIZ** 100 times on a genomic and ribosomal RNA alignment, respectively.

SISSIZ	DKISS	JLKISS	JLKISS - wo process ID
Period length	2^{124}	2^{191}	2^{191}
Run time (s)	5.697 ± 0.220	5.542 ± 0.190	5.565 ± 0.200
GENOMIC run (z-score)	-0.28 ± 0.12	-0.30 ± 0.12	-0.29 ± 0.11
rRNA run (z-score)	-2.74 ± 0.26	-2.71 ± 0.24	-2.74 ± 0.27

Table 2.4: **MONO-simulation:** The table compares three **SISSIZ** configurations (DKISS, JLKISS with and without seeding with process ID). The average runtime was calculated from **SISSIZ**'s simulation speed on a 3 sequence HIV alignment of about 9000 nucleotide length from 100 runs. The average z-scores were calculated from running **SISSIZ** 100 times on a genomic and ribosomal RNA alignment, respectively.

runs came to similar results (-0.05 VS -0.04 VS -0.06) with almost identical standard deviations (Table:2.3). The second alignment file contained 7 ribosomal RNA sequences. Also this time, the z-scores of the three versions converge (2.60 VS 2.56 VS 2.57). When **SISSIZ** is set to the mono-nucleotide mode, the z-scores of the two **SISSIZ** version remain coherent too (Table:2.4). The z-scores are higher in the mono-nucleotide mode compared to the di-nucleotide mode, as the simulation process is less restrictive. Since both versions come to the same result in terms of significance the JLKISS version of **SISSIZ** works appropriately.

The third benchmark shows the linear correlation between alignment length and runtime (Fig.:2.4) for **SISSIZ** simulations. The runtime behavior for an alignment of length n is $O(n)$. When **SISSIZ** is run in simulate mode, as in this benchmark the effect of the new PRNG on runtime can be seen more clearly as it is not distorted by RNAalifold. On average the runtime with JLKISS is faster than with DKISS, although not significantly.

The fourth and last benchmark shows the linear effect of alignment size on simulation runtime (Fig.:2.5). Thus, the runtime behavior for alignments of n sequences is $O(n)$ too.

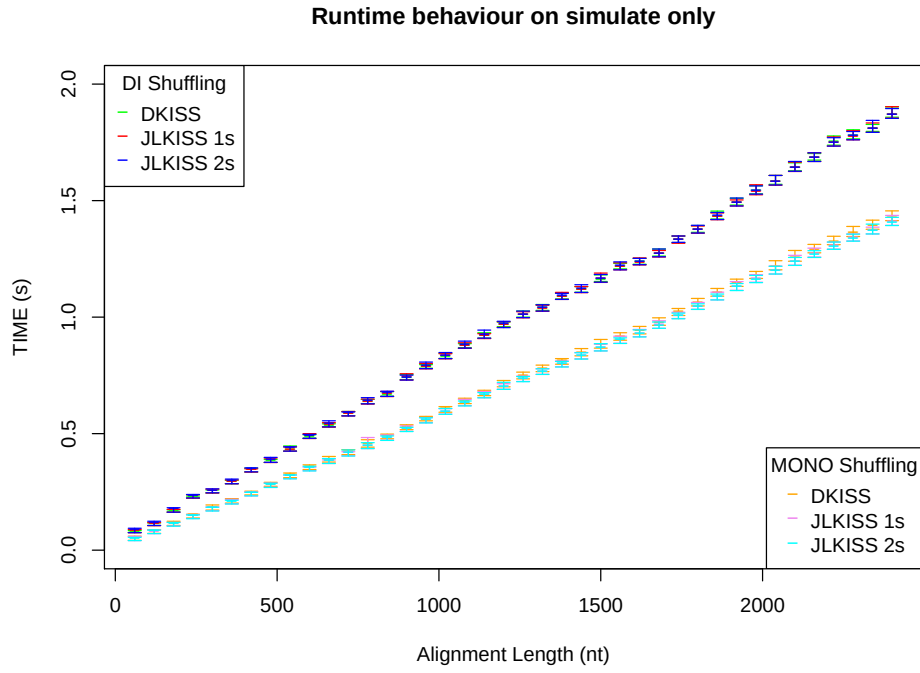


Figure 2.4: **Shuffling performance in dependence of alignment length:** An HIV alignment, was divided into sub-alignments, resulting in alignment lengths from 60 to 2400 nucleotides in 60 nt steps. SISSIZ was run in mono and di *simulation* mode, respectively. N=100

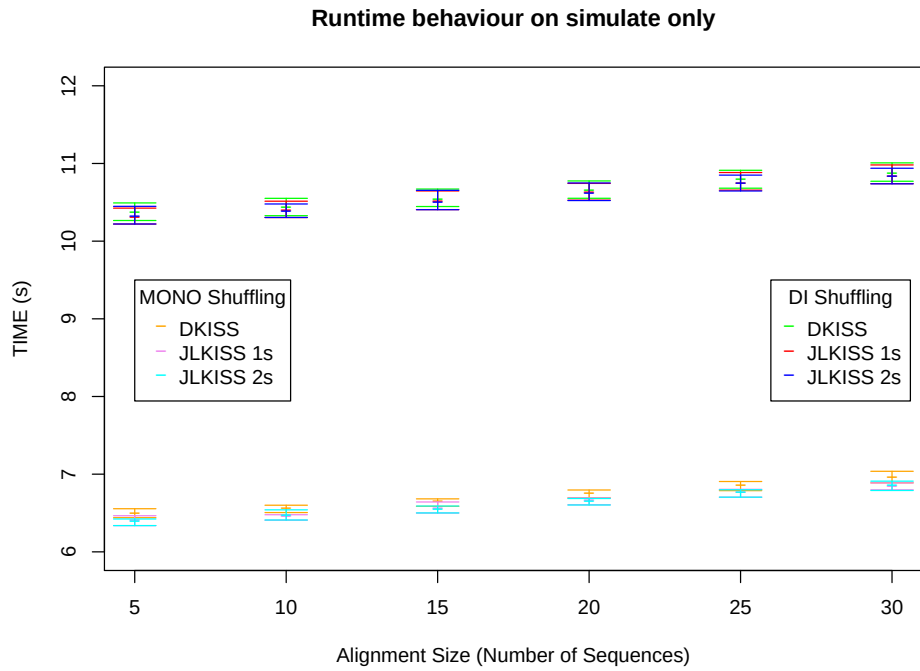


Figure 2.5: **Shuffling performance in dependence of alignment size:** SISSIZ was run on alignments with 5, 10, 15, 20, 25 and 30 HIV sequences in mono and di *simulation* mode, respectively. N=100

2.3.3 Conclusion

The upgrade of the PRNG adds some improvements and additional features to **SISSIZ**. Especially, the longer period length and the improved seeding can be advantageous for applications in clusters with parallel processes. The work also highlights the importance of PRNGs in computational biology. Although, the complexity of the PRNG has been increased, the run time of the new **SISSIZ** version's simulation mode could even be improved, at least tendentially. One could criticise that only a small set of sequences was used. However, the runtime change should be independent of the sequence composition since the underlying simulation algorithm of **SISSIZ** was not changed. Therefore, only the generation of the random numbers by the PRNG should influence the runtime. In the next section the quality of **SISSIZ**'s alignment simulations will be tested and compared to **multiPerm**.

2.4 Benchmark of simulation performance

Finally, the simulation performance of **SISSIZ** is compared to **multiPerm**. To this end, the alignment statistics of di-nucleotide content, mono-nucleotide content and MPI were compared between the randomized and the native alignments.

2.4.1 Methods

A diverse dataset comprising 104 RFAM (Nawrocki et al., 2015) families, which was also used by Rivas et al. (2017) was chosen for this benchmark. The protocol was as follows:

1. Download RFAM dataset from <http://eddylib.org/publications/RivasEddy16/RivasEddy16-supplement.tar.gz> (01.10.2017)
2. Extract RFAM seed families from *R-scape_Testset.sto*
3. Subsample every RFAM family to create 20 sub-alignments with 10 sequences per family
4. Remove gap-only columns from sub-alignments
5. Run **multiPerm** and **SISSIZ_beta** on sub-alignments
6. Calculate MPI, mono-nucleotide and di-nucleotide content from the original sub-alignments, **multiPerm** shuffled and **SISSIZ** shuffled sub-alignments.

2.4.2 Results

The differences of di-nucleotide frequencies, mono nucleotide frequencies and MPIs between the native alignments and **multiPerm** or **SISSIZ** shuffled counterpart, respectively are shown in figure 2.6. The differences were calculated pairwise. While **SISSIZ** creates alignments with deviations of the di/mono frequencies and MPIs averaging around 0, **multiPerm** alters the

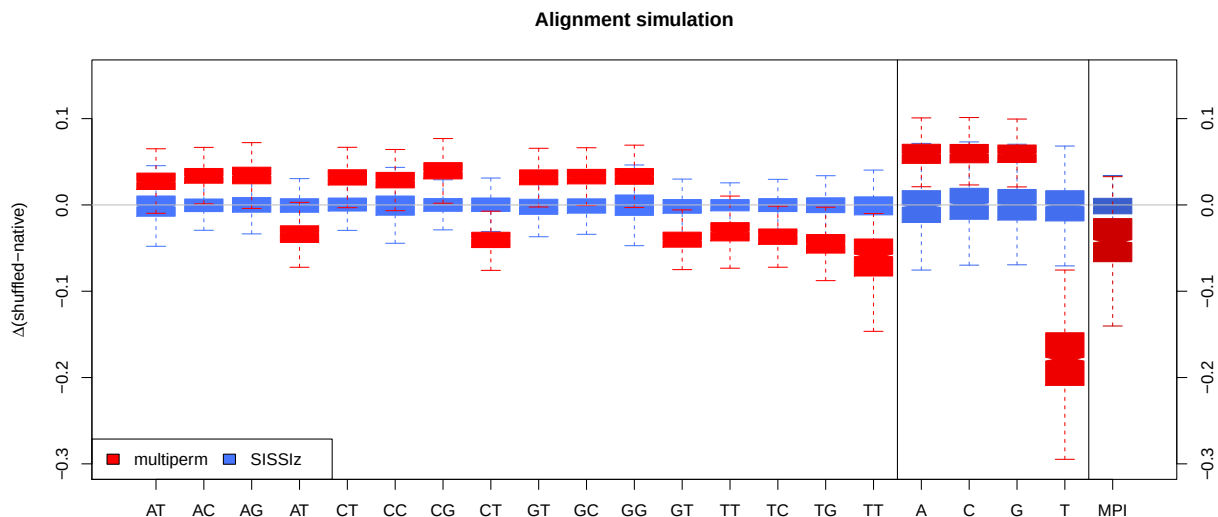


Figure 2.6: **Alignment statistics after simulation - RNA input alignments:** The plot shows the difference of di-nucleotide frequencies, mono-nucleotide frequencies and MPI between the native alignments and `multiperm` or `SSISz_beta` shuffled counterpart, respectively. The dataset consisted of 104 RFAM families, where every family is represented by 20 sub-sampled alignments with 10 sequences.

mono-nucleotide content by reducing the number of T/U nucleotides and thereby shifting the distribution of mono-nucleotides and also di-nucleotides as a results. Also the MPI identity is affected.

2.4.3 Discussion

On the one hand, the results are gratifying as it could be shown that `SSISz` reliably maintains all tested alignment statistics. On the other hand, the poor performance of `multiperm` is baffling. According the author of `multiperm`, “[the] essential idea behind the algorithm is to permute columns of the multiple alignment so as to preserve di-nucleotide frequencies...” (Anandam et al., 2009). By this definition and since the algorithm is not described to alter the base composition in any other way at least the mono-nucleotide content and MPI should remain unchanged. However, this is not the case. The underrepresentation of T nucleotides in the alignment is especially striking. Upon manual inspection of the alignment files two observations could be made: Firstly, that Us from the native (RNA) alignment were converted to Ts in the shuffled alignment; this would not change the counting statistics since T and U nucleotides are counted synonymously. The second observation was of course, that the number of T/U nucleotides in the `multiperm` shuffled alignment was lower than in the native alignment. Apparently, columns with a majority of U/Ts in the native alignment disappear in the shuf-

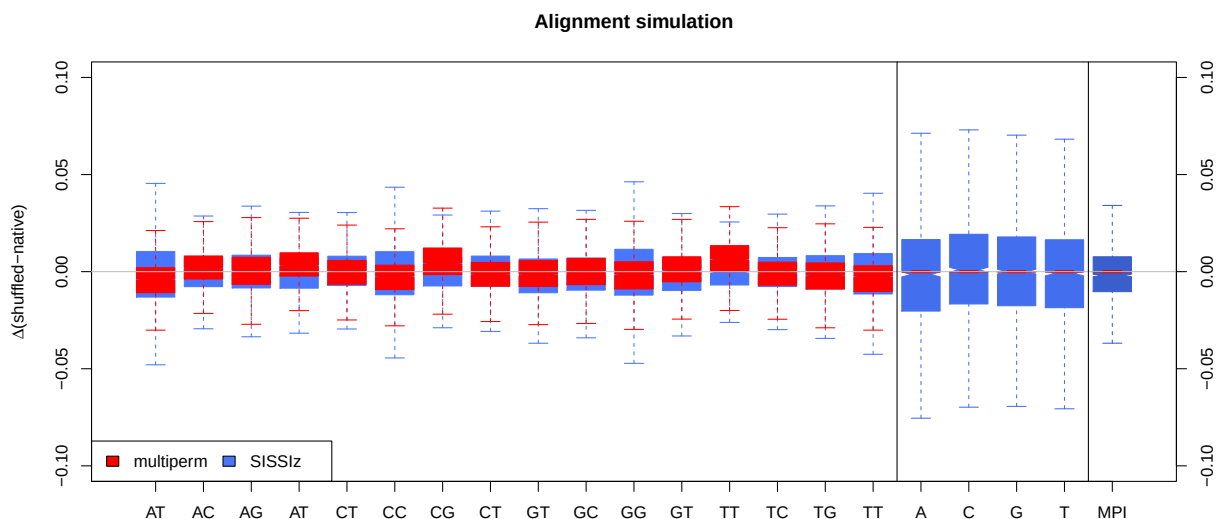


Figure 2.7: **Alignment statistics after simulation - DNA input alignments for multi-perm:** The plot shows the difference of di-nucleotide frequencies, mono-nucleotide frequencies and MPI between the native alignments and `multi-perm` or `SISSlz.beta` shuffled counterpart, respectively. The dataset consisted of 104 RFAM families, where every family is represented by 20 sub-sampled alignments with 10 sequences. U nucleotides in the native alignment were replaced by T nucleotides, in order to avoid problems with `multi-perm`.

fled alignments. This observation implies an incorrect handling of U nucleotides. To test this hypothesis all U nucleotides from the native alignments were replaced by Ts and `multi-perm` run again. The results are shown in figure 2.7. Now the statistics are as expected: The mono-nucleotide and MPI content does not change, since only columns are shuffled. Furthermore, the di-nucleotide content remains indeed approximately the same, something which was not shown by Anandam et al. (2009). Furthermore, the inspection of the `multi-perm` source code revealed that non canonical nucleotides are randomly converted to the canonical DNA nucleotides A,T,G and C. U nucleotides are seen by `multi-perm` as non canonical and are thus randomly converted which explains their relative change in abundance. One could argue whether this is a bug or not. Nevertheless, given that `multi-perm` was developed with the idea of randomizing RNA alignments such a handling of U nucleotides without providing a warning message is surprising. In any case, the user should be warned to provide the input alignment with DNA and not RNA nucleotides. Apparently, this error has found its way to publications; for example of Smith et al. (2013) who provided the raw data⁶ in the publication’s supplementary. Therein, a deviation of nucleotide counts between the native (RNA alignments) and shuffled alignments could be observed.

The question that remains is when to use which program for the creation of a negative

⁶http://www.martinalalexandersmith.com/ECS/benchmark_native.tgz

control dataset. While there is no general answer there are some approaches reported in literature. For example Gruber et al. (2010) used the shannon entropy of the alignment, which will be described in chapter 3.1.2, as a threshold: If the entropy is below 0.5, **multiPerm** is to be used and if the entropy exceeds 0.5 **SISSiZ**. The randomized alignments served as a negative control dataset to train their support vector machine RNAz (Chapter 3.1.2). Since the entropy is inversely correlated with average sequence identity of the alignment, **multiPerm** is applied on more conserved alignments. Conversely, **SISSiZ** is utilized on more diverse alignments, as Gruber et al. (2010) argues that **multiPerm** shuffling can then become *inefficient*, probably because information on covariation cannot be destroyed by merely shuffling columns. The alignment entropy threshold of 0.5 is also used by Will et al. (2013) to estimate the FDR (False Discovery Rate) in an RNAz screen on the Drosophila genome. Notably, the entropy threshold is used mostly in the context of RNAz. For other RNA gene finders different thresholds or parameters may be better suitable. An entirely threshold independent approach is taken by Smith et al. (2013) who tried either solely using **multiPerm** or combining both approaches by first using **SISSiZ** and then **multiPerm** to create a negative control dataset for their RNA gene finding pipeline. Even though, it is worth considering whether other parameters like the MPI could be used instead of the alignment entropy, it is not explored during this thesis. Since **SISSiZ** cannot be benchmarked by itself, only **multiPerm** can be used to create a negative control dataset for **SISSiZ**. This will become more clear in chapter 3.

Chapter 3

RNA gene finding

3.1 Introduction

3.1.1 SSSIz as an RNA genefinder

In addition to SSSIz’s capabilities as an alignment simulator, it can be used to identify structurally conserved ncRNAs in alignments. For this purpose, the simulation part of SSSIz is joined with RNAalifold from the Vienna RNA package (Lorenz et al., 2011): RNAalifold, which will be explained in more detail in chapter 3.1.3, is used to calculate the consensus MFE energy of the native input alignment and 100 corresponding SSSIz simulated alignments. Thus, it is possible to calculate a z-score (z) from the MFE of the native alignment ($native_{MFE}$) and the mean (μ_{MFE}) and the standard deviation (σ_{MFE}) of the simulated alignments’ MFEs as denoted in equation 3.1. The complete workflow of SSSIz is shown in figure 3.1 by Gesell and Washietl (2008).

$$z = \frac{native_{MFE} - \mu_{MFE}}{\sigma_{MFE}} \quad (3.1)$$

At this point it has to be mentioned that Alifoldz by Washietl and Hofacker (2004) uses a similar approach to calculate a z-score from a set of randomized alignments; however the null alignments are created by simple shuffling of alignment columns according to the rnazRandomizeAln.pl script (chapter: 2.1.2) without maintaining di-nucleotide statistics, although gap and local conservation patterns are maintained. The unique alignment simulation process of SSSIz which maintains the di nucleotide counts could be shown to reduce the FPR in ncRNA predictions (Gesell and Washietl, 2008).

The z-score serves as the predictor variable for ncRNAs in SSSIz. Low z-scores indicate a high likelihood that a structurally conserved RNA is present in the input alignment, as the

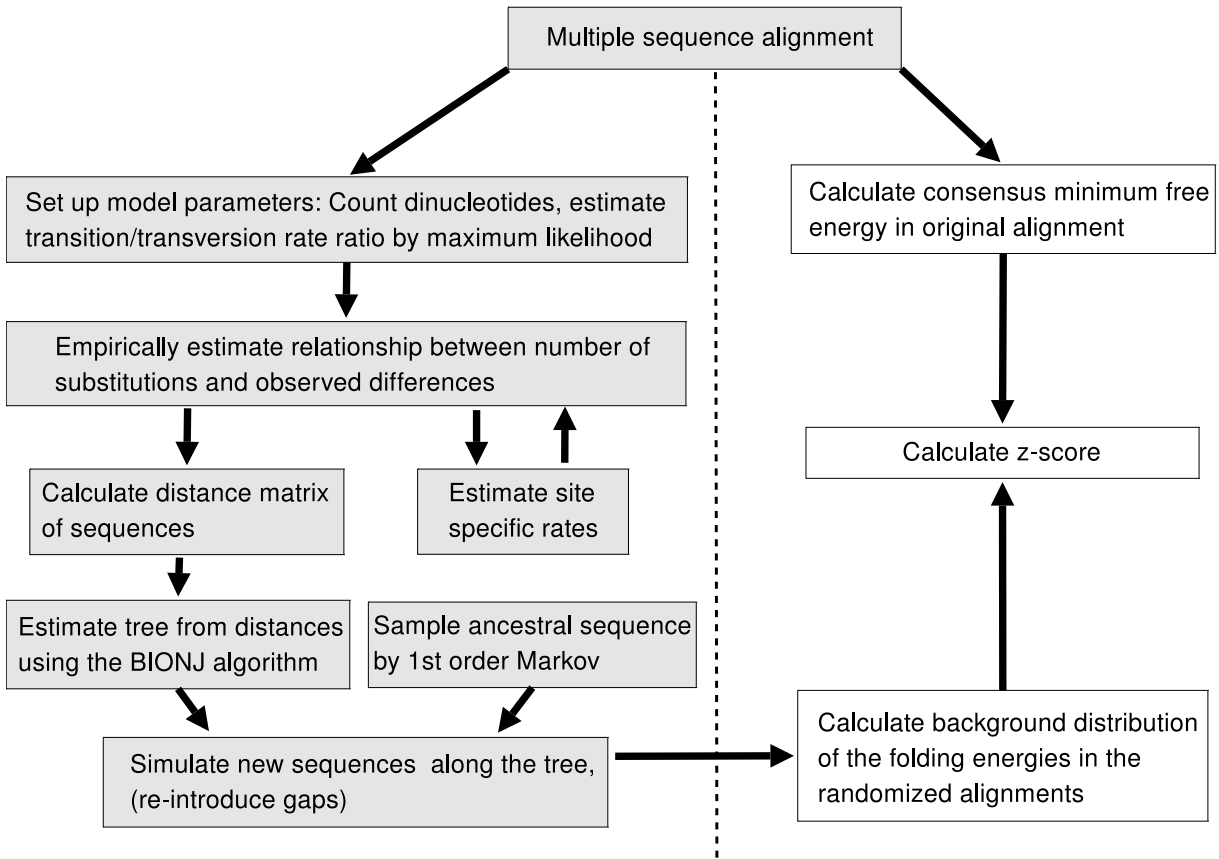


Figure 3.1: **SSISS RNA gene finder workflow:** When SSISS is used as an RNA gene finder its simulation module (chapter 2.1.2) is joined with a thermodynamic component: *RNAalifold* (chapter 3.1.3) is used to calculate the MFE of the consensus secondary structure for the native alignment and the simulated alignments (100 by default). As a results, a z-score can be calculated from the distribution of the consensus MFE of the simulated alignments and consensus MFE of the native alignment. The figure was taken from Gesell and Washietl (2008)

MFE of the native alignment is much lower then the MFE of the simulated alignments. The probability of a prediction to be true can be calculated from the z-score by integration over the standard normal distribution (equation: 3.2), at least theoretically.

$$P(z) = 1 - \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} \cdot e^{-\frac{1}{2}x^2} \quad (3.2)$$

For example a z-score of -2 would correspond to a probability of ≈ 0.997 and a z-score of -3 to a probability of ≈ 0.999 (Note: one-tailed integral). In practice the threshold values for the z-score at a given sensitivity and specificity are determined using ROC curves and have to be adjusted depending on the application. This will be discussed more throughly in the course of chapter 3.1.5.

3.1.2 RNAz

The first version of **RNAz** was published by Washietl et al. (2005) and has since been succeeded by version 2.0 (Gruber et al., 2010). In contrast to **SISSIZ**, **RNAz** is a support vector machine (SVM), a form of supervised machine learning approaches. This means that **RNAz** was trained to recognize structural RNA alignments based on thermodynamic stability, structural conservation and since version 2.0 the Shannon entropy of the alignment. To this end, actually two support vector are employed:

Firstly, a z-score is computed via support vector machine regression (SVR) as a measure for thermodynamic stability. Therefore, the z-score for every sequence in the alignment is derived from the sequence length and base composition via SVR descriptors. Only if the descriptors are out of the training dataset’s range, it is calculated as the average z-score $\bar{z}$ of all z-scores of the individual sequences in the alignment. The z-scores of the single sequences are in turn computed from the minimum free folding energy of the native sequence E_x and the distribution of its di-shuffled counterpart (μ_{E_x} and σ_{E_x}); the native sequence is therefore randomized using an eulerian algorithm (Altschul and Erickson, 1985) to generate 1000 random sequences per sequence in the alignment. The folding energy of the single sequences is computed by **RNAfold** which is also part of the Vienna RNA package (Lorenz et al., 2011). The formula is denoted in equation 3.3, where n refers to number of sequences x in alignment A . Hence, the calculation of the z-score differs substantially from the one calculated by **SISSIZ**.

$$\bar{z} = \frac{1}{n} \sum_{x \in A} \frac{E_x - \mu_{E_x}}{\sigma_{E_x}} \quad (3.3)$$

Secondly, the SCI is computed from the MFE of the alignment consensus secondary structure $E_{consensus}$ which is calculated by **RNAalifold**, and the folding energy of the individual sequences E_x in the alignment, computed by **RNAfold**, using formula 3.4.

$$SCI = \frac{E_{consensus}}{\frac{1}{n} \sum_{x \in A} E_x} \quad (3.4)$$

A $SCI < 1$ indicates low structural conservation, a $SCI \approx 1$ a perfectly conserved structure and a $SCI > 1$ a highly conserved structure that is additionally supported by compensatory mutations (Washietl and Hofacker, 2004). Particularly, low SCI values can make it impossible for **RNAalifold** to calculate a consensus minimum free energy as it implies that no common structure exists. At the same time no threshold values appear to exist. Moreover, even a SCI of 1 can be misleading if the sequences in the alignment are very similar (high MPI). Nevertheless, the SCI constitutes a powerful measure for structural conservation which is why

it was incorporated to the updated version of **SISSIZ**.

Thirdly, and since version 2.0 of **RNAz**, the normalized shannon entropy H is computed according to equation 3.5 as a measure for the degree of sequence conservation in the alignment. In the formula, Σ describes the alphabet $\{A, U, G, C, -\}$, l the length of the alignment and p_j^i the relative frequency of j in alignment column i . Note, that the shannon entropy has an inverse relationship to the MPI and ranges from 0 to 2.

$$H = -\frac{1}{l} \sum_i \sum_{j \in \Sigma} p_j^i \log_2 p_j^i \quad (3.5)$$

Eventually, the z-score $\bar{z}$, the structural conservation index SCI and the shannon entropy H are used by a second SVM to calculate a so called SVM RNA class probability, which is the final predictor for structurally conserved RNAs: It ranges from 0 (low likelihood) to 1 (high likelihood of structurally conserved RNA). Interestingly, Gruber et al. (2010) used **multiperm** shuffled alignments to train the second SVM of **RNAz** if the shannon entropy was below 0.5 and **SISSIZ** simulated alignments for entropies above 0.5, so that more diverse alignments are fully randomized.

3.1.3 RNAalifold

RNAalifold constitutes a crucial component of every **SISSIZ** prediction as it is used for the computation of the consensus Minimum Free Energy (MFE) of the native and simulated alignments. Hofacker et al. (2002) described their algorithm of **RNAalifold** "as a (complicated) variant of the circular matching problem (MCMP)". The MCMP has become widely known as the Nussinov algorithm (Nussinov et al., 1978), the goal of which is to find the maximum number of matching base pairs in a sequence $x = (x_1, \dots, x_l)$ with the imposition of the following restrictions:

- A base pair between x_i and x_j , hereafter denoted as $\{x_i, x_j\}$, may only be formed if it is contained in the set of allowed base pairs $B = \{AU, UA, GC, CG, GU, UG\}$. In addition to the canonical base pairs $\{AU, UA, GC, CG\}$, the so called wobble base pairs $\{GU, UG\}$ are usually contained in B .
- The graph, representing the RNA secondary structure, must be outerplanar. Therefore, the edges connecting two vertices must not cross. As a result of this simplification, pseudoknot structures are excluded from the prediction. Mathematically, the following condition for every base pair $\{x_i, x_j\}$ and $\{x_g, x_h\}$ with $i < j$ and $g < h$ has to be fulfilled: $i < g < h < j \vee g < i < j < h$

- For every base pair $\{x_i, x_j\}$ with $i < j$ the inequation $j - i > m$ must be fulfilled, where m is a parameter that defines the minimum loop size. Usually m is set to 3 in order to avoid thermodynamically impossible secondary structures.

In order to find the maximum number of matching base pairs, advantage of the dynamic programming paradigm was taken by which the Maximum Matching Size (MMS) can be derived from the MMS of the subsequences (Equation 3.6, adapted from Nussinov et al. (1978)).

$$MMS(i, j) = \max \left\{ \begin{array}{l} MMS(i, j - 1) \\ MMS(i, k - 1) + MMS(k + 1, j - 1) + 1 \\ k : i \leq k < (j - m) \wedge \{x_k, x_j\} \in B \end{array} \right\} \quad (3.6)$$

The $MMS(i, j)$ for the subsequence $x_{i,j}$ can be obtained from $x_{i,j-1}$ either by introducing an unpaired base at x_j ($MMS(i, j - 1)$) or by adding a base pair $\{x_k, x_j\}$ ($MMS(i, k - 1) + MMS(k + 1, j - 1) + 1$) with $i \leq k < (j - m)$. By following this recursion a $N \times N$ matrix (where N is the length of the sequence x) containing the MMS for every subsequence $x_{i,j}$ can be filled. Thus a time complexity of $O(n^3)$ and a memory complexity of $O(n^2)$ arises. Eventually, the secondary structure with the maximum number of base pairs can be derived from backtracking in the MMMS matrix. The university Freiburg offers a great web interface¹ for the visualization of the Nussinov algorithm. Similarly, the algorithm can readily be extended to calculate the minimum free folding energy E of a sequence by assigning every base pair $\{x_k, x_j\}$ a certain experimentally derived energy contribution $\beta(k, j)$ as denoted in equation 3.7. Instead of finding a secondary structure that maximizes the number of base pairs, it is now the goal of the algorithm to find a structure that minimizes the free folding energy (MFE).

$$E(i, j) = \min \left\{ \begin{array}{l} E(i, j - 1) \\ E(i, k - 1) + E(k + 1, j - 1) + \beta(k, j) \\ k : i \leq k < (j - m) \wedge \{x_k, x_j\} \in B \end{array} \right\} \quad (3.7)$$

In RNAalifold the MCMP was extended in several aspects:

- Instead of the maximum number of matching base pairs the minimum free energy is calculated using a loop based energy model. Thereby not only the energy contribution from base pairing sites are considered, but also the one of hairpin loops, interior loops, and multi loops. By this method, called loop decomposition, a more realistic MFE can be computed as the contribution of destabilizing loop regions and the number of branches is taken into account.

¹<http://rna.informatik.uni-freiburg.de/Teaching/index.jsp?toolName=Nussinov> (9.12.17)

- The algorithm is alignment based: Therefore, the MFE for the consensus structure is calculated by averaging the energy contribution of the single sequences in the alignment.
- Moreover, a covariation term $\gamma(i, j)$ is introduced that can modulate the energy contribution at base paired positions by favoring base pairs that are also supported by compensatory mutations.

It would be out of the scope of this thesis to fully describe the loop decomposition algorithm of **RNAalifold** in detail. Nevertheless, the most important aspects will be discussed briefly. Most importantly, the so called Ribosum scoring matrix which was added to **RNAalifold** by Bernhart et al. (2008) will be described as it differs greatly from the default covariance model of the original **RNAalifold** (Hofacker et al., 2002) and is also used by **SISSIZ** in the following benchmarks.

The loop decomposition works as follows: A base pair $\{x_i, x_j\}$ can either enclose a hairpin loop, be part of an interior loop or enclose a multiloop region. Multiloop regions can then again be decomposed to hairpin loops, interior loops or multi loops. Base stacking is treated as a special case of the interior loop where for two base pairs $\{x_i, x_j\}$ and $\{x_k, x_l\}$ with $i < k < l < j$ the condition $i = (k-1) \wedge j = (l+1)$ is full filled. **RNAalifold** uses the Turner energy parameters (Mathews et al. (2004a), Mathews and Turner (2006)) for all the respective loop types.

In **RNAalifold** there are two options to incorporate (the unadjusted) information on covariation $\gamma'(i, j)$ between sites i and j of an alignment A into the prediction of the MFE:

1. Based on the Hamming distance: Equation 3.8. The covariation contribution $\gamma'(i, j)$ is given as the sum of all pairwise hamming distances between sequences $\alpha, \beta \in A$ at sites i and j that fulfill the requirement that the base pairs $\{\alpha_i, \alpha_j\}$ and $\{\beta_i, \beta_j\}$ are in the set of allowed base pairs $B = \{AU, UA, GC, CG, GU, UG\}$.

$$\gamma'(i, j) = \frac{1}{2} \sum_{\substack{\alpha, \beta \in A \\ \alpha \neq \beta}} \left\{ \begin{array}{ll} h(\alpha_i, \beta_i) + h(\alpha_j, \beta_j) & \text{if } (\{\alpha_i, \alpha_j\} \in B) \wedge \\ & \text{if } (\{\beta_i, \beta_j\} \in B) \\ 0 & \text{else} \end{array} \right\} \quad (3.8)$$

2. Based on the Ribosum Matrix: Equation 3.9: The covariation contribution $\gamma'(i, j)$ is computed from a log-odds probability matrix $R(ab, cd)$ that has been derived from 13,500 aligned ribosomal SSU sequences from the European ribosomal RNA Database (Wuyts et al., 2004). Since the secondary structure of the SSU rRNA is well known, the frequencies of the aligned nucleotides a and c $f(ac)$ in one alignment column and the frequencies of nucleotides b and d $f(bd)$ in another column with respect to the frequency of the corre-

sponding base pairs $\{a, b\}$ and $\{c, d\}$ $f(ab, cd)$ could be determined. 99 Ribosum matrices for different MPI classes were crafted according to this (empirical) approach.

$$R(ab, cd) = \log \frac{f(ab, cd)}{f(ac) \cdot f(cd)} \text{ with } a, b, c, d \in \{A, U, G, C\} \quad (3.9)$$

In contrast, equation 3.10 shows how the Ribosum matrix is incorporated into the covariation score $\gamma'(i, j)$ of **RNAalifold**. Depending on the alignment's MPI, 1 of the 99 Ribosum matrices $R(ab, cd)$ is used to obtain a log-odds ratio for two sequences α and β at sites i and j in an alignment A . The scaling factor s makes sure that $s \cdot R(ab, cd)$ lies in the same range as the hamming distances. Eventually, $\gamma'(i, j)$ is calculated from the sum of all pairwise log-odds ratios at site i and j multiplied by s .

$$\gamma'(i, j) = \frac{1}{2} \sum_{\alpha \in A} s \cdot R(\alpha_i \alpha_j, \beta_i \beta_j) \quad (3.10)$$

Finally, $\gamma'(i, j)$ is adjusted by a $\delta \sum_{\alpha \in A}$ term, which penalizes for gaps and impossible base pairs, to obtain $\gamma(i, j)$ denoted in equation 3.11. $\gamma(i, j)$ then modulates the loop energies as an additive term.

$$\gamma(i, j) = \gamma'(i, j) + \delta \sum_{\alpha \in A} \begin{cases} 0 & \text{if } (\{\alpha_i, \alpha_j\} \in B) \\ 0.25 & \text{if } (\alpha_i \wedge \alpha_j = \text{gap}) \\ 1 & \text{else} \end{cases} \quad (3.11)$$

Bernhart et al. (2008) could show, that the use of the Ribosum matrix improves the consensus MFE and secondary structure prediction of **RNAalifold** in almost all cases.

3.1.4 SHAPE

Even though this master's thesis has a strong focus on *in silico* methods for the calculation of RNA structures in context of ncRNA gene finding, the experimental approach also plays an important role since support for probing data was added to the new **SISSIZ** version to aid the prediction of ncRNAs. One widely used probing technique to obtain information on RNA structure is called SHAPE, *Selective 2'-hydroxyl acylation analyzed by primer extension* (Mortimer and Weeks (2007), McGinnis et al. (2012)). The method exploits the differing reactivity of certain reagents such as 1M7, 1M6 and NMIA (1-methyl-7-nitroisatoic anhydride, 1-methyl-6-nitroisatoic anhydride and N-methyl-isatoic anhydride, respectively) for the 2' OH group of the RNA sugar backbone, to which the reagent covalently binds: While 1M7 reacts preferentially with flexible, non base pairing nucleotides, 1M6 and NMIA can be used to probe for

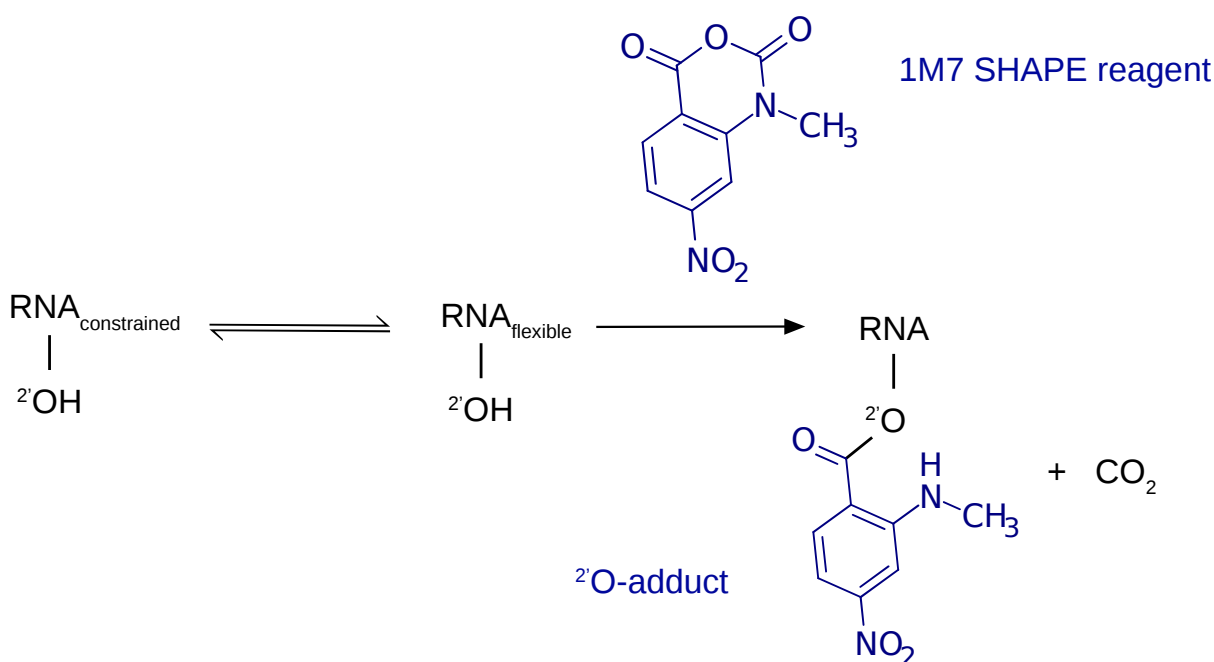


Figure 3.2: **SHAPE reaction:** The figure shows how the 1M7 reagent covalently binds to the 2' OH group of the RNA sugar backbone, releasing CO_2 . The acylation occurs preferentially at sides with higher flexibility, such as loop regions but less often at paired nucleotides. Adapted from https://en.wikipedia.org/wiki/Nucleic_acid_structure_determination#/media/File:1M7_RNA_adduct.png (January 15, 2018)

non-canonical and tertiary interactions (Smola et al., 2015). 1M7 constitutes the most important reagent as it sheds light on local nucleotide flexibility at nucleotide resolution. The reaction is depicted in figure 3.2. After the treatment with the SHAPE reagent, the RNA strands are reversely transcribed to cDNA and sequenced. At modified sites non-complementary nucleotides are introduced by the reverse transcriptase. Thus, it is possible to calculate a reactivity score for every side in the RNA sequence based on the observed mutations, which corresponds to the side's likelihood to be modified by the SHAPE reagent.

SHAPE is also amenable to high throughput experiments: For example, Siegfried et al. (2014) used SHAPE-MaP (mutational profiling) to determine the structure of the HIV genome or Lucks et al. (2011), who presented the SHAPE-Seq pipeline to obtain large scale information on RNA secondary and tertiary structure. Recently Selega et al. (2017) presented a novel pipeline that works on a transcriptome wide level with a higher sensitivity in detecting modified sites improving large scale structural probing.

The probing data can be incorporated in thermodynamic RNA folding algorithms by converting reactivity scores into pseudo energy contributions. There are various approaches how to achieve the best solution: Mathews et al. (2004b), Deigan et al. (2009), Zarrinhalam et al. (2012), Washietl et al. (2012) and Wu et al. (2015). Lorenz et al. (2016) found that the approach

of Deigan achieved on average the best results; the conversion is based on the formula 3.12,

$$\Delta G_{SHAPE}(i) = m * \ln(reactivity_{SHAPE}(i) + 1) + b \quad (3.12)$$

where i is the position in the sequence and m and b are adjustable parameters. **RNAalifold** uses Deigan's method by default and with the parameters $m = 1.8$ and $b = -0.6$. As a result, paired regions (with low reactivity) obtain stabilizing (negative) energy contributions and flexible regions (with high reactivity) destabilizing (positive) energy contributions. Even though, Lorenz et al. (2016) could show improved prediction accuracy in many cases, the accuracy decreased for some RNAs which was explained by the presence of pseudoknots and the possibility for errors in experimental data. Last but not least, it has to be mentioned that also non thermodynamic approaches for the integration of probing data exist. For example, the programs **Probfold** (Sahoo et al., 2016) and **PPfold** (Sukosd et al., 2012) both combine SCFGs (Stochastic Context Free Grammar) with probing data.

3.1.5 ROC curves

Introduction

Receiver Operator Characteristic (ROC) curves can be used to assess the performance of predictive tests and to find the optimal threshold value for a tests' predictor variable. In the case of **SISSIZ** the predictor variable is the z-score which describes how far the native alignment's Minimum Free Energy (MFE) differs from the mean MFE of the shuffled background alignments (**NOT** to be confused with the **multiPerm** (Anandam et al., 2009) shuffled alignments used as the negative control in the following benchmarks - chapter 3.3) and thus represents the likelihood of the given input alignment to be structurally conserved. For **RNAz** (Gruber et al., 2010) the SVM RNA-class probability serves as the predictor variable. In the ROC curve the sensitivity (SEN) is plotted against the False Positive Rate (FPR) at various thresholds for the predictor variable.

Thus the ROC curve can be described as a function of the predictor variable z :

$$f(z) = \begin{bmatrix} FPR(z) \\ SEN(z) \end{bmatrix} \quad (3.13)$$

The sensitivity and FPR can again be described as a function of z and are calculated as follows:

$$SEN(z) = \frac{TP(z)}{TP(z) + FN(z)} \quad (3.14)$$

		Predicted condition	
		Prediction positive	Prediction negative
True condition	Condition positive	True Positive (TP)	False Negative (FN) Type II error
	Condition negative	False Positive (FP) Type I error	True Negative (TN)

Table 3.1: The confusion matrix shows how the parameters True Positive (TP), False Negative (FN), False Positive (FP) and TN (True Negative) are defined by the true condition and the predicted condition.

TP ... True Positives, FN ... False Negatives

$$FPR(z) = 1 - SPE(z) = 1 - \frac{TN(z)}{TN(z) + FP(z)} = \frac{FP(z)}{TN(z) + FP(z)} \quad (3.15)$$

SPE ... specificity, TN ... True Negatives, FP ... False Positives

Table 3.1 shows that the number of False Positives and True Positives can be determined from the positive control dataset (condition positive) and the number of False Positives and True Negatives from the negative control dataset (condition negative). In the benchmark tests all native RFAM (Nawrocki et al., 2015) alignments were assumed to contain structurally conserved RNAs and where thus used to calculate the sensitivity. The FPR on the other hand was determined by the analysis of the **multiPerm** shuffled RFAM alignments (negative control data set) which were not expected to contain any structurally conserved RNAs.

The threshold values for z can be visualized in a ROC curve using a color code as shown in figure 3.3. But how can the ROC curve be interpreted? If the curve was on the diagonal from $[0, 0]$ to $[1, 1]$ the predictive power would be equal to random guessing. The higher the ROC curve rises over the diagonal the better the predictive test. Generally, the goal is to maximize the area under the curve (AUC) which is often used as measure for performance. It can be described as the integral of equation equation 3.13 from 0 to 1. In practise and since the function of the ROC curve is not continuous, the AUC can, for example, be determined by applying the trapezium rule. Even though, the AUC is a convenient measure to compare the performance from various tests with a single number, it is usually more important to find the sensitivity at a given FPR.

Algorithms for calculation

While many R-packages for the calculation of ROC-curves such as **ROCR** and **pROC** (Robin et al., 2011) exist, I decided to implement my own functions in R.

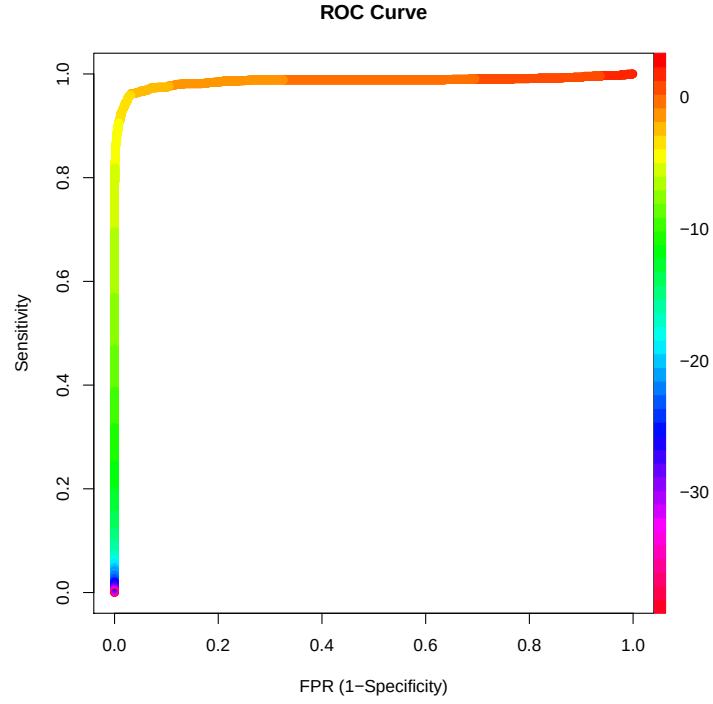


Figure 3.3: **The ROC curve:** In a ROC curve diagram the False Positive Rate is plotted against the sensitivity in dependence of the predictor variable z . In this plot the curve is color coded for the predictor variable which visualizes the threshold value for a given FPR and sensitivity pair (color gradient scale on right side of plot).

Method 1

In the first (heuristic) method, the calculation of the ROC curve is calculated for a set of predictor variables in a certain range $[a, b]$ where a and b represent the maximum and the minimum value of the predictor variable, respectively. A heuristic can be made by not calculating all values in the given range $[a, b,]$ but rather in equal sized intervals of size s . Thus the number of calculations for the ROC curve can thus be limited to $(|a - b|)/s$.

$$f(z) = \begin{bmatrix} SEN(z) \\ FPR(z) \end{bmatrix}, \{z \in \mathbb{R} \mid a < z < b\}$$

```

1 ROCI= function(positive ,negative ,from ,to ,width=0.1) {
2   out=data.frame(predictor=c(1:abs((to-from)/width-1))*NA,SEN=c(1:abs((to-from)/
   width-1))*NA,SPE=c(1:abs((to-from)/width-1))*NA)
3   length_input=length(positive)
4   length_output=length(out$SEN)
5   for (i in (1:length_output)) {
6     out$predictor[i]=from-0.1*(i-1)
7     out$SEN[i]=length(positive[positive<out$predictor[i]])/length_input

```

```

8 out$SPE[i]=length(negative[negative>out$predictor[i]])/length_input
9 }
10 return(out)
11 }

```

Method 2

While the first method can be advantageous on large datasets, its heuristic may deliver unsatisfactory results if the dataset is small or the range of the predictor variables is inconvenient. Then the second algorithm can be advantageous as it guarantees the maximum possible resolution of the ROC curve. To this end, a binary classification system is used: The list of predictor variables from the positive and negative dataset are merged and sorted to a joined vector $[z_i, c_i]$ with n entries, z as the predictor variable, c as the binary classifier (1 for positive control dataset, 0 for negative control dataset) and i as the index variable of the vector. Now the ROC curve can be calculated as follows:

$$f(z_i) = \begin{bmatrix} SEN(z_i) \\ FPR(z_i) \end{bmatrix} = \begin{bmatrix} \frac{n - \sum_{i=1}^n c_i}{n} \\ 1 - \frac{i - \sum_{i=1}^n c_i}{n} \end{bmatrix}, \{i \in \mathbb{N} \mid 1 \leq i \leq n\}$$

n ... number of entries in joined vector

```

1 ROC2 = function(x,y,option=TRUE) {
2   length=length(x)
3   if (length!=length(y)) print("ERROR")
4   class=c(c(1:length)*0+1,c(1:length)*0)
5   predictor=c(x,y)
6   df=data.frame(predictor , class)
7   df=df[order(df[,1],decreasing=option),]
8   return_object= data.frame(predictor=df[,1],SPE=c(1:(length*2))*0,SEN=c(1:(length*2))*0)
9   for (a in (1:length(df[,1]))) {
10    return_object$SEN[a]=(length-sum(df[(1:a),2]))/length
11    return_object$SPE[a]=(a-sum(df[(1:a),2]))/length
12  }
13  return=return_object
14 }

```

3.2 Upgrading Vienna RNA package’s RNAalifold

3.2.1 Motivations and Background

One of the main objectives of this master’s thesis was to integrate an updated version of **RNAalifold** in **SISSIZ** to support SHAPE data, which emerged as the most widely used method for chemical probing. As described in the previous section, the method exploits the shielding properties of RNA base-pairing against a mutagenic reagent. As paired bases are less likely to be modified by the chemical reagent it is possible to reconstruct the secondary structure of RNAs using the probing data for guidance. Since Version 2.2 of the Vienna RNA package, SHAPE data is supported by the **RNAalifold** module, which is used by **SISSIZ** to calculate the MFE (Minimum Free Energy) of the native and simulated background alignments; it is thus crucial to the calculation of the z-score. In this section the following questions will be addressed:

1. Runtime of **RNAalifold** independently from and with **SISSIZ**
2. Performance of **SISSIZ** with new **RNAalifold** version
3. Performance of **SISSIZ** with SHAPE data

3.2.2 Runtime behavior

Introduction

Before the new version of the Vienna RNA package was integrated in **SISSIZ** the runtime of several **RNAalifold** versions was compared: v.1.6.1, v.1.8.5, v.2.0.0 (Lorenz et al., 2011), v.2.1.0, v.2.2.0, v.2.3.2, v.2.4.1. **RNAalifold** versions 1.61 and 1.8.5 were chosen because they have been used in **SISSIZ** version 1 (Gesell and Washietl, 2008) and 2 (Smith et al., 2013) respectively. The later **RNAalifold** versions are all above 2.0.0, support constraint files and use the new Turner 2004 energy parameters. Since version 2.2.0 SHAPE reactivity is supported (table 3.2). Around September 2017, **RNAalifold** 2.4.1 was the latest available version. Therefore, all versions above 2.2.0 could theoretically have been used in the new **SISSIZ** version. Since early versions of the updated **SISSIZ** with **RNAalifold** 2.3.2 appeared to be notably slower than the former, it was necessary to find the fastest version of **RNAalifold**.

Version	Changelog concerning RNAalifold
1.6.1	NA
1.7.2	NA
1.8.5	Ribosum Matrix
2.0.0	Support of Constraint files, new Turner 2004 energy parameters

2.0.3	Bugfix for RNAalifold partition function and base pair probabilities, added Boltzmann factor scaling
2.1.0	G-Quadruplex support (command line parameter)
2.1.6	New commandline switches allow for elimination of non-canonical base pairs from constraint structures
2.1.9	Added Sequence Conservation index (SCI) option, Fixed bug in alifold partition function for circular RNAs, Fixed bug in alifold that scrambled backtracking with activated G-Quadruplex support, Fixed bug in alifold backtracking for larger G-Quadruplexes
2.2.0	SHAPE reactivity support
2.2.8	Fix partition function scaling for long sequences, Fix bad memory access in RNAalifold, Fix regression in RNAalifold to restore covariance contribution ratio determination for circular RNA
2.2.9	Add various new commandline options to manipulate sequence/alignment IDs, Support of Clustal, Stockholm, FASTA, or MAF alignments as input, Lift restriction of sequence number in alignments for RNAalifold , Enable ANSI colors for TTY output
2.3.1	Fix regression in <code>vrna_md_update()</code> that resulted in incomplete init of reverse-basepair type array
2.3.2	Report SCI of 0 when sum of single sequence MFEs is 0

Table 3.2: Benchmarked versions of RNAalifold and their changelog which was obtained from: <https://www.tbi.univie.ac.at/RNA/changelog.html> (January 15, 2018)

Methods

The dataset described in this section will be used in many of the following benchmarks. It contains a selection of structural RNA families from the RFAM (Nawrocki et al., 2015) database, which have been used in a recent benchmark of **Rscape**, a program for the detection of covariation in RNA alignments by Rivas et al. (2017).

The protocol shown below produces a total of 1760 sub-alignments with 10 sequences. Each of the 88 RFAM families is represented by 20 sub-alignments.

1. Download RFAM dataset from <http://eddylib.org/publications/RivasEddy16/RivasEddy16-supplement.tar.gz> (01.10.2017)
2. Extract RFAM seed families from *R-scape_Testset.sto*
3. Remove alignments longer than 500 nt

4. Subsample every RFAM family to create 20 sub-alignments with 10 sequences per family
5. Remove gap-only columns from sub-alignments
6. Run all **RNAalifold** versions with default parameters / ribosum scoring matrix on the sub-alignments and measure runtime (real time)
7. Run the updated **SISSIZ** and **SISSIZ** (v2.0 and v1.0) with default parameters / ribosum scoring matrix on the sub-alignments and measure runtime

Results

The runtime of all **RNAalifold** versions was plotted relative to version 1.8.5 as can be seen in figure 3.4. It is clearly visible that v.1.6.1 performed much faster than any other version of **RNAalifold**. This increase in runtime was also observed by Bernhart et al. (2008) after **RNAalifold** received a major update which among other improvements introduced the Ribosum matrix. Of the versions above 2.0, the latest **RNAalifold** version 2.4.1 performed best for the runs in default mode and second best for the runs in ribosum mode. With version 2.4.1 in default mode the runtime remains on average unchanged relative to version 1.8.5 but in ribosum mode the runtime increases by approximately 10%. Even though the changes may appear minute, they can have a big impact on the runtime of **SISSIZ** as it uses **RNAalifold** extensively. Nevertheless, **RNAalifold** has been improved with every version of the Vienna RNA package and bugs have been eliminated as can be seen in table 3.2.

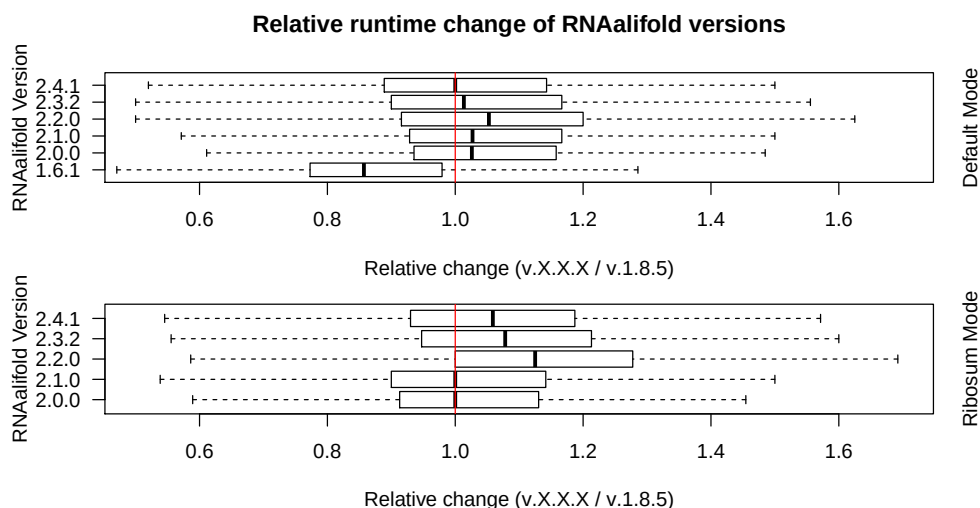


Figure 3.4: **RNAalifold runtime:** The figure shows the runtime of various **RNAalifold** versions relative to version 1.85 which is used by the second published version of **SISSIZ** (Smith et al., 2013). Version 1.6.1 was used in the first published version of **SISSIZ** (Gesell and Washietl, 2008). Version 2.4.1 was the latest available in September 2017. Since version 2.0.0, SHAPE data is supported by **RNAalifold**. All versions were run on a dataset of 88 RFAM families, where each family is represented in 20 sampled alignments with 10 sequences. In the upper boxplot **RNAalifold** was run in default mode and in the lower boxplot with the ribosum scoring matrix (not available for version 1.6.1).

Since version 2.4.1, which also was the latest version at the time, emerged as the fastest version of *RNAalifold* with SHAPE support it was chosen to be used in the updated *SISSiZ* version. The runtimes were analyzed relative to version 2 of *SISSiZ* (Smith et al., 2013) and are visualized in figure 3.5. It becomes apparent that the first version *SISSiZ* (Gesell and Washietl, 2008) is about 10% faster than the second version of *SISSiZ*, a finding that was expected from the previous benchmark with *RNAalifold*. Moreover, the runtime of the new *SISSiZ* version is increased by approximately 5% if the SCI is not calculated and by over 10% if the SCI is computed too. As a result, the calculation of SCI in *SISSiZ* should only be activated if needed since it notably adds up to the computation time. The benefits of the SCI in *SISSiZ* will be explored in chapter 3.3.2. Paradoxically, the runtime of the new *SISSiZ* version with the Ribosum matrix is slightly better than in default mode even though the benchmark of the respective *RNAalifold* versions would have suggested the opposite. To conclude, the runtime of the updated *SISSiZ* version increases by approx. 5%. In exchange *SISSiZ* benefits from various bugfixes in *RNAalifold* and can use new functions such as the incorporation of SHAPE data.

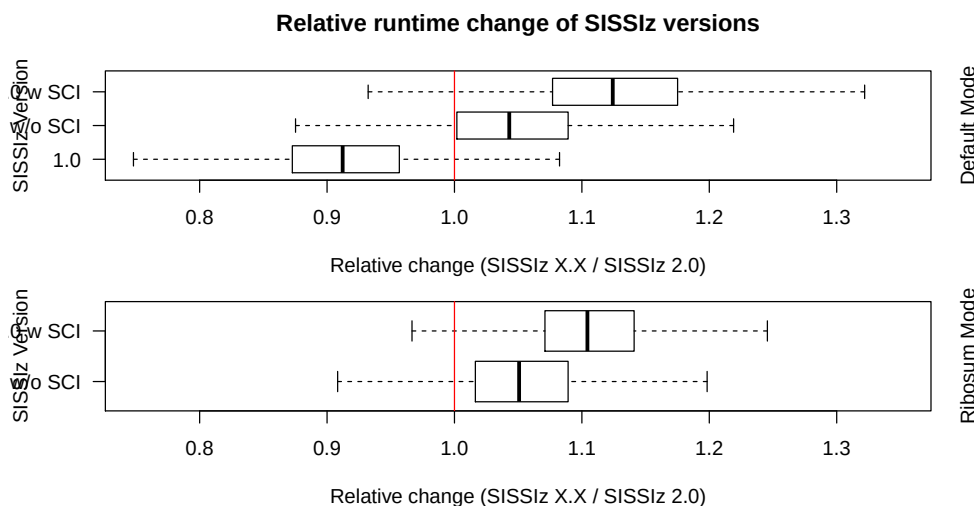


Figure 3.5: *SISSiZ* runtime: The figure shows the runtime of various *SISSiZ* versions relative to version 2, the second published version of *SISSiZ* (Smith et al., 2013). Version 1, the first published version of *SISSiZ* (Gesell and Washietl, 2008). The new version 3 of *SISSiZ* uses *RNAalifold* version 2.4.1 and can also calculate the SCI as an output parameter; thus one boxplot shows the relative runtime of *SISSiZ* with and without SCI calculation. All versions were run on a dataset of 88 RFAM families, where each family is represented in 20 sampled alignments with 10 sequences. In the upper boxplot *SISSiZ* was run in default mode and in the lower boxplot with the ribosum scoring matrix (not available for version 1 of *SISSiZ*)

3.3 Benchmarking

But how does the new version of `RNAalifold` affect the ncRNA prediction performance of the updated `SISSIZ` version? To this end, several benchmarks were conducted that measure the performance of the updated `SISSIZ` version against `SISSIZ` (v.2.0) and `RNAz` in ROC curves. Furthermore, the impact of different alignment programs, experimental SHAPE data and the SCI are investigated.

3.3.1 SISSIZ old VS new

For the first benchmark advantage of an already existing dataset was taken, that was readily available from Rivas et al. (2017) who used a set of 104 RFAM Nawrocki et al. (2015) families to assess the performance of their program `Rscape` which can be used to detect and assess the significance of covarying columns in alignments. The RFAM database itself contains a huge collection of alignments for structurally conserved RNAs which are grouped to families based on covariance models.

Methods

For this benchmark 88 of the 104 RFAM families from the dataset of Rivas et al. (2017) were utilized discarding seed alignments with a length of over 500 nt in order to reduce the runtime which increases to the power of 3 of the alignment length. (Not to be confused with the linear simulation runtime of `SISSIZ`.) More detailed information on the used RFAM families can be found in tables 6.1, 6.2 and 6.3 of the appendix. Protocol:

1. Download RFAM dataset from <http://eddylib.org/publications/RivasEddy16/RivasEddy16-supplement.tar.gz> (01.10.2017)
2. Extract RFAM seed families from `R-scape_Testset.sto`
3. Remove alignments longer than 500 nt
4. Subsample every RFAM family to create 20 sub-alignments with 10, 20 and 30 sequences per family using `random-sample-fasta.pl -n XX` ^a
5. Remove gap-only columns from sub-alignments
6. Replace U with T nucleotides
7. Shuffle all sub-alignments using `multiperm` (v.0.9.4) and `SISSIZ` simulation
8. Run `SISSIZ` (v.2.0 and update) and `RNAz` (v.2.1) on native and shuffled sub-alignments

^a<https://github.com/ranjit58/NGS/blob/master/random-sample-fasta.pl>

In contrast to the benchmark of the runtime in the previous section, the RFAM families were not only sampled to 10 sequence alignments but also to alignments with 20 and 30 sequences. Each of the 88 RFAM families is again represented by 20 sub-sampled alignments. No

realignment was performed, i.e. the native RFAM alignments were used. Alignment columns that contained only gaps were removed as they appeared to negatively impact the performance of RNAz. The replacement of U nucleotides by T nucleotides in the native alignments was necessary to ensure correct shuffling by **multiperm** (chapter 2.4). The sampled alignments constitute the positive control dataset of the benchmark. For the generation of the negative control dataset two methods were used and compared:

1. **multiperm**
2. **SISSIZ -s** (SISSIZ alignment simulation)

Then both prediction programs were employed on the native and shuffled/simulated alignments: **SISSIZ** was then run with the following command line parameters: `--mono --ribosum, --di --ribosum, --mono` and `--di` (default). The performance was compared to **RNAz** which was run with default command line parameters.

Results

To start with, it has to be pointed out that the second method for the generation of the negative control dataset with **SISSIZ**'s simulation is not without concerns, as the same module is used by **SISSIZ**'s gene finder for the calculation of the z-score. And a program should not be benchmarked by itself. However, since **multiperm** does not destroy the information on covariation in alignments as pointed out in chapter 2, **SISSIZ** was used as second method for the creation of the negative control just for comparison.

The results of the benchmark with **multiperm** as the negative control are shown in figure 3.6. In general, the new **SISSIZ** version performs just slightly, if at all, better than the old version of **SISSIZ**. If **SISSIZ** is used with the ribosum matrix the ROC curve improves too compared to **SISSIZ** in default mode, as can be seen in figure 3.8 which shows the area under the curve (AUC) of the ROC curves (although the AUC might not be the best measure if only the performance at low FPRs is of interest). Notably, the performance of **SISSIZ** declines with increasing number of sequences. This phenomenon is most likely explained by the **multiperm** negative control dataset, as the information on the covariation in the alignment is not destroyed. Since **RNAalifold** incorporates covariation in the prediction of the MFE, far negative **SISSIZ** z-scores are still possible.

Moreover, the di-nucleotide model of **SISSIZ** did not always appear to work better than the mono nucleotide model. This finding would contradict the hypothesis that thermodynamic predictions are biased by the di-nucleotide content. Nevertheless, the benefits of background

model also seem to depend on the RFAM family as the average z-scores per family differ depending on whether the mono or di nucleotide model is used (appendix figures 6.1-6.3). Thus, the choice of the background model may dependent on sequence composition or other qualities, a hypothesis that requires further investigation.

The results of the benchmark with **SISSIZ** simulated alignments are shown in figure 3.8. In this case the performance remains almost identical, independently from the number of sequences, probably because **SISSIZ** simulations do not preserve covarying columns. According to the second method **SISSIZ** outperforms **RNAz** for 10, 20 and 30 sequence alignments if the ribosum matrix is used, but not in default mode. With the first method (**multiPerm** negative control) **SISSIZ** performs better only on 10 sequence alignments and only if the ribosum matrix is used. This raises the question whether **SISSIZ** should ever be used in the default mode. According to Smith et al. (2013) the default mode of **SISSIZ** is favorable over the ribosum mode if the GC content is below 60% and the MPI over 60%. Furthermore, they found that **SISSIZ** performs better than **RNAz** at MPI below 85% and vice versa.

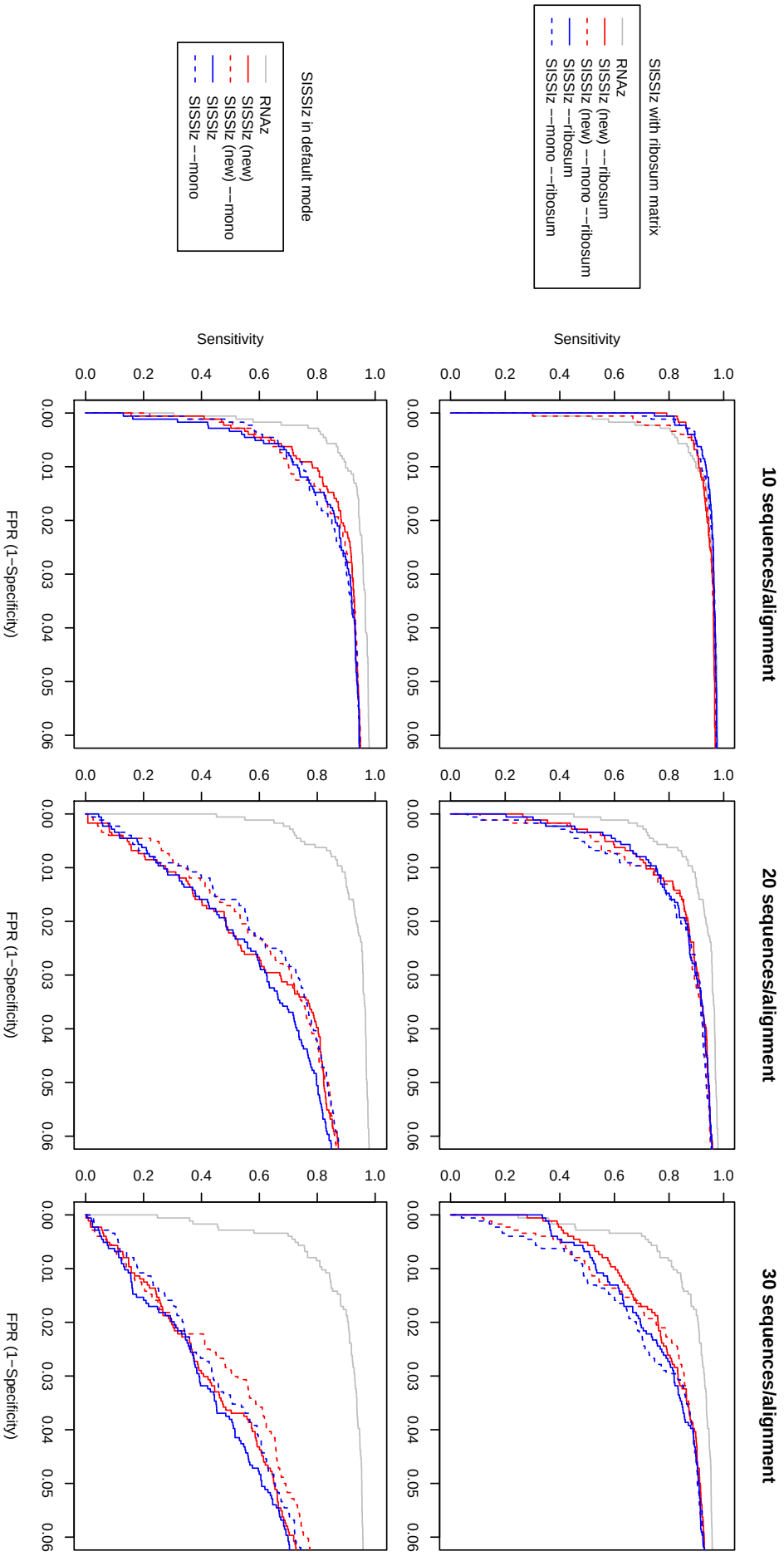


Figure 3.6: **Benchmark: SISSiZ old vs. new - multiperm as negative control:** The plots show the performance of SISSiZ (v.2.0), the updated SISSiZ version (new) and RNAz in dependence of the alignment size (10, 20 and 30 sequences/alignment). In the first row of the plots SISSiZ was used with the ribosome matrix and in the second row with default scoring. To each column of plots corresponds a dataset of 1760 sampled RFAM alignments from 88 families. Each family is represented by 20 alignments with 10, 20 or 30 sequences. The native alignments constituted the positive control dataset. `multiperm` shuffled alignments were used as the negative control.

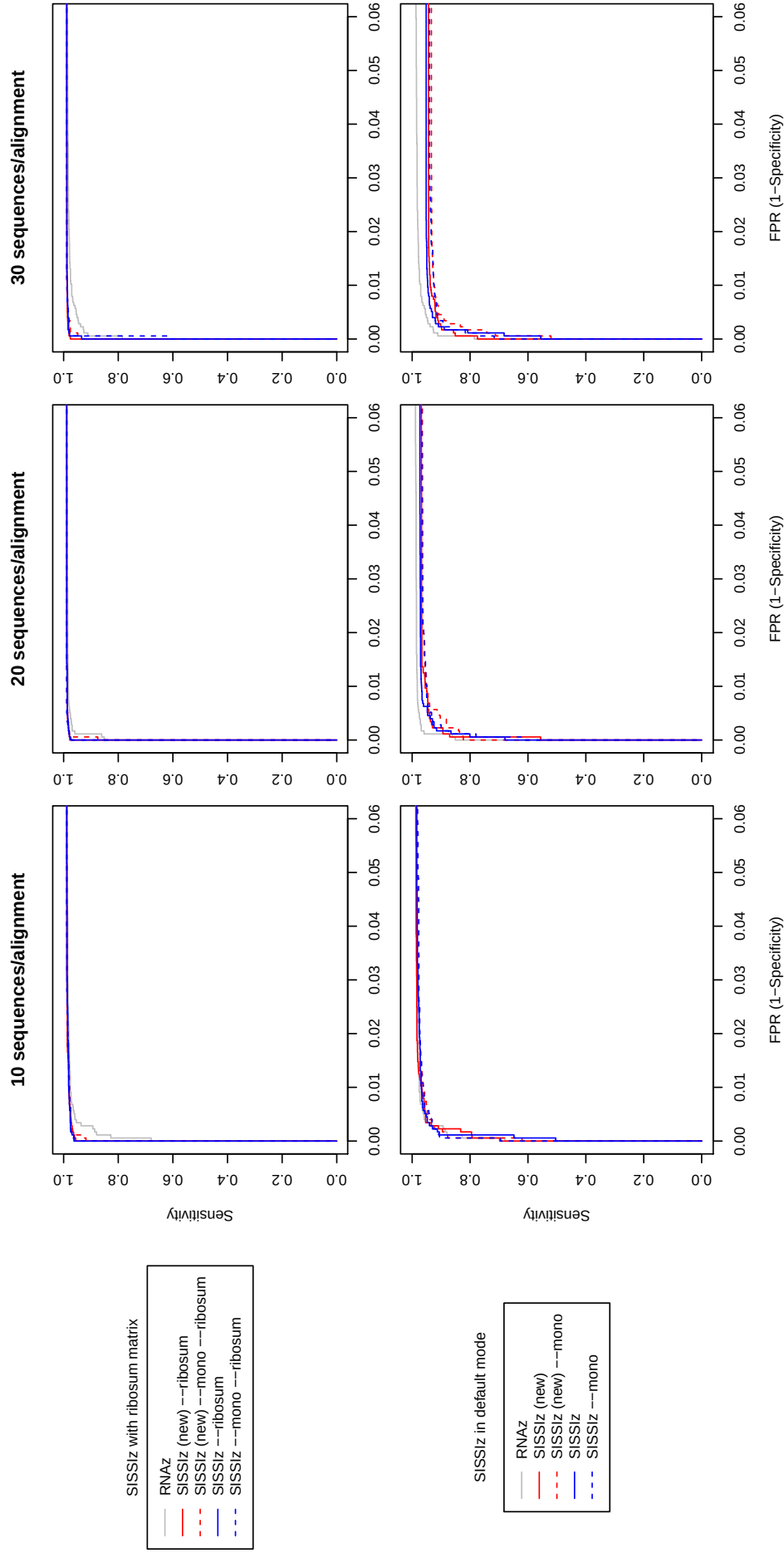


Figure 3.7: **Benchmark: SISSiZ old vs. new - SISSiZ simulation as negative control:** The plots show the performance of SISSiZ (v.2.0), the updated SISSiZ version (new) and RNAZ in dependence of the alignment size (10, 20 and 30 sequences/alignment). In the first row of the plots SISSiZ was used with the ribosum matrix and in the second row with default scoring. To each column of plots corresponds a dataset of 1760 sampled RFAM alignments from 88 families. Each family is represented by 20 alignments with 10, 20 or 30 sequences. The native alignments constituted the positive control dataset. SISSiZ -s simulated alignments were used as the negative control.

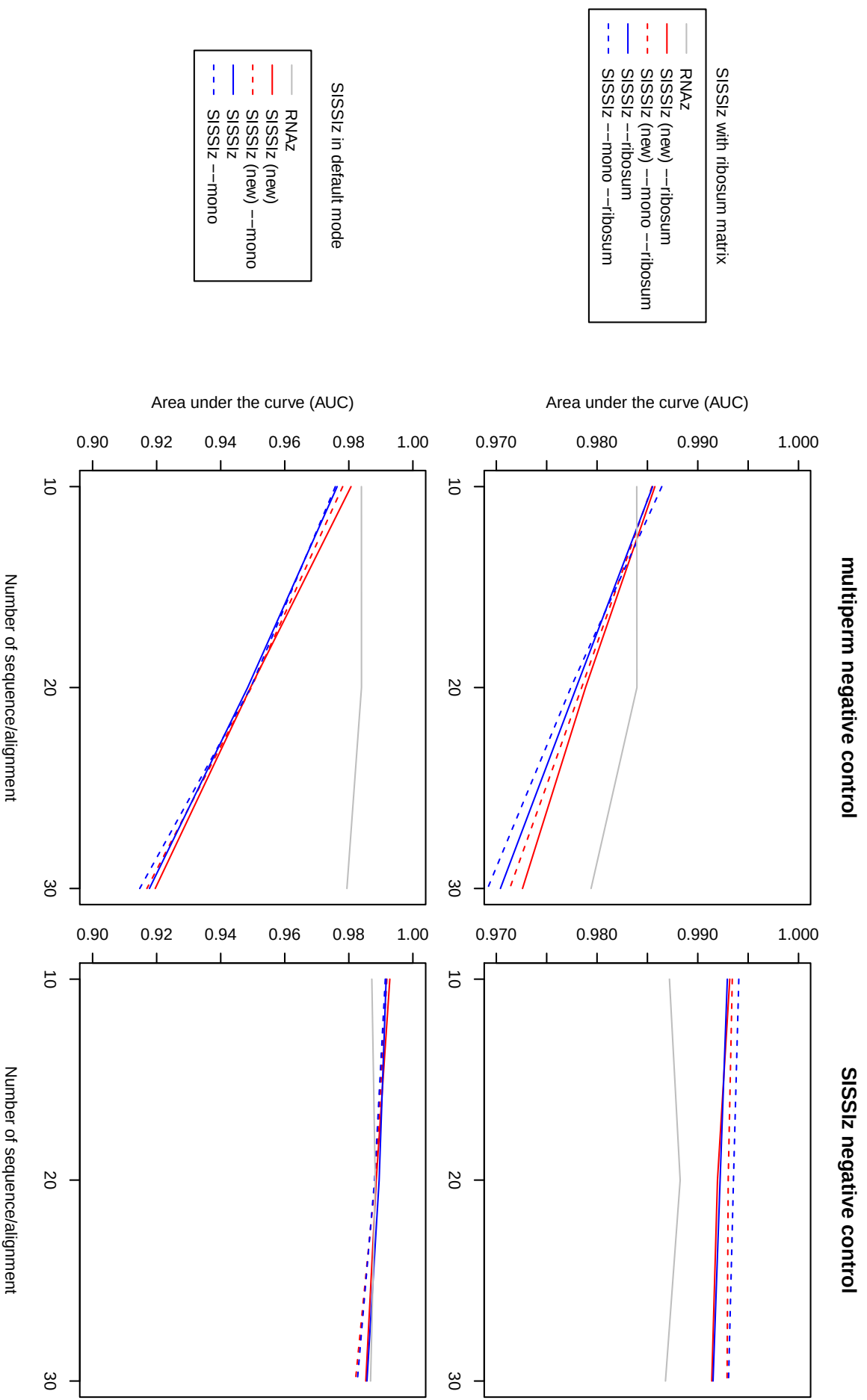


Figure 3.8. **Benchmark: SISSiz old vs. new - AUC:** The plots show the AUC (FPR from 0 to 1) of the ROC curves in figures 3.6 and 3.7 for SISSiz (v.2.0), the updated SISSiz version (new) and RNAz in dependence of the alignment size (10, 20 and 30 sequences/alignment). In the first row of the plots SISSiz was used with the ribosum matrix and in the second row with default scoring. The first column shows the results for **multiPerm** as the negative control and the second column the results for **SISSiz -s** simulated alignment, respectively

3.3.2 The SCI and SISSIZ

Introduction

The updated version of SISSIZ can calculate the SCI according to equation 3.4 used in RNAz by Washietl et al. (2005) and print it to the output (command line option: `--sci`). Since RNAz uses the SCI in addition to the z-score which is, however, calculated differently, to make it's prediction more accurate, it was of course interesting whether SCI could also be used to improve the predictions of SISSIZ. To this end, the z-scores and SCI values from the positive and negative control of the previous benchmark were plotted against each other (figure 3.9). It can be observed that z-score alone does not always suffice to separate the negative and positive control datapoints. This is especially pronounced when SISSIZ is run with default command line parameters. However, in any case the prediction performance of SISSIZ could be definitely improved if the SCI was included in the prediction. The best solution would probably be to use a support vector machine, like in RNAz to find the optimal parameters for SCI and z-score separation. However, this is out of the scope of this thesis. Nevertheless, the potential of the SCI to improve the predictions of SISSIZ could be demonstrated by a simple model.

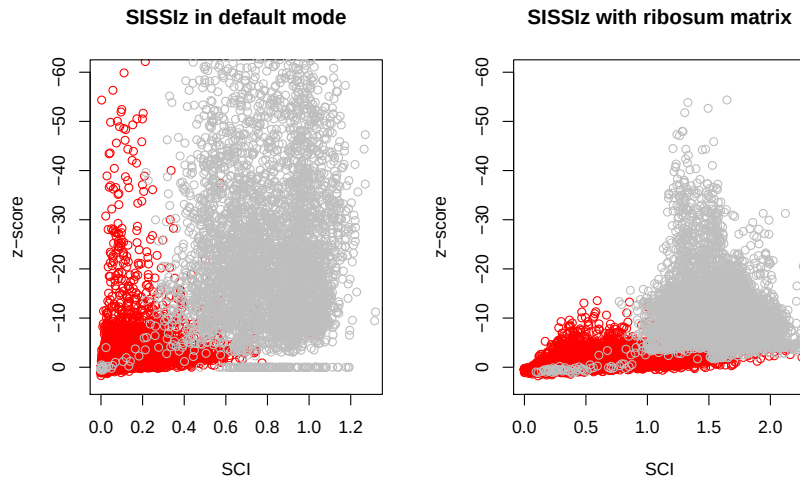


Figure 3.9: **Relationship between z-score and SCI:** The plot shows the relationship between the SCI and the SISSIZ z-score. Each dot represents the results from one prediction. gray dots represent data points from the positive control dataset and red dots data points from the negative control dataset. The positive control dataset contained all 10, 20 and 30 sequence alignments from the previous benchmark and the negative control dataset their respective `multiperm` shuffled counterpart. Each class is represented by 5280 data points. In the left plot the new SISSIZ version was run with default command line parameters and in the plot on the right site with the ribosum scoring matrix.

Results

Based on the figure 3.9, the following threshold values for the SCI were chosen: 0.4 for **SISSIZ** with default command line parameters and 0.8 for **SISSIZ** with ribosum matrix. If the SCI value of a prediction was below this threshold value the z-score was automatically set to 0. By this razor approach the ROC curve could be substantially improved as can be seen in figure 3.10. Note, that the same dataset as in the previous benchmark runs with **multiPerm** as the negative control was used. With this SCI adaption in the new **SISSIZ** version the old version of **SISSIZ** is outperformed in every case. Most of all the decline of performance with increasing number of sequences could be halted. This is due to the fact that the SCI in **multiPerm** shuffled alignments is relatively low as generally anticipated for unstructured RNAs. Tables 3.3 and 3.4 give a summary of the two benchmarks. The first table shows the sensitivity at fixed FPRs of 0.05 and 0.01 and shows again that the sensitivity decreases with increasing number of sequences per alignment. Nevertheless, the relationship is much weaker in **RNAz** and if the SCI is used in **SISSIZ**, though sensitivities decreases slightly too. The second table shows the corresponding predictor thresholds at fixed FPRs. For both **SISSIZ** and **RNAz** it can be observed that the thresholds for the z-score and RNA class probabilities become more stringed the more sequences the alignment contained. The table thus explains why the respective sensitivities are so low. However, if **SISSIZ**'s z-score is combined with the SCI the effect is much less pronounced and the sensitivities are much better at a given FPR.

Sensitivity at fixed FPR

	FPR 0.01			#sequences	FPR 0.05		
	10	20	30		10	20	30
SISSIZ (new) --ribosum +SCI	0.945	0.929	0.907		0.974	0.974	0.974
SISSIZ (new) --mono --ribosum +SCI	0.953	0.929	0.910		0.976	0.976	0.976
SISSIZ (new) --mono +SCI	0.934	0.898	0.838		0.977	0.973	0.954
SISSIZ (new) +SCI	0.936	0.858	0.843		0.981	0.974	0.960
SISSIZ (new) --ribosum	0.914	0.756	0.623		0.967	0.961	0.944
SISSIZ (new) --mono --ribosum	0.914	0.751	0.525		0.969	0.959	0.940
SISSIZ (new) --mono	0.706	0.363	0.158		0.940	0.870	0.729
SISSIZ (new)	0.802	0.293	0.167		0.940	0.857	0.696
SISSIZ --ribosum	0.936	0.766	0.545		0.973	0.964	0.946
SISSIZ --mono --ribosum	0.920	0.761	0.496		0.972	0.961	0.941
SISSIZ --mono	0.754	0.369	0.195		0.939	0.863	0.697
SISSIZ	0.738	0.290	0.157		0.938	0.831	0.652
RNAz	0.906	0.888	0.834		0.976	0.976	0.966

Table 3.3: **Sensitivity at fixed FPR:** Sensitivity at fixed FPR corresponding ROC curves in figures 3.6 and 3.10. Green cells indicate a high sensitivity and red cells a low sensitivity, respectively.

Predictor thresholds at fixed FPR

	FPR 0.01			#sequences	FPR 0.05		
	10	20	30		10	20	30
SISSlZ (new) --ribosum +SCI	-3.72	-4.01	-4.47		-2.57	-2.54	-2.54
SISSlZ (new) --mono --ribosum +SCI	-3.72	-4.26	-4.61		-2.79	-2.72	-2.82
SISSlZ (new) --mono +SCI	-5.78	-7.1	-9.12		-2.9	-3.53	-4.64
SISSlZ (new) +SCI	-5.24	-8.31	-8.57		-2.86	-3.36	-4.1
SISSlZ (new) --ribosum	-4.4	-6.05	-6.97		-3.07	-3.4	-3.74
SISSlZ (new) --mono --ribosum	-4.5	-6.48	-8.46		-3.25	-3.56	-4
SISSlZ (new) --mono	-12.87	-25.22	-46.12		-5.44	-8.28	-12.3
SISSlZ (new)	-9.52	-27.39	-41.44		-5.1	-8.36	-12.17
SISSlZ --ribosum	-4.25	-6.2	-8.13		-3.27	-3.58	-4.04
SISSlZ --mono --ribosum	-4.74	-6.73	-9.08		-3.31	-3.87	-4.36
SISSlZ --mono	-12.01	-26.3	-41.56		-5.82	-8.51	-13.68
SISSlZ	-11.19	-29.05	-46.3		-5.49	-9.1	-13.52
RNAz	0.856	0.915	0.969		0.050	0.057	0.147

Table 3.4: **Predictor thresholds at fixed FPR:** Predictor thresholds at fixed FPR corresponding ROC curves in figures 3.6 and 3.10. Green cells indicate a high z-score and red cells a low z-score, respectively.

Conclusion

Although the results look promising and demonstrate the great potential of the SCI in SISSlZ predictions, they should be seen with caution as the SCI threshold determined for this dataset might not be suitable for other RNA classes. However, already this simple model yielded excellent results without any SVM optimization. Therefore, the results could very likely be refined to a greater extent.

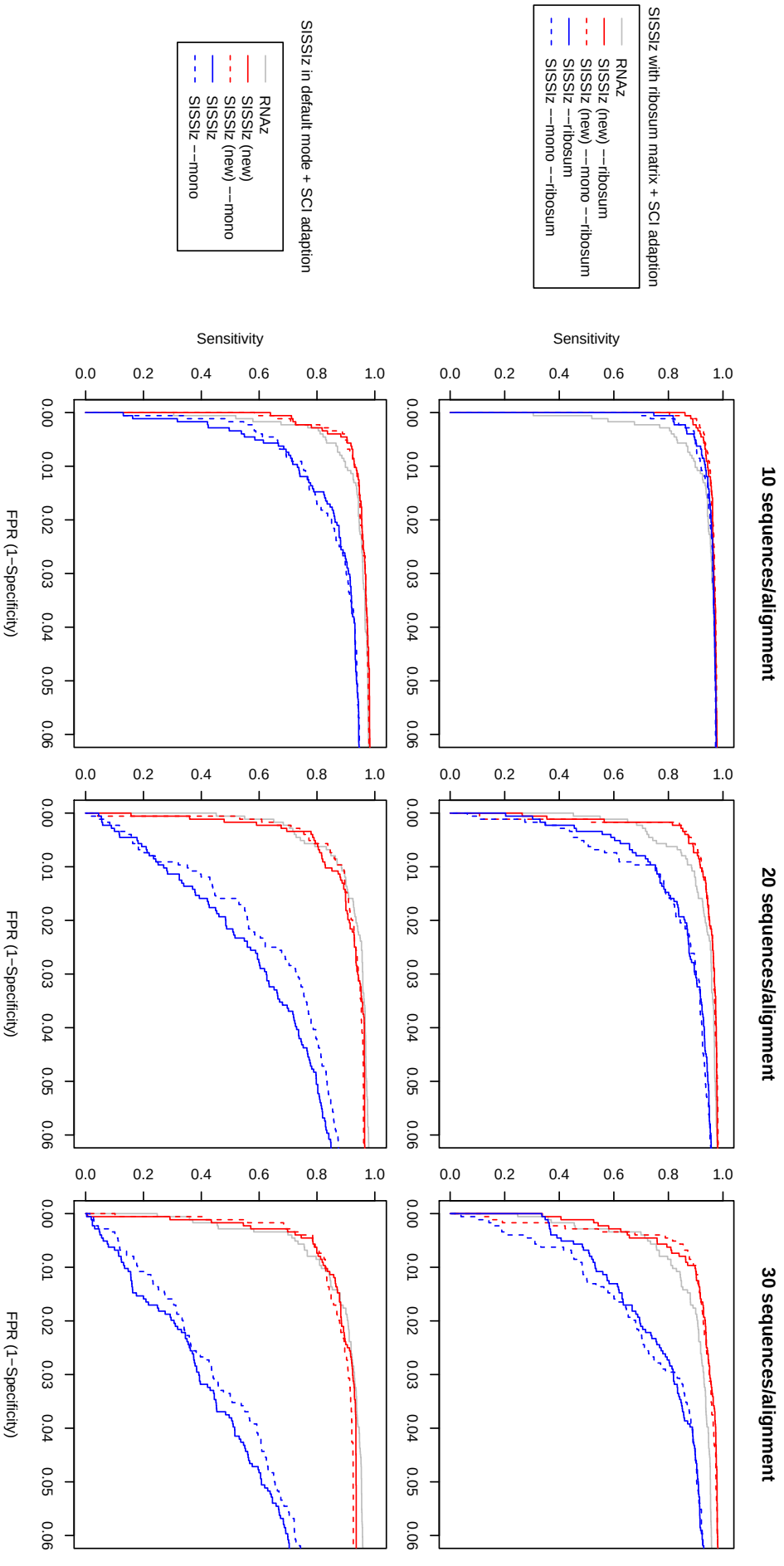


Figure 3.10: **Benchmark: SISSiZ with SCI:** The plots show the performance of SISSiZ (v.2.0), the updated SISSiZ version (new) and RNAz in dependence of the alignment size (10, 20 and 30 sequences/alignment). In the first row of the plots SISSiZ was used with the ribosum matrix and in the second row with default scoring. To each column of plots corresponds a dataset of 1760 sampled RFAM alignments from 88 families. Each family is represented by 20 alignments with 10, 20 or 30 sequences. The native alignments constituted the positive control dataset. `multiPerm` shuffled alignments were used as the negative control. Additionally, the z-score of predictions with a SCI below 0.4 for SISSiZ in default mode and below 0.8 for SISSiZ predictions with ribosum matrix was set to 0 in the new SISSiZ version (SCI adaption).

3.3.3 The impact of alignment programs

As the alignment is the basis of every SISSIz prediction it was interesting to analyze the impact of alignment programs on the performance of SISSIz.

Methods

To this end, the same benchmark dataset from the previous section was used. However, the native alignments were realigned with `clustalw2` (Larkin et al., 2007), `mafft` (Katoh and Standley, 2013), `tcoffee` (Notredame et al., 2000) and `clustalo` (Sievers and Higgins, 2014). `multiperm` was used again to create the negative control dataset.

1. Download RFAM dataset from <http://eddylib.org/publications/RivasEddy16/RivasEddy16-supplement.tar.gz> (01.10.2017)
2. Extract RFAM seed families from *R-scape_Testset.sto*
3. Remove sequences longer than 500 nt
4. Subsample every RFAM family to create 20 sub-alignments with 10, 20 and 30 sequences per family using `random-sample-fasta.pl -n XX` ^a
5. Realign with `clustalw2` (Larkin et al., 2007), `mafft` (Katoh and Standley, 2013) and `t-coffee` (Notredame et al., 2000) and `clustalo` (Sievers and Higgins, 2014)
6. Replace U with T nucleotides
7. Shuffle all sub-alignments using `multiperm`
8. Run the updated SISSIz version and RNAz on native and shuffled sub-alignments

^a<https://github.com/ranjit58/NGS/blob/master/random-sample-fasta.pl>

Results

At 10 sequences per alignment a clear pattern can be observed no matter whether SISSIz is used with or without ribosum matrix (Figure 3.11): The performance of SISSIz is best with the native RFAM alignments, second best with `t-coffee`, third best with `mafft` and worst with `clustalw2` and `clustalo`. While `clustalo` leads to the worst performance of SISSIz independently from the number sequences per alignment, `mafft` and `t-coffee` manage to improve the ROC curve at 20 and 30 sequences/alignment even above the level of the native RFAM alignments when SISSIz is run in default mode. When SISSIz is used with the ribosum matrix the performance remains best with the native RFAM alignments at 20 sequences; however at 30 sequences per alignment `mafft` und `t-coffee` improve the ROC curve at low FPRs.

Conclusion

In any case, the choice of the alignment program greatly influences the prediction accuracy of SISSIz. In practice it will not always be possible to use the "best", that is most accurate,

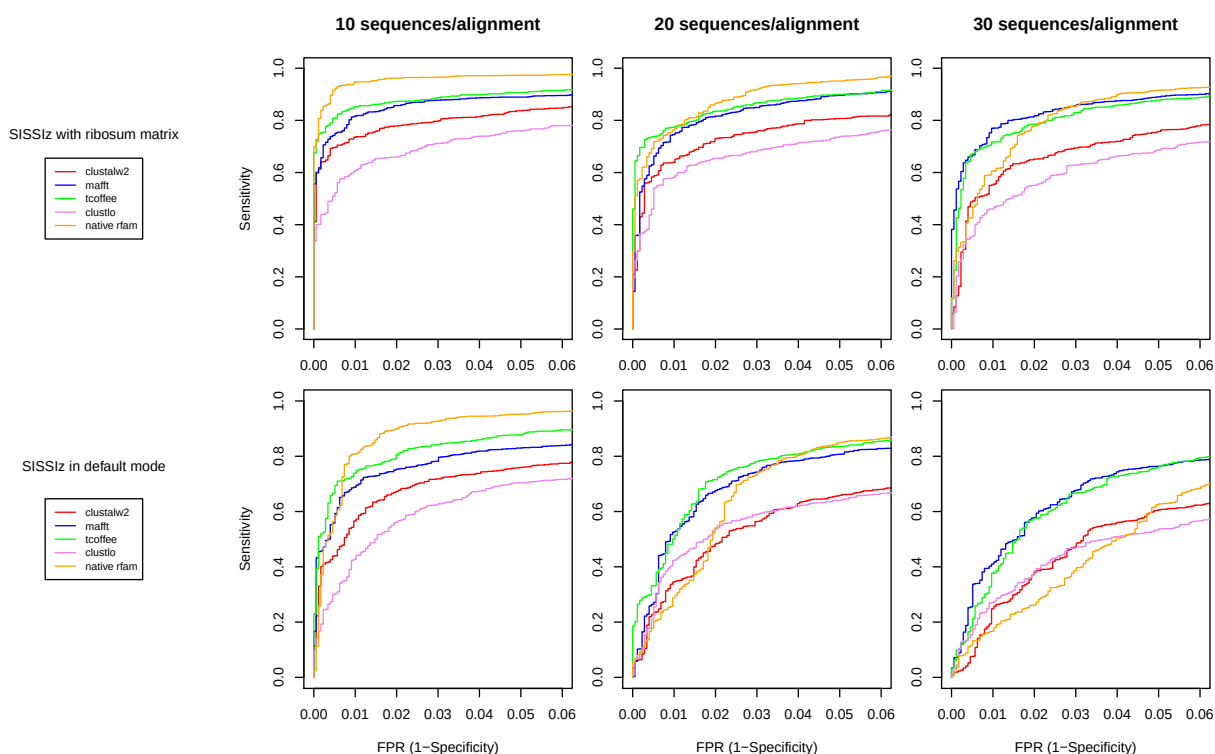


Figure 3.11: **Benchmark: The impact of alignment programs on SISSiz:** The plot shows the performance of SISSiz in dependence of the used alignment method. The same benchmark dataset from the previous section was used. However, the native alignments were also realigned with `clustalw2`, `clustlo`, `mafft` and `tcoffee`. `multiperm` was used again to create the negative control dataset. In the first row SISSiz was used with ribosum matrix and in the second row with default command line parameters.

alignment program, as accuracy comes at the cost of runtime. The best trade-off between performance and runtime of the three tested alignment programs is provided by `mafft`. However, with all sequence based alignment programs the chicken-egg problem between alignment and secondary structure remains. One solution is offered by the Sankoff algorithm (Sankoff, 1985) which solves the alignment and secondary structure simultaneously. The biggest drawback is the algorithm's time complexity $O(n^6)$ (with sequences of length n) rendering it impractical for most applications. LocARNA (Will et al., 2007) offers an improved variant of the Sank-off algorithm and will be applied in chapter 4.1.

3.3.4 SISSIZ with SHAPE

Now it is time to test the performance of SISSIZ with SHAPE data. To this end, advantage of an existing benchmark dataset has been taken: Lorenz et al. (2016) used a dataset² of 24 sequences and SHAPE files to analyze the effect of probing data files on the prediction of RNA secondary structures. While a similar dataset was used by Deng et al. (2016), both refer to the original dataset by Hajdin et al. (2013).

Methods

Of the 24 sequences, 12 sequences to which a corresponding RFAM (Nawrocki et al., 2015) family was available were selected for this benchmark. Table 3.5 shows the represented sequences.

Reference Sequence	Length	Accession	RFAM ID	#Seq.
5S rRNA, E. coli.fa	120	RF00001	5S_rRNA	712
tRNA(asp), yeast.fa	75	RF00005	tRNA	954
tRNA(phe), E. coli.fa	76			
cyclic-di-GMP riboswitch, V. cholerae.fa	97	RF01051	c-di-GMP-I	155
RNase P, B. subtilis.fa	401	RF00011	RNaseP_bact_b	114
SAM I riboswitch, T. tengcongensis.fa	118	RF00162	SAM	433
Lysine riboswitch, T. maritime.fa	174	RF00168	Lysine	47
Signal recognition particle RNA, human.fa	301	RF00017	Metazoa_SRP	91
Group I Intron, T. thermophila.fa	425	RF00028	Intron_gpI	12
Group I intron, Azoarcus sp..fa	214			
Group II intron, O. iheyensis.fa	412	RF00029	Intron_gpII	92
Pre-Q1 riboswitch, B. subtilis.fa	34	RF00522	PreQ1	35
TPP riboswitch, E. coli.fa	79	RF00059	TPP	109
Hepatitis C virus IRES domain.fa	336	RF00061	IRES_HCV	79

Table 3.5: RFAM families and reference sequences used in the benchmark test. For the reference sequences, shape data was available from <https://github.com/stanti/shapebenchmark>

The following protocol was applied to the dataset:

1. Download RFAM seed sequences from <ftp://ftp.ebi.ac.uk/pub/databases/Rfam/12.3/Rfam.seed.gz> and extract the families described in table 3.5
2. Convert Stockholm format to fasta: `stockholm2fasta.perla`
3. Randomly sample 3, 6, 12 and 24 sequences from each RFAM family to create 300 fasta files per RFAM family (no replacement): `random-sample-fasta.pl -n XXb`
4. Add the reference sequences corresponding to the RFAM family to each sampled fasta file at the first position
5. Create positive control dataset: Align all fasta files with `clustalo --seqtype=RNA --outfmt=clu --output-order=input-order --full` (v.1.2.4, Sievers and Higgins (2014))
6. Create negative control dataset: Shuffle all alignment files from the positive control

²<https://github.com/stanti/shapebenchmark>

dataset with `multiperm` (v.0.9.4, Anandam et al. (2009)) + create 300 shuffled shape files per RFAM family: self written R `script`, permutes the order of reactivity values of a given input file.

^a<https://gist.github.com/venuthatikonda/e0233043ba77c5ce48aa>

^b<https://github.com/ranjit58/NGS/blob/master/random-sample-fasta.pl>

For every RFAM family 300 alignments were sampled with 3,6,12 and 24 sequences, additionally the SHAPE reference sequence(s) was/were added to the sampled alignments. The alignment of the sequences was performed with `clustalo` (Sievers and Higgins, 2014); even though it did not perform well in the previous benchmark `clustalo` alignments yielded better results than `mafft` alignments in the benchmark with SHAPE data. The negative control dataset was created by two methods, both of which are compared in the result section:

1. Use only `multiperm` shuffled alignments as negative control
2. Use `multiperm` shuffled and permuted SHAPE data as negative control

Results

To start with, the results differ drastically, depending on which method was used for the negative control as can be seen in figure 3.12. If only `multiperm` shuffled alignments are used as the negative control the performance of `SISSiZ` with SHAPE is estimated to be much better than in the case where permuted SHAPE data is additionally used in the negative control. With the latter, more conservative method, SHAPE data would only be considered advantageous for alignments with less than 6 six sequences and only if `SISSiZ` was used with the `ribosum` matrix. Conversely, with the first method SHAPE data can be seen beneficial in any case, independently of the number of sequences and (other) `SISSiZ` parameters. Nevertheless, the impact of SHAPE data declines with increasing number of sequences as the relative weight of the SHAPE data diminishes slightly.

Discussion

Therefore, SHAPE data is most recommend for alignments with few sequences, unless SHAPE data for more than one sequence is available. The drawback of this benchmark is undoubtedly the relatively small benchmarking dataset, as only one or at most two SHAPE dataset(s) was/were available per alignment. The RMDB (RNA Mapping Database, Cordero et al. (2012)) would have provided much more probing data. However, the structure of the database would have made the benchmarking too complex, thus the RMDB could not be used for time reasons. Additionally, the question on the best method for the generation of the negative control dataset arises. Simple permutation of the SHAPE reactivities as performed in this benchmark could still

lead to significantly low MFEs. Another solution would be to create simulated null SHAPE data which may be possible by adapting the approach of Sukosd et al. (2013) to model experimental SHAPE data.

However, independently from the negative control one could question whether the z-score is at all a valid measure in **SISSIZ** itself if SHAPE data is used. Internally, **SISSIZ** simulated null alignments do not receive any pseudo energy contributions while the native alignment does. The pseudo energies contributions from the SHAPE data could artificially decrease the MFE calculated by **RNAalifold**. Thus, it could be argued that the MFE has become a pseudo MFE that is no longer representative of the consensus structure and serves the sole purpose of guiding **RNAalifold** towards that structure. This could in the end be illusive for the calculation of the z-score in **SISSIZ**.

Last but not least, the SHAPE data can even worsen predictions if the data does not match the sequence or alignment well. This observation was already made by Lorenz et al. (2016) who found that SHAPE data did not improve the prediction accuracy of **RNAalifold** for every RNA family. Probing data can be seen analogous to priors in Bayesian networks; if estimated badly the prediction will be bad too.

Conclusion

It could be demonstrated that the performance of **SISSIZ** improves if SHAPE data is used, especially if just a small number of sequences is present in the alignment. Of course, the degree of improvement will always depend on the quality of the experimental data and may work better for certain RNA classes than others. Furthermore, the results greatly depend on what negative control is used and whether the z-score can be still be seen as a valid predictor, so that a definitive conclusion to the benefits of SHAPE data cannot be drawn without reservations. Nevertheless, the potential advantages of using SHAPE data in **SISSIZ** are very high.

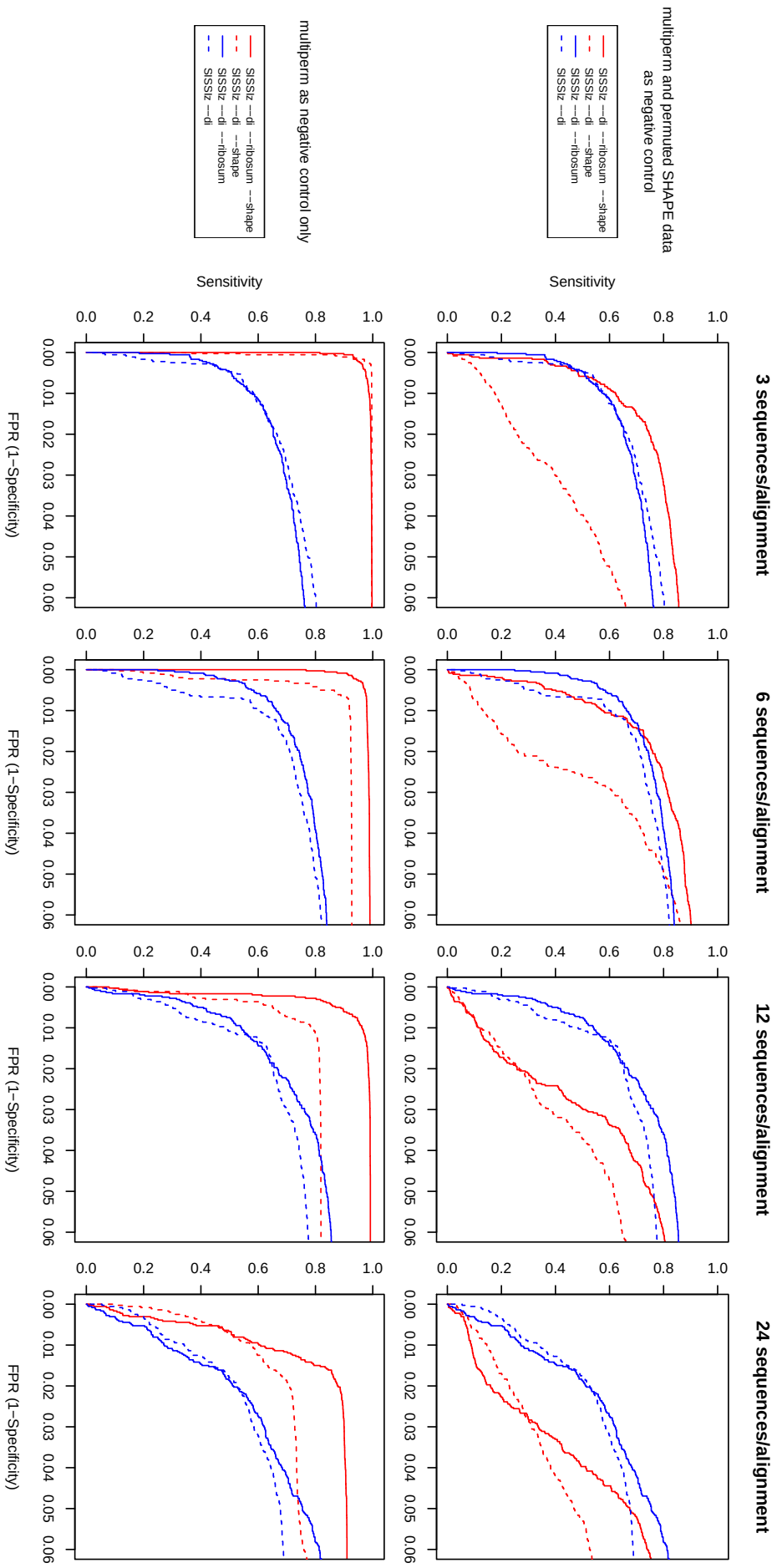


Figure 3.12: **Benchmark: SISSiz with SHAPE data:** SISSiz was run on a set of 12 Rfam families with the command line parameters given in the figure's legend. Each Rfam family is represented by 300 alignments consisting of 3,6,12 or 24 sampled Rfam sequences + one or two reference sequence(s) provided in table 3.5. Therefore, each reference sequence is present in all 300 alignments of a family. This was necessary to run the new SISSiz version with the shape files which were only available for the reference sequences. ROC curve calculation: The sensitivity was calculated from the positive control dataset, the specificity from the negative control dataset, respectively. For the calculation of the specificity two methods were used: In the first row of the plots: `multiperm` shuffled alignments were utilized as the negative control along with permuted SHAPE data. In the second row only `multiperm` shuffled alignments were used as the negative control.

Chapter 4

Application: Screening for non-coding RNAs

After all the benchmarking it is time to apply the updated **SISSIz** version in real ncRNA screens and to compare its performance to **RNAz**. To this end, two genomic screens were conducted: While the first one aims to find differences in the predictions of **SISSIz** and **RNAz** on the *Drosophila* X-chromosome, the second focuses on the benefits of SHAPE data in a screen on the HIV genome.

4.1 *Drosophila* Screen

4.1.1 Introduction

Since *Drosophila* is a popular model organism it has been the target of many ncRNAs screens, e.g.: Rose et al. (2007) who conducted an **RNAz** screen or Bradley et al. (2009) who relied on **EvoFold** predictions. More recently, Will et al. (2013) presented an **RNAz** pipeline for the screening of ncRNAs, called REAPR (RE-Alignment for Predictions of structural ncRNA) and tested it on a *Drosophila* alignment of 12 sequences. The crux of their pipeline was to prefilter the overlapping whole genome alignment windows based on thermodynamic stability (**RNAz** z-score). While "stable" loci are realigned with **LocARNA** (Will et al., 2007) "unstable" loci are removed. Eventually, **RNAz** is employed on the realigned sequences. Even though, it would be possible to adapt the pipeline to use **SISSIz** instead of **RNAz**, a self written pipeline is employed during this screen since the filtering approach in the pipeline of Will et al. (2013) lacked independence from **RNAz** and could therefore cause a bias in the results.

As pointed out by Gardner et al. (2005), sequence based alignment programs perform poorly at sequence identities below 50-60% causing a seemingly lower SCI. Washietl and Ho-

facker (2004) demonstrated that the z-score of sequence based alignment methods is much lower than the z-score of structural alignments at MPIs below 60%. However, at higher MPIs the difference becomes insignificant. Therefore, structure based realignment could improve the result of ncRNA prediction programs at low MPIs, especially if they are utilizing information on base-pair covariation. Hence, the pipeline for this screen divides the sliced alignment windows into two groups based on the MPI and realigns the group with low MPI ($< 60\%$) using **mlocarna**, a LocARNA version adapted for structural multiple sequence alignments. Since the runtime behavior of **mlocarna** is still suboptimal and structural realignment leads to no improvement at higher MPIs, only the subgroup of alignments with a MPI $< 60\%$ are realigned. Nevertheless, alignments with a MPI $> 60\%$ were realigned with the progressive and sequence based aligner **mafft** (Katoh and Standley, 2013) in order to compensate for errors in the initial **multiz** (Blanchette et al., 2004) alignments.

4.1.2 Methods

With the idea of a potential collaboration with a group studying the evolution of *Drosophila* sex chromosomes, a X-chromosome alignment of 27 fly species from the UCSC genome browser (Tyner et al., 2017) was chosen for this screen. To be more specific the alignment contained sequences from 23 *Drosophila* species, the house fly, anopheles, honey bee and the red flour beetle, with *D. melanogaster* as the reference sequence.

In the following the individual steps of the ncRNA screen are described, while figure 4.1 provides a more schematic overview of the pipeline:

1. Download **multiz** 27 way alignment from UCSC^a (Tyner et al., 2017)
2. Slice maf alignment blocks using **rnazWindow.pl** (Gruber et al., 2010) with the opt.:
`--min-seqs=3 --max-seqs 27 --min-length 30 --min-id 10 -w 200 -s 100`
3. Realign windows with a MPI $< 60\%$ using **mlocarna** (Will et al., 2007) with default command line parameters (the windows were also separately realigned with **mafft** in order to compare the two methods)
4. Realign windows with a MPI $> 60\%$ using **mafft** (Katoh and Standley, 2013) with the following command line parameters: `--auto --clustalout --inputorder --preserveCase`
5. Shuffle all windows with **multiperm** (Anandam et al., 2009) (U nucleotides were replaced by Ts before shuffling)
6. Run the updated **SISSiZ** and **RNAz** (v.2.1) (Gruber et al., 2010) on all native and shuffled windows
7. Calculate thresholds for predictor variable (z-score and RNA class probability for **SISSiZ** and **RNAz**, respectively) at FPR 0.01 and 0.05.
8. Convert all predictions at a given FPR to gff files

9. Compare whether predictions overlap with annotated data^b from flybase (Gramates et al., 2017) and NCBI^c (type: ncRNA, pre_miRNA, miRNA, snoRNA, tRNA, snRNA, rRNA) using `bedtools sort`, `merge` and `intersect` (Quinlan and Hall, 2010)

^a<http://hgdownload.soe.ucsc.edu/goldenPath/dm6/multiz27way/> (November 30, 2017)

^bftp://ftp.flybase.net/genomes/Drosophila_melanogaster/dmel_r6.03_FB2014_06/gff/dmel-gff_all_scaffolds-r6.03.tar.gz (November 30, 2017)

^c<https://www.ncbi.nlm.nih.gov/nuccore/AE014298.5> (November 30, 2017)

Of the initial 30329 alignment windows, 28429 had a MPI greater or equal than 60% and exactly 1900 windows less than 60%. As a result, realignment with `mlocarna` was necessary for a relatively small fraction (approx. 6%) of the alignments.

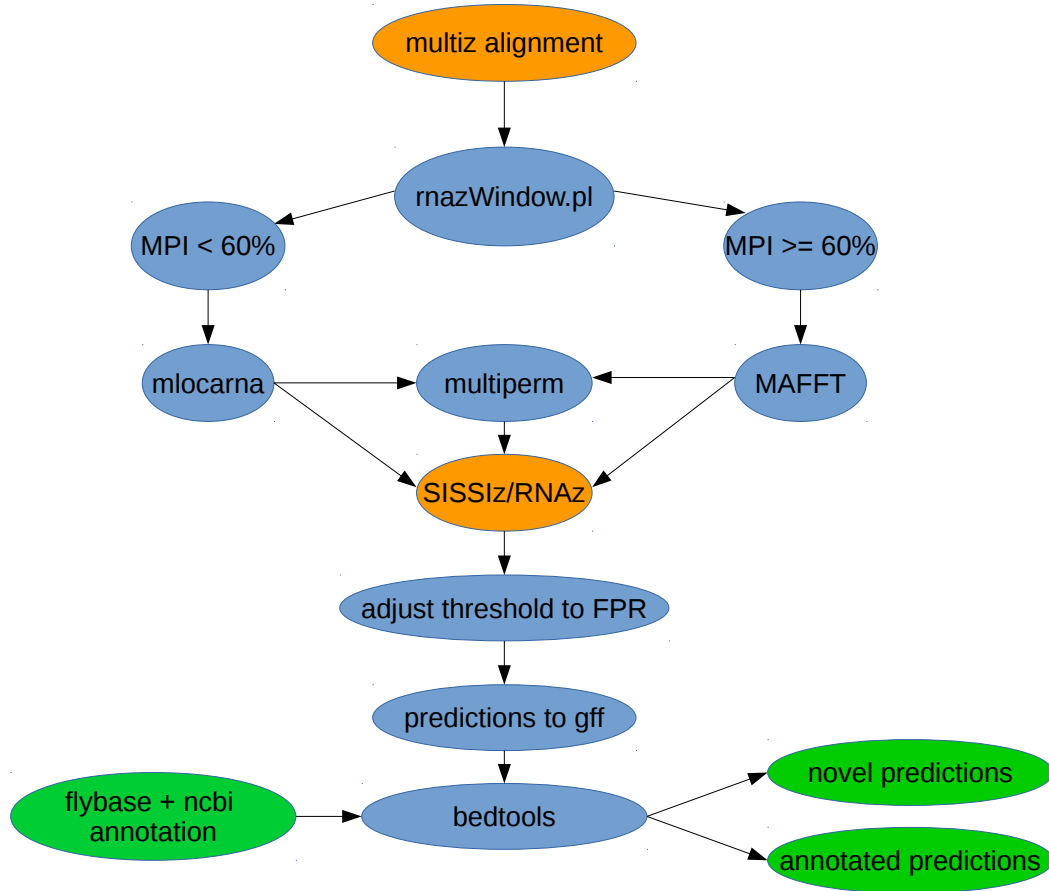


Figure 4.1: **Pipeline of ncRNA screen:** First the 27 way `multiz` alignment blocks are sliced into 200 nucleotide wide and 100 nucleotide overlapping windows by `rnazWindow.pl`. Then the alignments are realigned by `mafft` if their MPI is above 60% or otherwise realigned by `mlocarna` before they are screened by `SISSIZ` and `RNAz`, respectively. Additionally, the native alignments from both groups are shuffled by `multiperm` in order to calculate a predictor threshold value for a given FPR of 1% and 5%, respectively. Thus, the numbers of predictions between `RNAz` and `SISSIZ` are comparable. In the end, the predictions are converted to gff files and compared with annotations (type: ncRNA, pre_miRNA, miRNA, snoRNA, tRNA, snRNA, rRNA) from flybase and ncbi using `bedtools sort`, `merge` and `intersect`.

It would have been without a doubt more meaningful to adjust the thresholds of **SISSiZ** and **RNAz** to the FDR (False Discovery Rate) instead of the FPR. However, the distribution of the **multiPerm** shuffled alignments was too close the distribution of the native alignments. Thus, it was not possible to determine reasonable thresholds values applying by a p-value adjustment according to Benjamini and Hochberg (1995) as performed by Will et al. (2013) in their *Drosophila* screen. Thus and just for the sake of making the results between **SISSiZ** and **RNAz** comparable the threshold were simply set to a fixed FPR. Since the **multiPerm** shuffled dataset is assumed to contain no structurally conserved RNAs it is possible to determine a threshold value for the z-score, in the case of **SISSiZ**, and the RNA class-probability, in case of **RNAz**, at any desired False Positive Rate (FPR): Let t be the sorted list threshold values¹, then the $FPR(t_i) = i/n$, with $i \in \{1..n\}$ and n as the length of the list. t was sorted in ascendant order for **SISSiZ** and in descendant order for **RNAz**. Table 4.1 shows the calculated threshold values.

		SISSiZ		RNAz	
FPR		0.05	0.01	0.05	0.01
MPI<0.6	MAFFT	-1.55	-2.22	0.000008	0.036329
	mlocarna	-1.67	-2.67	0.000014	0.021917
MPI>0.6	MAFFT	-2.39	-3.2	0.009320	0.293272

Table 4.1: The z-score and RNA-class probability threshold values were determined from a **multiPerm** shuffled negative control for **SISSiZ** and **RNAz**, respectively. The threshold values were calculated separately for alignments with a MPI > 0.6 aligned by **mafft**, alignments with a MPI < 0.6 aligned by **mafft** and alignments with a MPI < 0.6 aligned by **mlocarna**.

4.1.3 Results

The thresholds

For **SISSiZ** all threshold values, aside from one are as stringed as expected from the simple one-tailed z-score to p-value conversion from equation 3.2, but the threshold values are more lenient than expected from the benchmark in the previous chapter (table 3.4). For **RNAz** the thresholds for the RNA class probabilities also appear much lower than expected from the benchmarking. However, this comes as no surprise since the dataset used for the benchmark contained known structural RNAs, thus requiring more stringed thresholds.

Annotated and overlapping predictions

To analyze how the predictions match existing annotation data, annotations for ncRNA (also weakly supported), pre-miRNA, miRNA, snoRNA, tRNA, snRNA and rRNA were compared with the predictions using **bedtools intersect**. Overlapping predictions were before merged

¹z-score or RNA class probability of every prediction in the shuffled dataset

with `bedtools merge`. In figure 4.2 the results are visualized in four Venn diagrams. Every predicted annotation was counted non redundantly, i.e. even if one annotation was predicted by two or more windows it is counted just once. Conversely, every window that did not overlap with an annotation was marked as unannotated. **RNAz**'s predictions contained more annotations than **SISSiZ** at both FPRs and independently from the realignment with **mlocarna**. Interestingly, only a small fraction of the predictions between **RNAz** and **SISSiZ** overlaps ($\approx 10\text{-}20\%$ at a FPR of 0.01) indicating that both prediction programs have differing strengths. Figure 4.3 shows that **RNAz** tends to predict more annotations in the higher MPI region than **SISSiZ**. On the other hand, **SISSiZ** appears to find more annotations at low SCI and MPI ranges. Despite the relatively large number of predictions only around 25% at a FPR of 0.01 and 50% at a FPR of 0.05 of all 279 annotation could be recovered if the predictions of both programs are taken together. The largest proportion of predictions consists of unannotated data which either means that there are more structurally conserved RNAs than known or that the FPR is not as low as estimated. Indeed the FDR is higher then expected from the FPR. However, in the latter case even stricter thresholds would have to be set and the sensitivity would even further decrease so that a meaningful comparison for the programs would not have been possible.

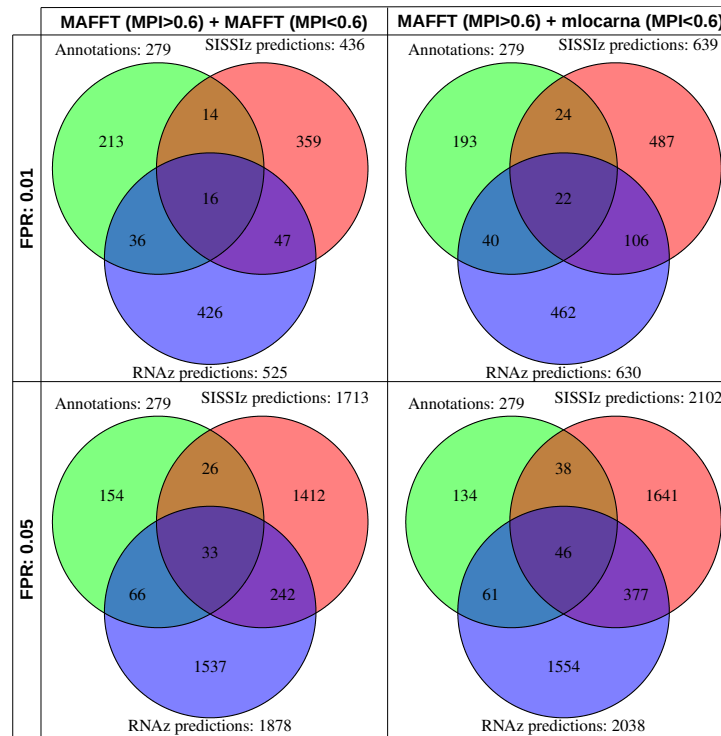


Figure 4.2: **Drosophila screen - Venn Diagrams:** The figure shows the number of flybase and NCBI annotations for ncRNA, pre-miRNA, miRNA, snoRNA, tRNA, snRNA, rRNA that could be detected by **SISSiZ** and **RNAz** respectively. The first column contains the results of the screen with **mafft** realignment of both MPI classes ($> 60\%$ and $< 60\%$). The second column shows the results with **mafft** realignment of the MPI $> 60\%$ class and **mlocarna** realignment of the MPI $< 60\%$ class. The rows correspond to the fixed FPR of 0.01 and 0.05, respectively.

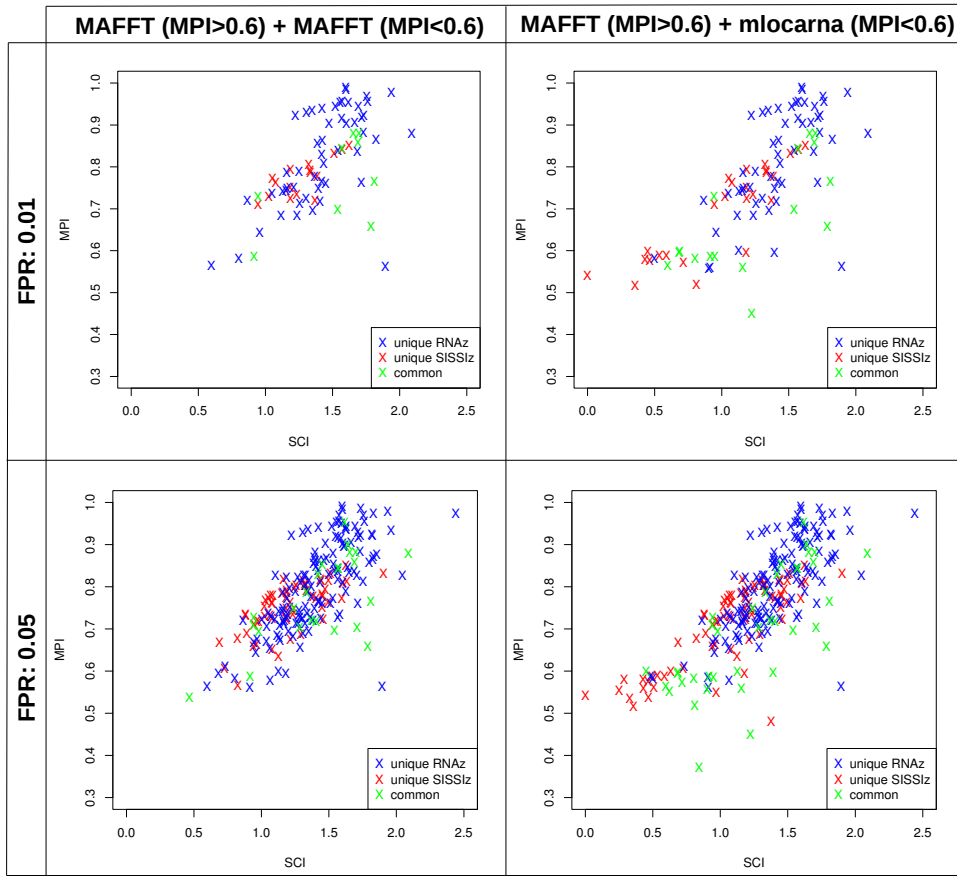


Figure 4.3: **Drosophila screen - statistics of common annotated predictions:** The plots refer to the Venn diagrams in figure 4.2 and shows the relationship between SCI and MPI in the predicted annotated windows. The datapoints were selected from the Venn diagram as follows: Unique RNAz: $(annotated \cup RNAz \text{ predictions}) \setminus (annotated \cup SISSiZ \text{ predictions})$, Unique SISSiZ $(annotated \cup SISSiZ \text{ predictions}) \setminus (annotated \cup RNAz \text{ predictions})$, common: $(annotated \cup RNAz \text{ predictions} \cup SISSiZ \text{ predictions})$

Annotation and predictions in more detail

As shown in figure 4.3 SISSiZ and RNAz tend to make annotated predictions in differing MPI and SCI ranges. Thus, it was of interest how these tendencies impact the type of annotations found. Figure 4.4 depicts the percentage of recovered annotations for every annotation type. Note, that the percentage refers to the fraction of possible annotation types for each MPI class. At a MPI over 60% RNAz manages to predict slightly more annotations from each type than SISSiZ with the exception of the snoRNA (small nucleolar RNA) in which case SISSiZ finds more. Both programs were able to find all annotation of snRNA (small nuclear RNA). At a MPI below $< 60\%$ both programs could recover all annotations for pre miRNA and miRNA independently from the realignment program. However, the realignment with mlocarna could greatly improve the number of predicted ncRNAs. SISSiZ manages to find $\approx 40\%$ of annotated ncRNA at a MPI $< 60\%$ and mlocarna realignment compared to $\approx 20\%$ in the case of RNAz. It

can be concluded that both programs benefit from the realignment with **mlocarna**, especially **SISSiZ**.

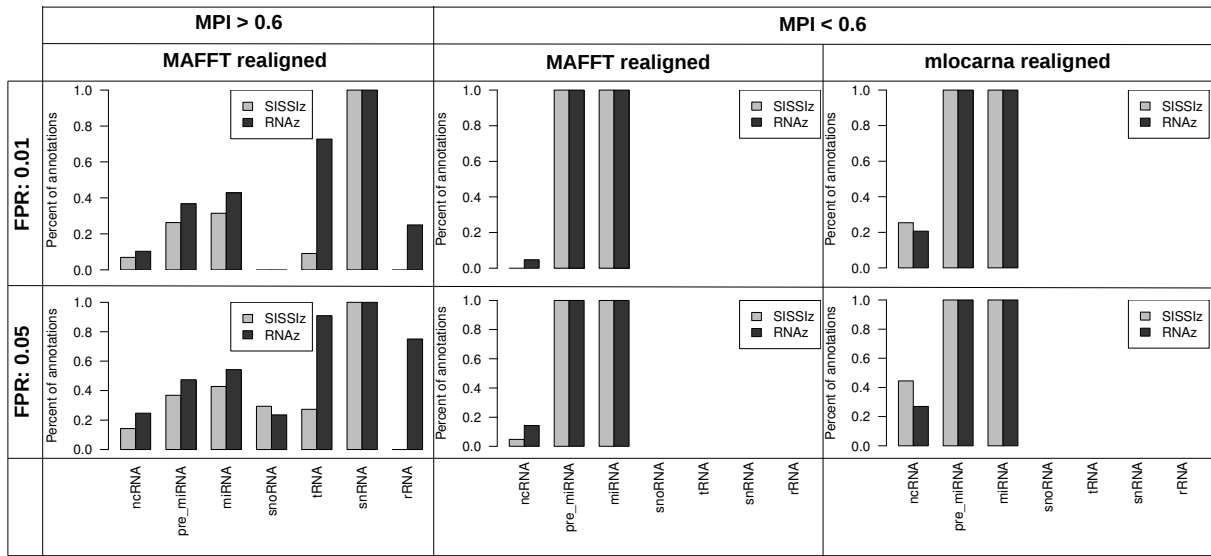


Figure 4.4: **Drosophila screen - Recovered annotation types:** The bar plots show the percentage of recovered annotation types per MPI class for **SISSiZ** and **RNAz**. Alignments with a MPI over 60% were realigned with **mafft** and alignments with a MPI below 60% with **mlocarna** and **mafft** for comparison. The percentage of annotation types refers to the fraction of possible annotation types in the respective MPI class.

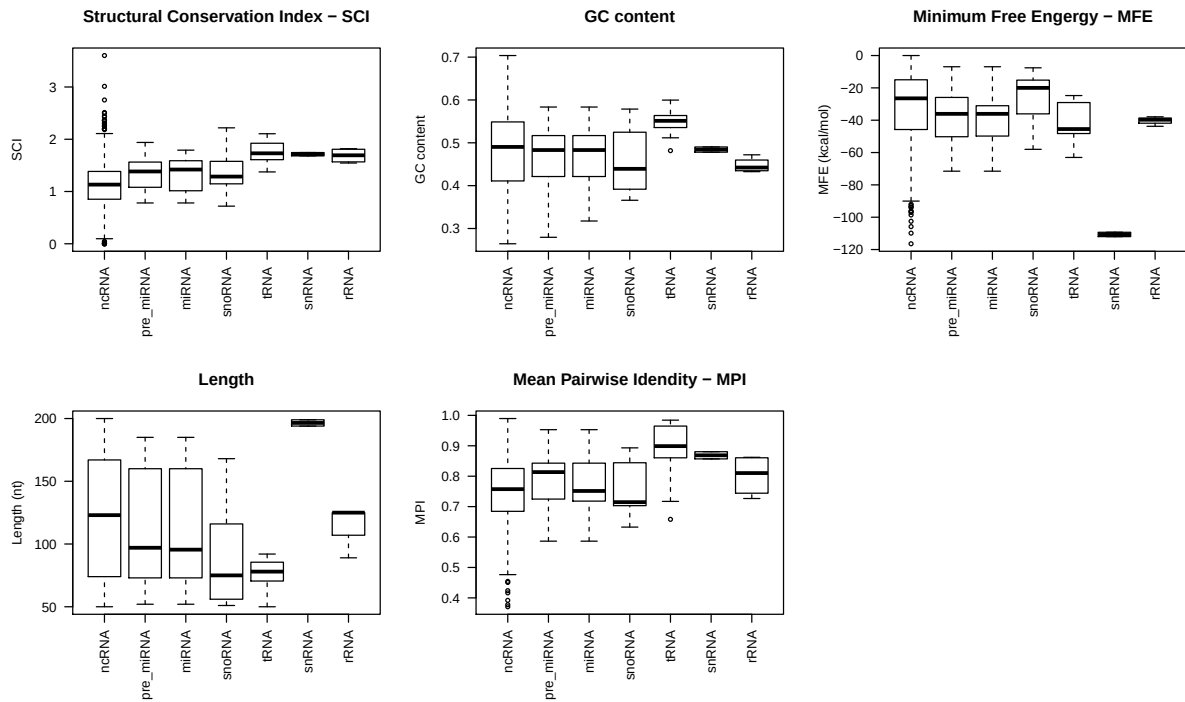


Figure 4.5: **Statistics of annotation types:** The boxplots show the statistics for the SCI, GC content, MFE, length and MPI of the ncRNA, pre_miRNA, miRNA, snoRNA, tRNA, snRNA and rRNA annotations found on the X chromosome of the Drosophila genome.

While all the median SCIs of all RNA families were above 1, it appeared the lowest for the broader ncRNA class where also outliers at SCIs around 0 can be observed as can be seen in figure 4.5. This observation could be explained by miRNAs, snoRNAs, snRNAs, rRNAs and tRNAs usually having a well defined structure that accounts for their function and thus a stronger conservation. The distributions of MFEs, GC content and length also vary greatly. Apparently the MPI identity of the alignments containing tRNAs is especially high, which explains why **RNAz** could predict them much better than **SISSIZ**. On the other hand, **SISSIZ** has advantages in predicting the broader class of non-coding RNAs and snoRNA which show the lowest median MPIs.

4.1.4 Conclusion

The screen on the *Drosophila* X chromosome revealed that **RNAz** and **SISSIZ** do not share many common predictions. These differences are also the reason why Smith et al. (2013) implemented a hybrid **SISSIZ-RNAz** pipeline that uses **RNAz** at MPIs above 85% and **SISSIZ** at MPIs below this threshold to ensure that both programs operate at their optimum range. Even though, realignment with **mlocarna** increases the runtime of the pipeline, it could be shown to be beneficial for alignments with low MPIs as the number of (annotated) predictions could be increased notably. At a FPR of 0.05 at best 30% of the annotations could be recovered by **SISSIZ** and 38% by **RNAz**. At the same time many new predictions on unannotated grounds were made, which have to be viewed with caution: Since the FDR is much higher than the FPR, the predictions of the pipeline should not be used to argue for novel ncRNAs but to highlight the differences in the prediction of ncRNAs between **SISSIZ** and **RNAz**.

4.2 HIV Screen

4.2.1 Introduction

In the final application example a screen is performed on a genomic HIV (Human Immunodeficiency Virus) alignment using additional probing data. The aim of this screen was not to discover novel structural features in the HIV RNA genome which already is well studied, but to analyze the benefits of probing data in the prediction of ncRNAs. Lavender et al. (2015a) used SHAPE data on an alignment of the HIV-1 NL4-3 (Human Immunodeficiency Virus 1), SIVcpz (Simian Immunodeficiency Virus of the chimpanzee), and SIVmac (Simian Immunodeficiency Virus of the macaque) to identify new structured regions in the HIV genome. The origins of HIV-1 could be traced back to the SIVcpz thus they are phylogenetically more related to each other than the SIVcpz, which on the other hand is viewed as the source of HIV-2 (Sharp and Hahn, 2010).

The structure of the HIV genome

The human immunodeficiency virus belongs to the family of Retroviridae and more specifically to the genus of Lentiviruses. Therefore, HIV is an enveloped positive-sense single stranded RNA viruses which is widely known for its cause of AIDS (Acquired Immuno Deficiency Syndrome). The envelope of the HIV contains two copies of its single stranded RNA genome which is reverse transcribed by the virally encoded reverse transcriptase and integrated into the host genome by the viral enzyme integrase. Seitz (2016). Since the viral genome is positive-sensed it can be directly translated to proteins. In many parts of the viral replication cycle the structural RNA features of the HIV genome play an important role and will be illustrated in the following. Despite global efforts to discover a cure for AIDS it has not been found until today, although the progression of the disease can be slowed down by combination antiretroviral therapy (cART) (Cihlar and Fordyce, 2016). The treatment comprises inhibitors of the viral reverse transcriptase, integrase and protease. However, also the conserved RNA structure in the HIV genome could be a potential target for antiviral drugs:

1. RRE (Rev Response Element): At the early stages of HIV replication the RNA, transcribed from the integrated HIV DNA, is spliced by the cellular machinery and exported from the nucleus. The Rev protein is one of the early products of the exported spliced HIV RNA. Rev enters the nucleus and binds to the RRE element of the unspliced viral RNA enabling its export to the cytosol. As a results, the unspliced RNA can be translated to produce proteins that are required in the later stage of HIV replication cycle (Fernandes et al.,

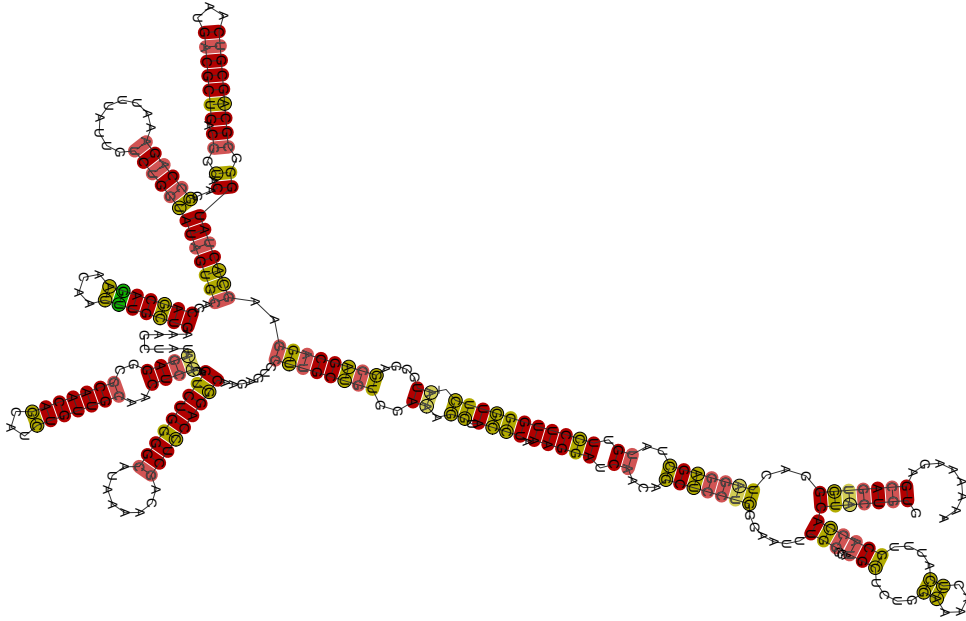


Figure 4.6: **RRE Secondary Structure:** The figure shows the consensus secondary structure of the RRE alignment calculate by RNAalifold with SHAPE data and ribosum scoring.

2012). The multiloop RNA structure of the RRE thus serves an important regulatory function and is highly conserved (Lusvarghi et al., 2013). The RRE secondary structure is visualized in figure 4.6.

2. LTR (Long Terminal Repeat): Another product of the early spliced HIV RNAs is the Tat (Trans-Activator of Transcription) protein which increases, as the name suggests, the transcription of viral DNA. Both, at the 5' and 3' LTR, a stem loop structure (TAR-trans-activating response element) is located and can be bound by Tat and leads to the recruitment of the positive transcription factor pTEFb (Karn and Stoltzfus (2012), Klaver and Berkhout (1994)).
3. Ribosomal frameshift: HIV uses overlapping open reading frames (ORFs) to maximize the capacity of its genome. Between the gag and pol gene one such an overlap exists. In order to shift the ORF by one nucleotide, a stem loop structure is required that stalls the ribosome at an upstream heptanucleotide slippery sequence (Biswas et al., 2004). Mouzakis et al. (2013) could show that the thermodynamic stability of the loop structure is vital for the successful replication of the virus by assuring a $\approx 5\%$ ratio between gag and gag-pol proteins. Therefore the ribosomal frameshift site appeared as a potential target for novel drugs. Ofori et al. (2014) discovered that compounds binding to the stem loop (FSS-frameshift stimulating signal) could increase frame shifting by up to 50 % and lower infectivity (figure 4.7).

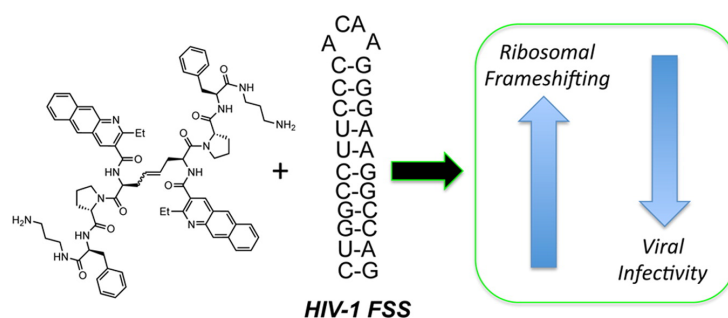


Figure 4.7: **HIV frameshift stimulating signal:** The ribosomal frameshift stimulating signal (FSS) could be shown to be a promising drug target. By increasing the ribosomal frameshift the ratio of gag and gag-pol proteins is disturbed lowering infectivity. Figure from Ofori et al. (2014) (ACS AuthorChoice license)

4.2.2 Methods

Both the alignment and probing data were available from their publications supplementary: The multiple alignment was created from pairwise SHAPE guided structural alignments (Lavender et al., 2015b) which were then merged to a single multiple sequence alignment using `tcocoffee` (Notredame et al., 2000). In the following the individual steps of the screen are outlined:

1. Download alignment and SHAPE data from the supplementary^a of Lavender et al. (2015a)
2. Slice the alignment file (*lavender_et_al_hiv_msa.aln*) and the corresponding SHAPE reactivity files (*hiv_aug2013.txt*, *sivcpz_1m7.txt* and *sivmac_1m7.txt*) into windows of length 200 and 100 overlap (self written script)
3. Shuffle all window files using `multiperm` (Anandam et al., 2009) to create a negative control dataset
4. Run the updated `SISSIZ` version with and without SHAPE data and `RNAz` (v-2.1) on all native and shuffled windows
5. Set thresholds for `SISSIZ` and `RNAz` to a fixed FDR of 0.01 using Benjamini-Hochberg p-value transformation (Benjamini and Hochberg, 1995)
6. Map annotation data from Siegfried et al. (2014) supplementary to the alignment

^a<http://journals.plos.org/ploscompbiol/article/file?type=supplementary&id=info:doi/10.1371/journal.pcbi.1004230.s002>

Although the alignment and SHAPE data for the screen was readily available, the dataset had to be prepared for the screen. As a first step, the initial alignment was divided into 200 nt long and 100 nt overlapping windows. In order to use the SHAPE data in the `SISSIZ` screen, it had to be mapped to each sequence in every alignment window. Therefore, a script was written in R that performs the alignment and SHAPE window creation simultaneously. The original alignment of 10026 nt length was divided into exactly 100 alignment windows and 300 respective SHAPE windows. For `SISSIZ` the following command line parameters were used: `--shape`, `--shape -j`, `-j` and `default`. `RNAz` was used with `default` parameters. In order, to make the results

comparable the threshold values were set to a fixed FDR of 0.01. Therefore, all multiperm shuffled windows were assumed not to contain structurally conserved RNAs and a Benjamini-Hochberg p-value transformation (Benjamini and Hochberg, 1995) was used. Following the adjustment, the z-score threshold values for **SISSiZ** were set to -4.92 in default mode and to -2.72 when the ribosum option was used. For **RNAz**'s RNA-class probability the threshold was set to 0.296649. The annotation data for verified functional RNA structures from Siegfried et al. (2014) refers to the HIV-1 NL4-3 sequence and had to be first mapped to the alignment.

4.2.3 Results and Discussion

The impact of SHAPE

The results of the HIV screen are visualized in figure 4.8 (**SISSiZ** with ribosum matrix and SHAPE) and figure 4.9 (**SISSiZ** with default parameters and SHAPE). It can be observed that the z-score is especially decreased at highly structured regions like the 5' and 3' LTR (Long Terminal Repeat), the RRE (Rev Response Element) and Ribosomal frameshift when SHAPE data is supplemented. In fact, these elements only become visible at a FDR of 0.01 if **SISSiZ** is run with SHAPE data. Only the RRE element could be recovered by **SISSiZ** without SHAPE data. Thus, SHAPE data seems to improve the sensitivity of **SISSiZ**. At the same time only predictions in regions with verified functional RNA structures are made, indicating a high specificity. Interestingly, the V4 element could be detected with **SISSiZ** in default SHAPE mode but not in the ribosum SHAPE mode. On the other hand **RNAz**, which does not support experimental probing data, manages to find the same regions without SHAPE data, while making one prediction on the *vif* coding region, for which no verified structural annotation is available.

Short structural features

Splice sites and RT PK (Reverse Transcriptase Pseudoknot), which are relatively short structural features (10-20 nt), could not be recovered by either program. Apparently, this is due to the inherent limitation of the prediction programs to screen genomic regions in windows. As a result, small elements may be averaged out by surrounding unstructured regions; conversely longer structural features may be divided so that they cannot be detected. According to Smith et al. (2013), a window size of 200 nt and 100 nt overlap yields the best results. On the other hand, the perl script `rnaWindow.pl` (Gruber et al., 2010) uses a window size of 120 nt and 0 overlap by default. However, the optimal window size is likely to differ greatly depending on the structure. Therefore, it might have been possible to detect small structural motives if the

window size was adapted accordingly. Nevertheless, this was not the aim of this investigation.

Conclusion

It could be shown that the number of recovered structural HIV features is greatly increased by the use of SHAPE data in **SISSIZ**. As a result, even more annotated regions could be found than by **RNAz**. Although, the question of the correct negative control for the SHAPE data still remains, the HIV screen shows a promising future for probing data supported **SISSIZ** screens.

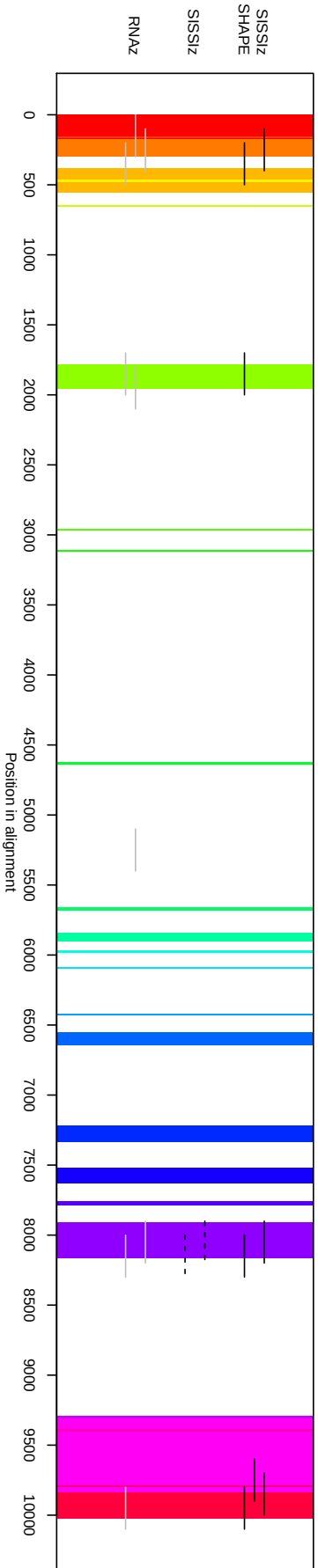
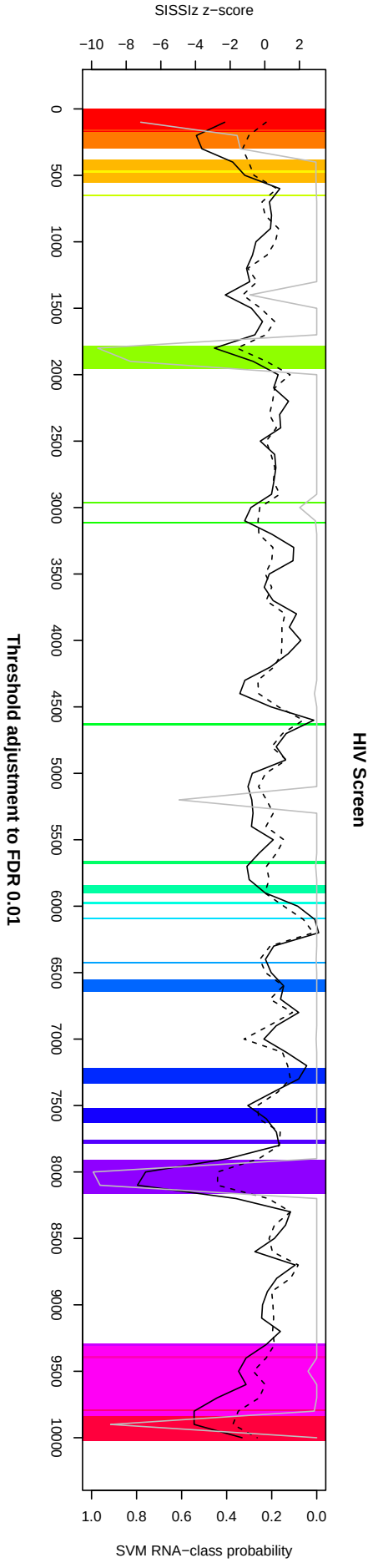


Figure 4.8: **HIV-screen 1: SISSiz with Ribosum matrix and SHAPE:** The figure shows the results of a screen on an HIV alignment, comprising 3 sequences: HIV-1NL4-3, SIVcpz, and SIVmac. SISSiz was run with and without SHAPE data using the ribosum matrix as an additional command line parameter while RNAz was run with default options. The initial alignment was split into 200 nt windows with an overlap of 100 nt. The upper plot shows the z-score and the RNA-class probability for SISSiz and RNAz respectively in dependence of the alignment position. (The alignment position of the i -th window was set to $i * \text{overlap}$.) The lower plot shows significant windows at a FDR of 0.01. The thresholds values for each program were determined from `multiPerm` shuffled alignments using the Benjamini-Hochberg p-value transformation. The legend beneath shows structural features of the HIV virus.

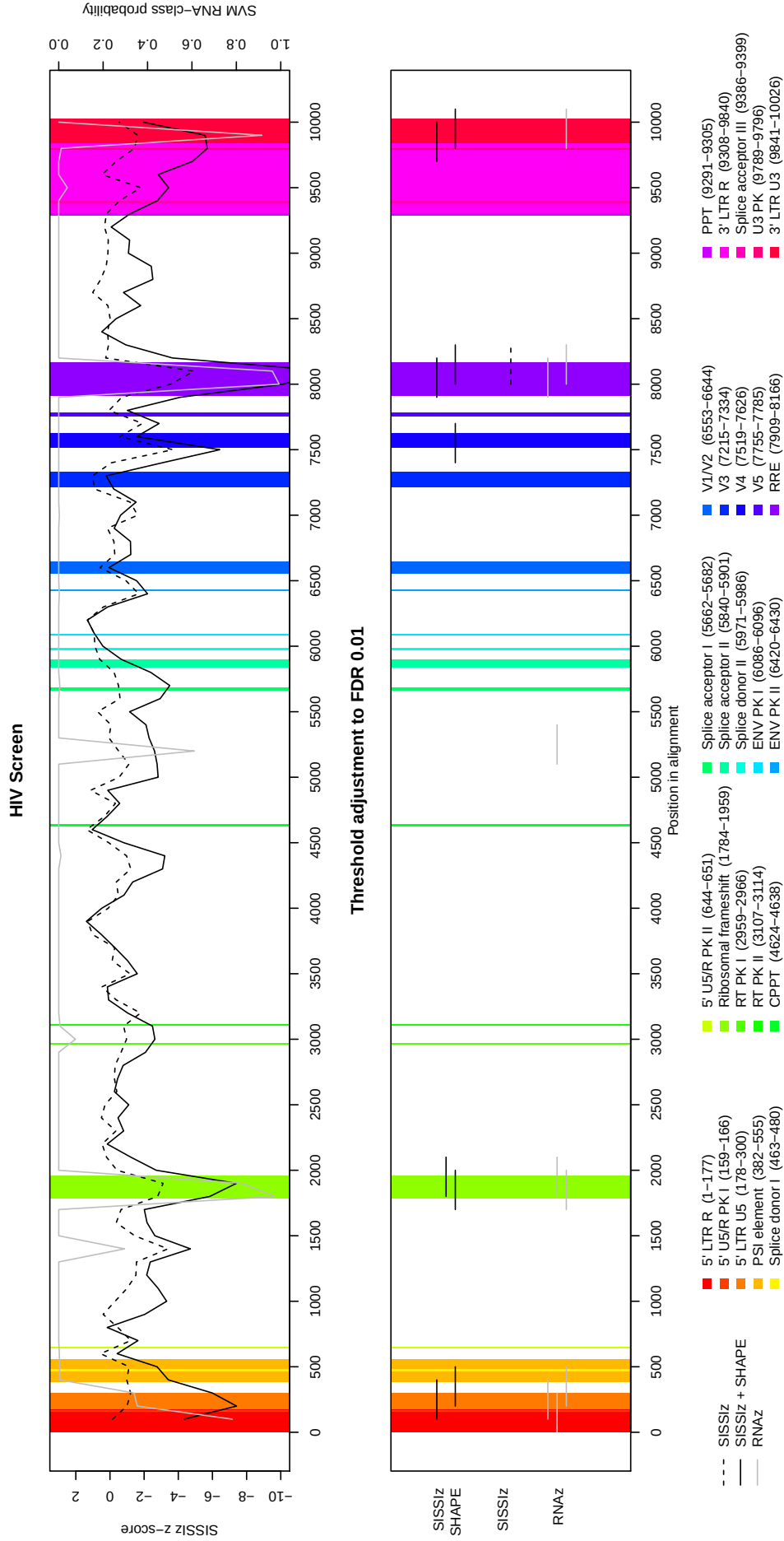


Figure 4.9: HIV-screen 2: SSIz with default parameters and SHAPE: The figure shows the results of a screen on an HIV alignment, comprising 3 sequences: HIV-1NL4-3, SIVcpz, and SIVmac. SSIz was run with and without SHAPE data using default command line parameter while RNAz was run with otherwise default options. The initial alignment was split into 200 nt windows with an overlap of 100 nt. The upper plot shows the z-score and the RNA-class probability for SSIz and RNAz respectively in dependence of the alignment position. (The alignment position of the i -th window was set to $i * \text{overlap}$.) The lower plot shows significant windows at a FDR of 0.01. The thresholds values for each program were determined from **multiper** shuffled alignments using the Benjamini-Hochberg p-value transformation. The legend beneath shows structural features of the HIV virus.

Chapter 5

Outlook

At the end of this thesis, it is time to draw a summarizing conclusion of the work and provide an outlook on the future of **SISSIZ**.

5.1 Simulation

Even though, the algorithm behind the simulation module of **SISSIZ** remained untouched, it could be indirectly improved. The new PRNG scores with an improved period length and a slightly better runtime. Also the advanced seeding options increased the versatility of **SISSIZ** along with added options to read seeds which allows for the reproduction of simulations. As a result, the new functions might further increase the appeal to use **SISSIZ** for simulating alignments in which **SISSIZ**'s neighborhood system takes a unique place. Future work could further explore the question when to use **SISSIZ** and when **multiPerm** for alignment randomization. The underlying **SISSI** framework Gesell and von Haeseler (2006) is not limited to neighboring dependences only so that more complex neighborhood systems could be realized. With respect to RNA gene finding modeling the tri nucleotide content by introducing a $|A|^5 \times |A|^5$ matrix could facilitate the search of structurally conserved RNAs in coding regions. This could be particularly relevant in screens of viral genomes which consist mostly of coding regions due to their genomic compactness. The influence of codon usage on the prediction of ncRNAs would have thus been especially interesting to see in the HIV screen. Still, other codon models have not been considered yet and could be evaluated. The importance of structurally conserved RNAs in various viruses was discussed by Hofacker et al. (2004). Last but not least, the option to provide **SISSIZ** with a tree according to which alignments are to be simulated would be an appealing option that would allow the simulation of alignments with specified sequence diversities. In this context it would also be possible to provide the species tree instead of using the (gene) tree calculated from the input alignment and to analyze the impact on the ncRNA prediction.

5.2 RNA gene finding

One of the main objectives of this master’s thesis was to include an updated version of Vienna RNA library (Lorenz et al., 2011) into **SISSIZ** to enable the incorporation of experimental probing data into the prediction of ncRNAs. The results from the benchmarks show a great potential for probing data in the assessment of structural conserved RNAs. The growing size of the probing database RMDB (RNA Mapping Database) (Cordero et al., 2012) indicates that also more SHAPE data will be available in the future. Nevertheless, even with RMDB in existence, there is a need to generate alignments from sequences with available probing data and to provide them in a similar format as in the RFAM database (Nawrocki et al., 2015). Most available probing data sets are only available for short stretches of RNA and in an unsuitable way for alignment based ncRNA prediction programs. However, for the screening of genomic regions longer probing datasets are required. With respect to ncRNA gene finding, probing data could also be substituted by data that computationally models SHAPE data independently. Thereby the requirement for experimental data could be overcome. One such approach, called the meta-structure (Konrat, 2009), was originally developed for proteins and is currently being adapted for RNAs with promising results. A prove of concept is currently under preparation by Gesell et al.. Moreover the SCI could be shown to improve the ROC curves of **SISSIZ** especially if many sequences are used. Therefore, it could be interesting to further refine the incorporation of the SCI into the predictions of **SISSIZ**. The fact that **SISSIZ** uses a phylogenetic approach and processes alignments differently compared to **RNAz** would still make **SISSIZ** unique. Moreover, the **RNAalifold** could be replaced by a non thermodynamic approach, for example a model solely relying on covariation. Lastly, the underlying test statistic could be changed too by replacing the z-score. Regarding the used model, **SISSIZ**’s framework offers great flexibility.

5.3 Application: Screening for non-coding RNAs

The screens on the Drosophila X-chromosome screen could confirm that **RNAz** performs better at higher MPI and **SISSIZ** at lower MPI. This knowledge is also used by the pipeline of Smith et al. (2013) which employs both programs depending on the MPI and GC content. Thus, the pipeline could be updated to include the new version of **SISSIZ**. Maybe the SCI value could serve as an additional decision value for the pipeline. At long last, the HIV screen demonstrated how SHAPE data could be incorporated into full genomic screens and furthermore highlights the potential of SHAPE data to aid the prediction of ncRNAs. In the future, predictions on larger scales will hopefully be possible, either with experimental or modeled probing data.

Chapter 6

Appendix

Program and command line parameters				updated SISSiz version Average z-score						SISSiz (v.2.0) Average z-score						RNAz (v.2.1) RNA class probability
Rfam ID	Family Name	#Sequences	Alignment Length	Avg Length	Avg MPI %	#Base pairs	default	--mono	--ribosom	--ribosom --mono	default	--mono	--ribosom	--ribosom --mono	default	
RF00002	5 Ss rRNA	61	207	152	67	25	-4.32	-4.39	-3.17	-3.22	-5.46	-5.36	-3.79	-3.88	0.00	
RF00003	U1	100	203	162	63	40	-10.59	-10.89	-7.34	-7.53	-10.42	-10.73	-7.57	-7.82	0.73	
RF00004	U2	208	278	191	58	45	-16.15	-17.37	-10.51	-11.21	-16.46	-16.82	-10.95	-11.40	1.00	
RF00006	Vault	73	163	101	59	19	-9.50	-9.66	-5.50	-5.70	-10.78	-11.05	-5.88	-6.07	0.98	
RF00007	U12	62	269	161	61	42	-13.09	-15.56	-8.95	-10.39	-12.65	-14.67	-9.03	-10.33	0.99	
RF00012	U3	87	316	209	58	63	-16.88	-16.05	-10.36	-9.87	-18.30	-17.36	-11.11	-10.69	0.98	
RF00013	6S	149	264	180	43	49	-79.71	-82.39	-17.40	-17.60	-117.02	-116.63	-19.17	-19.46	1.00	
RF00015	U4	170	261	144	57	31	-15.01	-17.05	-8.22	-9.15	-15.29	-17.89	-8.28	-9.52	0.63	
RF00017	Metazoa SRP	91	412	293	68	120	-15.06	-14.95	-10.15	-9.91	-16.65	-16.39	-11.43	-10.80	0.97	
RF00020	U5	180	192	117	52	30	-24.56	-27.09	-12.61	-13.76	-26.56	-29.10	-12.43	-13.61	0.92	
RF00027	let-7	63	95	80	68	31	-13.54	-17.70	-8.83	-10.09	-13.67	-18.30	-8.62	-10.27	1.00	
RF00029	Intron gpiI	92	186	89	53	19	-28.63	-28.82	-11.48	-11.34	-38.35	-40.69	-12.72	-13.06	1.00	
RF00031	SECIS 1	61	93	65	41	18	-112.52	-120.73	-14.09	-13.16	-267.25	-140.21	-15.15	-12.89	0.67	
RF00037	IRE 1	62	38	35	77	15	-7.02	-6.31	-5.20	-4.66	-5.16	-4.80	-4.29	-3.93	0.26	
RF00050	FMN	144	221	136	68	23	-7.28	-8.03	-5.39	-6.04	-7.70	-8.41	-5.96	-6.49	0.96	
RF00053	miR-7	57	104	88	65	28	-10.93	-10.25	-8.87	-8.02	-10.95	-9.88	-9.01	-7.96	1.00	
RF00059	TPP	109	211	107	53	28	-10.83	-12.51	-7.23	-7.88	-15.39	-17.47	-8.79	-9.43	0.66	
RF00066	U7	49	75	62	65	11	-10.94	-12.59	-5.92	-6.11	-12.77	-14.44	-6.13	-6.60	0.91	
RF00096	U8	55	189	134	66	24	-10.67	-10.66	-5.99	-5.87	-10.41	-10.77	-6.26	-6.11	0.80	
RF00127	144	78	140	96	63	22	-11.08	-11.47	-6.89	-6.86	-11.69	-12.31	-6.88	-6.93	0.99	
RF00162	SAM	433	231	111	63	27	-10.90	-12.93	-7.69	-8.66	-11.43	-13.07	-8.34	-9.15	0.66	
RF00167	Purine	133	113	101	55	22	-18.31	-22.06	-6.60	-6.70	-23.79	-28.51	-6.68	-6.89	0.99	
RF00168	Lysine	47	274	184	48	53	-46.27	-57.04	-14.81	-15.45	-47.64	-59.38	-14.29	-15.87	0.97	
RF00169	Bacteria small SRP	261	127	86	53	33	-15.65	-15.92	-7.82	-8.08	-21.67	-21.36	-9.90	-9.86	0.89	
RF00174	Cobalamin	430	441	203	50	42	-14.18	-14.04	-10.14	-10.03	-14.67	-14.40	-11.04	-10.85	0.04	
RF00379	ydaO-yuaA	106	335	144	56	26	-7.17	-9.23	-5.49	-7.02	-9.86	-12.56	-7.13	-8.48	0.05	
RF00380	ykoK	157	278	169	58	47	-20.54	-22.58	-10.43	-11.48	-22.47	-25.13	-11.88	-12.67	0.84	
RF00442	ykkC-ykkD	97	177	108	62	20	-8.60	-9.38	-6.07	-6.54	-9.10	-9.32	-6.58	-6.79	0.20	
RF00504	Glycine	44	215	101	50	22	-14.01	-14.76	-8.05	-7.98	-18.87	-18.70	-9.46	-9.09	0.67	
RF00517	serC	70	55	52	75	12	-3.55	-3.97	-2.15	-2.32	-3.74	-4.42	-2.34	-2.60	0.20	
RF00519	subB	87	100	77	55	13	-17.22	-27.36	-6.82	-8.02	-21.03	-31.97	-7.32	-8.61	1.00	
RF00525	Flavivirus DB	111	84	72	72	19	-6.07	-6.59	-4.35	-4.74	-6.21	-6.89	-4.49	-4.99	0.89	

Table 6.1: **RFAM families used in Benchmark-Part 1:** The table shows the Rfam ID, family name, the number of sequences per family and alignment length in the RFAM database (Navrocki et al., 2015), the average alignment length and MPI of the sampled alignments (calculated by SISSiz) and the number of base pairs as retrieved from the RFAM database. For each family the average z-score (updated SISSiz version and SISSiz v.2.0, Smith et al. (2013)) and the average RNA class probability (RNAz (Gruber et al., 2010) computed from 20 sampled alignments with 10 sequences are displayed. The command line parameters are given underneath in the second row. The RFAM families were selected from a dataset utilized by Rivas et al. (2017) where families with alignment lengths over 500 nt were discarded.

Program and command line parameters							updated SSIz version Average z-score				SSIz (v.2.0) Average z-score				RNAz (v.2.1) RNA class probability	
Rfam ID	Family Name	#Sequences	Alignment Length	Avg Length	Avg MPI %	#Base pairs	default	--mono	--ribosom	--ribosum --mono	default	--mono	--ribosom	--ribosum --mono	default	RNA class probability
RF00548	U11	72	350	153	58	42	-22.91	-24.78	-12.46	-13.65	-23.42	-25.40	-13.19	-14.06	0.79	
RF00618	U4atac	61	224	135	61	30	-13.42	-13.25	-7.50	-7.39	-13.06	-12.87	-7.55	-7.48	0.44	
RF00619	U6atac	62	156	121	69	14	-5.91	-6.01	-4.91	-4.71	-6.32	-6.21	-5.02	-4.93	0.31	
RF00685	mir-36	46	119	101	54	28	-18.41	-20.88	-10.59	-11.64	-18.85	-21.89	-10.74	-11.20	1.00	
RF00711	mir-449	56	116	91	57	28	-14.04	-16.94	-8.75	-9.74	-13.75	-17.85	-8.62	-9.63	1.00	
RF00782	MIR480	69	192	139	68	19	-0.46	-0.71	-0.15	-0.22	-0.69	-1.05	-0.34	-0.45	0.00	
RF01051	c-di-GMP-I	155	136	87	59	22	-14.28	-17.81	-7.54	-8.60	-16.44	-20.84	-8.33	-9.50	0.56	
RF01055	MOCO RNA motif	160	255	142	55	42	-15.99	-17.57	-9.85	-10.39	-16.02	-18.23	-9.96	-10.60	0.51	
RF01057	SAH riboswitch	52	172	85	59	13	-6.49	-6.34	-4.30	-4.18	-6.96	-6.60	-4.70	-4.50	0.34	
RF01068	mini-ykkC	72	55	48	55	13	-13.12	-11.52	-6.51	-6.00	-15.14	-13.63	-7.26	-6.63	1.00	
RF01118	PK-G12rRNA	347	124	108	74	18	-9.65	-9.49	-6.34	-6.00	-11.97	-11.38	-7.76	-7.29	1.00	
RF01394	isrK	52	92	78	60	18	-15.99	-15.77	-7.29	-7.69	-17.46	-17.18	-7.62	-7.90	0.99	
RF01685	6S-Flavo	82	148	107	70	17	-11.09	-13.13	-5.00	-5.31	-13.63	-13.78	-5.54	-5.99	0.04	
RF01687	Acido-Lenti-1	87	46	46	64	13	-12.81	-16.32	-6.88	-7.68	-15.10	-19.25	-7.48	-8.21	0.97	
RF01689	AdoCbl-variant	144	175	131	78	30	-4.88	-6.18	-3.59	-4.43	-5.26	-6.74	-3.96	-4.92	0.00	
RF01690	Bacillaceae-1	95	75	67	63	17	-10.66	-12.60	-5.96	-6.53	-11.77	-14.28	-6.08	-6.83	0.99	
RF01695	C4	456	129	87	56	27	-28.75	-27.77	-12.21	-12.58	-35.53	-32.72	-14.46	-14.34	1.00	
RF01699	Clostridiales-1	194	264	169	53	62	-58.05	-60.58	-21.71	-23.55	-56.82	-65.49	-21.58	-23.46	1.00	
RF01701	Cyano-1	265	90	68	61	14	-16.55	-17.93	-7.83	-8.24	-16.32	-18.93	-7.76	-8.34	1.00	
RF01704	Downstream-peptide	627	117	60	61	13	-12.72	-13.14	-6.70	-6.78	-15.13	-14.11	-6.63	-6.45	0.08	
RF01705	Flavo-1	201	184	82	56	19	-29.48	-47.82	-9.71	-11.83	-35.29	-67.33	-10.67	-13.62	0.34	
RF01717	PhotoRC-II	445	141	95	61	13	-15.36	-21.40	-5.47	-6.60	-16.29	-23.13	-5.44	-6.37	0.76	
RF01725	SAM-I-IV-variant	437	239	99	55	24	-19.47	-26.16	-10.32	-12.32	-23.25	-27.20	-10.98	-12.39	0.25	
RF01727	SAM-SAH	53	54	49	66	10	-6.30	-6.82	-3.63	-3.86	-6.47	-7.03	-3.68	-3.85	0.95	
RF01731	TwoAYGGAY	210	193	161	48	45	-60.11	-61.62	-16.30	-17.02	-80.04	-89.70	-18.34	-19.30	1.00	
RF01734	crcB	287	225	69	56	10	-10.22	-10.06	-5.10	-4.99	-12.70	-13.07	-5.71	-5.67	0.00	
RF01739	glnA	956	274	57	69	16	-5.82	-7.31	-4.28	-5.15	-8.59	-9.68	-5.79	-6.42	0.03	
RF01745	manA	189	342	190	73	52	-12.75	-13.57	-8.54	-8.69	-12.50	-12.45	-8.66	-8.67	0.15	
RF01747	msiK	120	68	59	67	13	-3.96	-4.13	-3.83	-3.95	-4.52	-4.56	-4.23	-4.26	0.96	
RF01749	pan	74	145	91	51	21	-40.28	-86.23	-13.85	-15.56	-48.54	-72.25	-13.88	-16.01	0.98	
RF01750	pfl	182	319	95	55	22	-9.43	-9.31	-5.87	-6.00	-10.73	-11.06	-6.53	-6.65	0.01	
RF01751	potC	136	85	70	80	17	-6.46	-10.49	-4.32	-6.27	-6.49	-10.58	-4.17	-6.07	1.00	

Table 6.2: **RFAM families used in Benchmark-Part 2:** The table shows the Rfam ID, family name, the number of sequences per family and alignment length in the RFAM database (Nawrocki et al., 2015), the average alignment length and MPI of the sampled alignments (calculated by SSIz) and the number of base pairs as retrieved from the RFAM database. For each family the average z-score (updated SSIz version and SSIz v.2.0, Smith et al. (2013)) and the average RNA class probability (RNAz (Gruber et al., 2010) computed from 20 sampled alignments with 10 sequences are displayed. The command line parameters are given underneath in the second row. The RFAM families were selected from a dataset utilized by Rivas et al. (2017) where families with alignment lengths over 500 nt were discarded.

Program and command line parameters					updated SISSiz version Average z-score					SISSiz (v.2.0) Average z-score					RNAz (v2.1) RNA class probability
Rfam ID	Family Name	#Sequences	Alignment Length	Avg Length	Avg MPI %	#Base pairs	default	--mono	--ribosome	--ribosome --mono	default	--mono	--ribosome	--ribosome --mono	default
RF01754	radC	130	193	146	58	35	-20.45	-17.59	-8.85	-8.41	-23.09	-19.91	-10.16	-9.31	1.00
RF01761	wcaG	118	136	98	67	18	-7.42	-9.22	-5.13	-5.95	-8.46	-9.60	-5.73	-6.23	0.13
RF01764	yidF	62	212	105	66	24	-10.26	-11.10	-6.24	-6.71	-10.48	-11.02	-6.37	-6.92	0.08
RF01786	c-di-GMP-II	54	113	85	55	22	-11.24	-13.62	-5.58	-6.40	-15.15	-18.50	-6.41	-7.30	0.14
RF01793	fth	89	62	52	64	11	-7.86	-7.87	-4.80	-4.89	-9.52	-8.86	-5.60	-5.32	0.99
RF01794	sok	91	298	162	56	27	-19.78	-22.64	-10.12	-11.04	-19.68	-22.56	-10.35	-10.97	0.99
RF01831	THF	97	179	100	59	30	-22.68	-23.53	-9.56	-10.13	-27.22	-27.91	-10.77	-11.27	0.84
RF01844	Smy	61	93	80	59	15	-8.71	-12.57	-5.58	-6.85	-9.45	-12.92	-5.67	-6.94	1.00
RF01852	tRNA-Sec	109	119	91	55	30	-18.28	-18.35	-11.57	-11.36	-19.87	-19.72	-12.60	-12.36	1.00
RF01857	Archaea SRP	53	363	306	52	132	-26.28	-28.75	-14.80	-15.92	-31.47	-34.68	-17.38	-18.66	1.00
RF01859	Phe leader	71	179	124	63	40	-24.71	-28.34	-12.49	-13.06	-25.40	-29.90	-12.73	-13.20	1.00
RF01998	group-II-D1D4-1	237	175	85	55	22	-11.01	-9.27	-6.78	-5.95	-12.82	-10.83	-7.26	-6.51	0.01
RF01999	group-II-D1D4-2	144	166	118	59	38	-20.08	-20.57	-10.45	-10.87	-23.54	-24.16	-12.18	-12.54	0.98
RF02001	group-II-D1D4-3	406	403	173	54	48	-20.74	-22.37	-9.89	-10.31	-24.26	-26.01	-11.39	-11.79	0.00
RF02003	group-II-D1D4-4	90	237	138	50	27	-34.55	-38.70	-11.92	-11.91	-50.31	-52.44	-13.81	-13.63	0.44
RF02004	group-II-D1D4-5	139	372	197	54	54	-23.94	-27.44	-11.01	-11.91	-27.89	-29.82	-12.05	-12.90	0.02
RF02005	group-II-D1D4-6	174	417	226	50	58	-39.90	-40.25	-15.34	-15.01	-45.15	-43.66	-17.65	-17.13	0.00
RF02012	group-II-D1D4-7	244	267	144	50	41	-33.71	-34.46	-11.76	-12.48	-41.40	-44.99	-13.73	-14.23	0.13
RF02033	HEARO	110	362	144	53	35	-24.00	-26.46	-10.88	-10.96	-25.53	-25.88	-10.91	-10.71	0.11
RF02034	IMES-1	219	241	191	70	35	-10.08	-12.22	-7.33	-7.69	-8.89	-10.54	-7.00	-7.50	0.42
RF02035	IMES-2	366	153	141	79	39	-7.96	-9.21	-5.78	-6.54	-8.63	-9.89	-6.33	-7.11	1.00
RF02221	sRNA-Xcc1	74	108	86	60	20	-15.67	-15.17	-7.86	-8.06	-21.44	-19.88	-9.72	-9.91	1.00
RF02278	Betaproteobacteria toxic sRNA	51	76	63	71	16	-11.48	-14.27	-6.20	-7.10	-11.49	-14.60	-5.98	-6.90	1.00
RF02379	csRNA	48	138	94	78	20	-11.21	-16.74	-5.56	-7.41	-11.13	-17.06	-5.44	-7.19	1.00

Table 6.3: **RFAM families used in Benchmark-Part 3:** The table shows the Rfam ID, family name, the number of sequences per family and alignment length in the RFAM database (Nawrocki et al., 2015), the average alignment length and MPI of the sampled alignments (calculated by SSIStiz) and the number of base pairs as retrieved from the RFAM database. For each family the average z-score (updated SSIStiz version and SSIStiz v.2.0, Smith et al. (2013)) and the average RNA class probability (RNAz (Gruber et al., 2010) computed from 20 sampled alignments with 10 sequences are displayed. The command line parameters are given underneath in the second row. The RFAM families were selected from a dataset utilized by Rivas et al. (2017) where families with alignment lengths over 500 nt were discarded.

Bibliography

- N. Agrawal, P. V. Dasaradhi, A. Mohammed, P. Malhotra, R. K. Bhatnagar, and S. K. Mukherjee. RNA interference: biology, mechanism, and applications. *Microbiol. Mol. Biol. Rev.*, 67(4): 657–685, Dec 2003. [PubMed Central:PMC309050] [PubMed:14665679].
- S. F. Altschul and B. W. Erickson. Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage. *Mol. Biol. Evol.*, 2(6):526–538, Nov 1985. [PubMed:3870875].
- P. Anandam, E. Torarinsson, and W. L. Ruzzo. Multiperm: shuffling multiple sequence alignments while approximately preserving dinucleotide frequencies. *Bioinformatics*, 25(5):668–669, Mar 2009. [DOI:10.1093/bioinformatics/btp006] [PubMed:19136551].
- T. Babak, B. J. Blencowe, and T. R. Hughes. Considerations in the identification of functional RNA structural elements in genomic alignments. *BMC Bioinformatics*, 8:33, Jan 2007. [PubMed Central:PMC1803800] [DOI:10.1186/1471-2105-8-33] [PubMed:17263882].
- D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–297, Jan 2004. [PubMed:14744438].
- Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, 57(1):289–300, 1995. ISSN 00359246. URL <http://www.jstor.org/stable/2346101>.
- S. H. Bernhart, I. L. Hofacker, S. Will, A. R. Gruber, and P. F. Stadler. RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, 9:474, Nov 2008. [PubMed Central:PMC2621365] [DOI:10.1186/1471-2105-9-474] [PubMed:19014431].
- P. Biswas, X. Jiang, A. L. Pacchia, J. P. Dougherty, and S. W. Peltz. The human immunodeficiency virus type 1 ribosomal frameshifting site is an invariant sequence determinant and an important target for antiviral therapy. *J. Virol.*, 78(4):2082–2087, Feb 2004. [PubMed Central:PMC369415] [PubMed:14747573].
- M. Blanchette, W. J. Kent, C. Riemer, L. Elnitski, A. F. Smit, K. M. Roskin, R. Baertsch, K. Rosenbloom, H. Clawson, E. D. Green, D. Haussler, and W. Miller. Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res.*, 14(4):708–715, Apr 2004. [PubMed Central:PMC383317] [DOI:10.1101/gr.1933104] [PubMed:15060014].
- R. K. Bradley, A. V. Uzilov, M. E. Skinner, Y. R. Bendana, L. Barquist, and I. Holmes. Evolutionary modeling and prediction of non-coding RNAs in *Drosophila*. *PLoS ONE*, 4(8):e6478, Aug 2009. [PubMed Central:PMC2721679] [DOI:10.1371/journal.pone.0006478] [PubMed:19668382].
- T. Cihlar and M. Fordyce. Current status and prospects of HIV treatment. *Curr Opin Virol*, 18:50–56, Jun 2016. [DOI:10.1016/j.coviro.2016.03.004] [PubMed:27023283].
- T. H. Click, A. Liu, and G. A. Kaminski. Quality of random number generators significantly affects results of Monte Carlo simulations for organic and biological systems. *J Comput*

- Chem*, 32(3):513–524, Feb 2011. [PubMed Central:PMC2992609] [DOI:10.1002/jcc.21638] [PubMed:20734313].
- P. Cordero, J. B. Lucks, and R. Das. An RNA Mapping DataBase for curating RNA structure mapping experiments. *Bioinformatics*, 28(22):3006–3008, Nov 2012. [PubMed Central:PMC3496344] [DOI:10.1093/bioinformatics/bts554] [PubMed:22976082].
- F. H. Crick. On protein synthesis. *Symp. Soc. Exp. Biol.*, 12:138–163, 1958. [PubMed:13580867].
- D. M. Crothers and B. H. Zimm. THEORY OF THE MELTING TRANSITION OF SYNTHETIC POLYNUCLEOTIDES: EVALUATION OF THE STACKING FREE ENERGY. *J. Mol. Biol.*, 9:1–9, Jul 1964. [PubMed:14200383].
- K. E. Deigan, T. W. Li, D. H. Mathews, and K. M. Weeks. Accurate SHAPE-directed RNA structure determination. *Proc. Natl. Acad. Sci. U.S.A.*, 106(1):97–102, Jan 2009. [PubMed Central:PMC2629221] [DOI:10.1073/pnas.0806929106] [PubMed:19109441].
- F. Deng, M. Ledda, S. Vaziri, and S. Aviran. Data-directed RNA secondary structure prediction using probabilistic modeling. *RNA*, 22(8):1109–1119, Aug 2016. [PubMed Central:PMC4931104] [DOI:10.1261/rna.055756.115] [PubMed:27251549].
- I. Dunham, A. Kundaje, S. F. Aldred, P. J. Collins, C. A. Davis, F. Doyle, C. B. Epstein, S. Fietze, and more. An integrated encyclopedia of DNA elements in the human genome. *Nature*, 489(7414):57–74, Sep 2012. [PubMed Central:PMC3439153] [DOI:10.1038/nature11247] [PubMed:22955616].
- S. R. Eddy. The C-value paradox, junk DNA and ENCODE. *Curr. Biol.*, 22(21):R898–899, Nov 2012. [DOI:10.1016/j.cub.2012.10.002] [PubMed:23137679].
- S. R. Eddy. Computational analysis of conserved RNA secondary structure in transcripts and genomes. *Annu Rev Biophys*, 43:433–456, 2014. [PubMed Central:PMC5541781] [DOI:10.1146/annurev-biophys-051013-022950] [PubMed:24895857].
- M. Esteller. Non-coding RNAs in human disease. *Nat. Rev. Genet.*, 12(12):861–874, Nov 2011. [DOI:10.1038/nrg3074] [PubMed:22094949].
- J. Felsenstein. Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol.*, 17(6):368–376, 1981. [PubMed:7288891].
- J. Felsenstein and G. A. Churchill. A Hidden Markov Model approach to variation among sites in rate of evolution. *Mol. Biol. Evol.*, 13(1):93–104, Jan 1996. [DOI:10.1093/oxfordjournals.molbev.a025575] [PubMed:8583911].
- J. Fernandes, B. Jayaraman, and A. Frankel. The HIV-1 Rev response element: an RNA scaffold that directs the cooperative assembly of a homo-oligomeric ribonucleoprotein complex. *RNA Biol*, 9(1):6–11, Jan 2012. [PubMed Central:PMC3342944] [DOI:10.4161/rna.9.1.18178] [PubMed:22258145].
- Ronald Aylmer Fisher and Frank Yates. Statistical tables for biological, agricultural and medical research. 1948.
- P. P. Gardner, A. Wilm, and S. Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Res.*, 33(8):2433–2439, 2005. [PubMed Central:PMC1087786] [DOI:10.1093/nar/gki541] [PubMed:15860779].
- O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol. Biol. Evol.*, 14(7):685–695, Jul 1997. [PubMed:9254330].

- T. Gesell and A. von Haeseler. In silico sequence evolution with site-specific interactions along phylogenetic trees. *Bioinformatics*, 22(6):716–722, Mar 2006. [DOI:10.1093/bioinformatics/bti812] [PubMed:16332711].
- T. Gesell and S. Washietl. Dinucleotide controlled null models for comparative RNA gene prediction. *BMC Bioinformatics*, 9:248, May 2008. [PubMed Central:PMC2453142] [DOI:10.1186/1471-2105-9-248] [PubMed:18505553].
- Tanja Gesell. A Phylogenetic Definition of Structure. 2009. University of Vienna.
- Tanja Gesell, Jana Terenova, and Perschy Lorenz. Rna meta-structure: A proof of concept on viruses.
- W Gilbert. Origin of life: The rna world. *Nature*, page 319: 618, 1986.
- L. Sian Gramates, Steven J. Marygold, Gilberto dos Santos, Jose-Maria Urbano, Giulia Antonazzo, Beverley B. Matthews, Alix J. Rey, Christopher J. Tabone, Madeline A. Crosby, David B. Emmert, Kathleen Falls, Joshua L. Goodman, Yanhui Hu, Laura Ponting, Andrew J. Schroeder, Victor B. Strelets, Jim Thurmond, Pinglei Zhou, and . Flybase at 25: looking to the future. *Nucleic Acids Research*, 45(D1):D663–D671, 2017. doi: 10.1093/nar/gkw1016. URL +<http://dx.doi.org/10.1093/nar/gkw1016>.
- N. C. Grassly, J. Adachi, and A. Rambaut. PSeq-Gen: an application for the Monte Carlo simulation of protein sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, 13(5):559–560, Oct 1997. [PubMed:9367131].
- D. Graur, Y. Zheng, N. Price, R. B. Azevedo, R. A. Zufall, and E. Elhaik. On the immortality of television sets: "function" in the human genome according to the evolution-free gospel of ENCODE. *Genome Biol Evol*, 5(3):578–590, 2013. [PubMed Central:PMC3622293] [DOI:10.1093/gbe/evt028] [PubMed:23431001].
- A. R. Gruber, S. Findeiss, S. Washietl, I. L. Hofacker, and P. F. Stadler. RNAz 2.0: improved noncoding RNA detection. *Pac Symp Biocomput*, pages 69–79, 2010. [PubMed:19908359].
- C. E. Hajdin, S. Bellaousov, W. Huggins, C. W. Leonard, D. H. Mathews, and K. M. Weeks. Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. *Proc. Natl. Acad. Sci. U.S.A.*, 110(14):5498–5503, Apr 2013. [PubMed Central:PMC3619282] [DOI:10.1073/pnas.1219988110] [PubMed:23503844].
- M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J. Mol. Evol.*, 22(2):160–174, 1985. [PubMed:3934395].
- R. Hershberg and D. A. Petrov. Selection on codon bias. *Annu. Rev. Genet.*, 42:287–299, 2008. [DOI:10.1146/annurev.genet.42.110807.091442] [PubMed:18983258].
- I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, 319(5):1059–1066, Jun 2002. [DOI:10.1016/S0022-2836(02)00308-X] [PubMed:12079347].
- I. L. Hofacker, P. F. Stadler, and R. R. Stocsits. Conserved RNA secondary structures in viral genomes: a survey. *Bioinformatics*, 20(10):1495–1499, Jul 2004. [DOI:10.1093/bioinformatics/bth108] [PubMed:15231541].
- M. Jiang, J. Anderson, J. Gillespie, and M. Mayne. uShuffle: a useful tool for shuffling biological sequences while preserving the k-let counts. *BMC Bioinformatics*, 9:192, Apr 2008. [PubMed Central:PMC2375906] [DOI:10.1186/1471-2105-9-192] [PubMed:18405375].

- David Jones. Good Practice in (Pseudo) Random Number Generation for Bioinformatics. *UCL Bioinformatics Group*, 2010.
- T. H. Jukes and C. R. Cantor. Evolution of protein molecules, in *Mammalian Protein Metabolism*. (New York, NY: Academic Press;), 21–132., 1969.
- D. Kandel, Y. Matias, R. Unger, and P. Winkler. Shuffling biological sequences. *Discrete Applied Mathematics*, 71(1):171 – 185, 1996. ISSN 0166-218X. doi: [https://doi.org/10.1016/S0166-218X\(97\)81456-4](https://doi.org/10.1016/S0166-218X(97)81456-4). URL <http://www.sciencedirect.com/science/article/pii/S0166218X97814564>.
- J. Karn and C. M. Stoltzfus. Transcriptional and posttranscriptional regulation of HIV-1 gene expression. *Cold Spring Harb Perspect Med*, 2(2):a006916, Feb 2012. [PubMed Central:PMC3281586] [DOI:10.1101/cshperspect.a006916] [PubMed:22355797].
- K. Katoh and D. M. Standley. MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, 30(4):772–780, Apr 2013. [PubMed Central:PMC3603318] [DOI:10.1093/molbev/mst010] [PubMed:23329690].
- M. Kimura. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, 16(2):111–120, Dec 1980. [PubMed:7463489].
- B. Klaver and B. Berkhout. Comparison of 5’ and 3’ long terminal repeat promoter function in human immunodeficiency virus. *J. Virol.*, 68(6):3830–3840, Jun 1994. [PubMed Central:PMC236888] [PubMed:8189520].
- Donald E. Knuth. The art of computer programming: Volume 2. *Reading, Mass: Addison-Wesley. Print.*, 1969.
- R. Konrat. The protein meta-structure: a novel concept for chemical and molecular biology. *Cell. Mol. Life Sci.*, 66(22):3625–3639, Nov 2009. [DOI:10.1007/s00018-009-0117-0] [PubMed:19690801].
- Pierre L’ Ecuyer and Richard Simard. TestU01: A C Library for Empirical Testing of Random Number Generators. *ACM Transactions on Mathematical Software*, 2007.
- M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, Nov 2007. [DOI:10.1093/bioinformatics/btm404] [PubMed:17846036].
- C. A. Lavender, R. J. Gorelick, and K. M. Weeks. Structure-Based Alignment and Consensus Secondary Structures for Three HIV-Related RNA Genomes. *PLoS Comput. Biol.*, 11(5):e1004230, May 2015a. [PubMed Central:PMC4439019] [DOI:10.1371/journal.pcbi.1004230] [PubMed:25992893].
- C. A. Lavender, R. Lorenz, G. Zhang, R. Tamayo, I. L. Hofacker, and K. M. Weeks. Model-Free RNA Sequence and Structure Alignment Informed by SHAPE Probing Reveals a Conserved Alternate Secondary Structure for 16S rRNA. *PLoS Comput. Biol.*, 11(5):e1004126, May 2015b.
- R. Lorenz, S. H. Bernhart, C. Honer Zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. ViennaRNA Package 2.0. *Algorithms Mol Biol*, 6:26, Nov 2011. [PubMed Central:PMC3319429] [DOI:10.1186/1748-7188-6-26] [PubMed:22115189].

- R. Lorenz, D. Luntzer, I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. SHAPE directed RNA folding. *Bioinformatics*, 32(1):145–147, Jan 2016. [PubMed Central:PMC4681990] [DOI:10.1093/bioinformatics/btv523] [PubMed:26353838].
- J. B. Lucks, S. A. Mortimer, C. Trapnell, S. Luo, S. Aviran, G. P. Schroth, L. Pachter, J. A. Doudna, and A. P. Arkin. Multiplexed RNA structure characterization with selective 2'-hydroxyl acylation analyzed by primer extension sequencing (SHAPE-Seq). *Proc. Natl. Acad. Sci. U.S.A.*, 108(27):11063–11068, Jul 2011. [PubMed Central:PMC3131332] [DOI:10.1073/pnas.1106501108] [PubMed:21642531].
- S. Lusvarghi, J. Sztuba-Solinska, K. J. Purzycka, G. T. Pauly, J. W. Rausch, and S. F. Grice. The HIV-2 Rev-response element: determining secondary structure and defining folding intermediates. *Nucleic Acids Res.*, 41(13):6637–6649, Jul 2013.
- D. Managadze, I. B. Rogozin, D. Chernikova, S. A. Shabalina, and E. V. Koonin. Negative correlation between expression level and evolutionary rate of long intergenic non-coding RNAs. *Genome Biol Evol.*, 3:1390–1404, 2011. [PubMed Central:PMC3242500] [DOI:10.1093/gbe/evr116] [PubMed:22071789].
- D. H. Mathews and D. H. Turner. Prediction of RNA secondary structure by free energy minimization. *Curr. Opin. Struct. Biol.*, 16(3):270–278, Jun 2006. [DOI:10.1016/j.sbi.2006.05.010] [PubMed:16713706].
- D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. U.S.A.*, 101(19):7287–7292, May 2004a. [PubMed Central:PMC409911] [DOI:10.1073/pnas.0401799101] [PubMed:15123812].
- D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. U.S.A.*, 101(19):7287–7292, May 2004b. [PubMed Central:PMC409911] [DOI:10.1073/pnas.0401799101] [PubMed:15123812].
- John S. Mattick and Marcel E. Dinger. The extent of functionality in the human genome. *The HUGO Journal*, 7(1):2, Jul 2013. ISSN 1877-6566. doi: 10.1186/1877-6566-7-2. URL <https://doi.org/10.1186/1877-6566-7-2>.
- J. L. McGinnis, J. A. Dunkle, J. H. Cate, and K. M. Weeks. The mechanisms of RNA SHAPE chemistry. *J. Am. Chem. Soc.*, 134(15):6617–6624, Apr 2012. [PubMed Central:PMC4337229] [DOI:10.1021/ja2104075] [PubMed:22475022].
- S. A. Mortimer and K. M. Weeks. A fast-acting reagent for accurate analysis of RNA secondary and tertiary structure by SHAPE chemistry. *J. Am. Chem. Soc.*, 129(14):4144–4145, Apr 2007. [DOI:10.1021/ja0704028] [PubMed:17367143].
- K. D. Mouzakis, A. L. Lang, K. A. Vander Meulen, P. D. Easterday, and S. E. Butcher. HIV-1 frameshift efficiency is primarily determined by the stability of base pairs positioned at the mRNA entrance channel of the ribosome. *Nucleic Acids Res.*, 41(3):1901–1913, Feb 2013. [PubMed Central:PMC3561942] [DOI:10.1093/nar/gks1254] [PubMed:23248007].
- E. P. Nawrocki, S. W. Burge, A. Bateman, J. Daub, R. Y. Eberhardt, S. R. Eddy, E. W. Floden, P. P. Gardner, T. A. Jones, J. Tate, and R. D. Finn. Rfam 12.0: updates to the RNA families database. *Nucleic Acids Res.*, 43(Database issue):D130–137, Jan 2015. [PubMed Central:PMC4383904] [DOI:10.1093/nar/gku1063] [PubMed:25392425].

- C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.*, 302(1):205–217, Sep 2000. [DOI:10.1006/jmbi.2000.4042] [PubMed:10964570].
- Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for Loop Matchings. *SIAM J. Appl. Math.*, July 1978. [DOI:10.1137/0135006].
- L. O. Ofori, T. A. Hilimire, R. P. Bennett, N. W. Brown, H. C. Smith, and B. L. Miller. High-affinity recognition of HIV-1 frameshift-stimulating RNA alters frameshifting in vitro and interferes with HIV-1 infectivity. *J. Med. Chem.*, 57(3):723–732, Feb 2014. [PubMed Central:PMC3954503] [DOI:10.1021/jm401438g] [PubMed:24387306].
- A. F. Palazzo and E. S. Lee. Non-coding RNA: what is functional and what is junk? *Front Genet*, 6:2, 2015. [PubMed Central:PMC4306305] [DOI:10.3389/fgene.2015.00002] [PubMed:25674102].
- J. S. Pedersen, G. Bejerano, A. Siepel, K. Rosenbloom, K. Lindblad-Toh, E. S. Lander, J. Kent, W. Miller, and D. Haussler. Identification and classification of conserved RNA secondary structures in the human genome. *PLoS Comput. Biol.*, 2(4):e33, Apr 2006. [PubMed Central:PMC1440920] [DOI:10.1371/journal.pcbi.0020033] [PubMed:16628248].
- A. R. Quinlan and I. M. Hall. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, Mar 2010. [PubMed Central:PMC2832824] [DOI:10.1093/bioinformatics/btq033] [PubMed:20110278].
- A. Rambaut and N. C. Grassly. Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees. *Comput. Appl. Biosci.*, 13(3):235–238, Jun 1997. [PubMed:9183526].
- J. L. Rinn and H. Y. Chang. Genome regulation by long noncoding RNAs. *Annu. Rev. Biochem.*, 81:145–166, 2012. [PubMed Central:PMC3858397] [DOI:10.1146/annurev-biochem-051410-092902] [PubMed:22663078].
- E. Rivas, J. Clements, and S. R. Eddy. A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat. Methods*, 14(1):45–48, Jan 2017. [PubMed Central:PMC5554622] [DOI:10.1038/nmeth.4066] [PubMed:27819659].
- Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12:77, 2011.
- F. Rodriguez, J. L. Oliver, A. Marin, and J. R. Medina. The general stochastic model of nucleotide substitution. *J. Theor. Biol.*, 142(4):485–501, Feb 1990. [PubMed:2338834].
- D. Rose, J. Hackermuller, S. Washietl, K. Reiche, J. Hertel, S. Findeiss, P. F. Stadler, and S. J. Prohaska. Computational RNomics of drosophilids. *BMC Genomics*, 8:406, Nov 2007. [PubMed Central:PMC2216035] [DOI:10.1186/1471-2164-8-406] [PubMed:17996037].
- S. Sahoo, M. P. Switnicki, and J. S. Pedersen. ProbFold: a probabilistic method for integration of probing data in RNA secondary structure prediction. *Bioinformatics*, 32(17):2626–2635, Sep 2016. [DOI:10.1093/bioinformatics/btw175] [PubMed:27153612].
- David Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM Journal on Applied Mathematics*, 45(5):810–825, 1985. doi: 10.1137/0145048. URL <https://doi.org/10.1137/0145048>.

- Rainer Seitz. Human Immunodeficiency Virus (HIV). *Transfus Med Hemother*, 43(3):203–222, May 2016. [PubMed Central:PMC4924471] [DOI:10.1159/000445852] [PubMed:27403093].
- A. Selega, C. Sirocchi, I. Iosub, S. Granneman, and G. Sanguinetti. Robust statistical modeling improves sensitivity of high-throughput RNA structure probing experiments. *Nat. Methods*, 14(1):83–89, Jan 2017.
- P. M. Sharp and B. H. Hahn. The evolution of HIV-1 and the origin of AIDS. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 365(1552):2487–2494, Aug 2010. [PubMed Central:PMC2935100] [DOI:10.1098/rstb.2010.0031] [PubMed:20643738].
- N. A. Siegfried, S. Busan, G. M. Rice, J. A. Nelson, and K. M. Weeks. RNA motif discovery by SHAPE and mutational profiling (SHAPE-MaP). *Nat. Methods*, 11(9):959–965, Sep 2014. [PubMed Central:PMC4259394] [DOI:10.1038/nmeth.3029] [PubMed:25028896].
- F. Sievers and D. G. Higgins. Clustal Omega, accurate alignment of very large numbers of sequences. *Methods Mol. Biol.*, 1079:105–116, 2014.
- F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J. D. Thompson, and D. G. Higgins. Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol. Syst. Biol.*, 7:539, Oct 2011. [PubMed Central:PMC3261699] [DOI:10.1038/msb.2011.75] [PubMed:21988835].
- M. A. Smith, T. Gesell, P. F. Stadler, and J. S. Mattick. Widespread purifying selection on RNA structure in mammals. *Nucleic Acids Res.*, 41(17):8220–8236, Sep 2013. [PubMed Central:PMC3783177] [DOI:10.1093/nar/gkt596] [PubMed:23847102].
- M. J. Smola, G. M. Rice, S. Busan, N. A. Siegfried, and K. M. Weeks. Selective 2'-hydroxyl acylation analyzed by primer extension and mutational profiling (SHAPE-MaP) for direct, versatile and accurate RNA structure analysis. *Nat Protoc*, 10(11):1643–1669, Nov 2015. [PubMed Central:PMC4900152] [DOI:10.1038/nprot.2015.103] [PubMed:26426499].
- Z. Sukosd, B. Knudsen, J. Kjems, and C. N. Pedersen. PPfold 3.0: fast RNA secondary structure prediction using phylogeny and auxiliary data. *Bioinformatics*, 28(20):2691–2692, Oct 2012. [DOI:10.1093/bioinformatics/bts488] [PubMed:22877864].
- Z. Sukosd, M. S. Swenson, J. Kjems, and C. E. Heitsch. Evaluating the accuracy of SHAPE-directed RNA secondary structure predictions. *Nucleic Acids Res.*, 41(5):2807–2816, Mar 2013. [PubMed Central:PMC3597644] [DOI:10.1093/nar/gks1283] [PubMed:23325843].
- K. Tamura and M. Nei. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Mol. Biol. Evol.*, 10(3):512–526, May 1993.
- D. Tollervey and T. Kiss. Function and synthesis of small nucleolar RNAs. *Curr. Opin. Cell Biol.*, 9(3):337–342, Jun 1997. [PubMed:9159079].
- C. Tyner, G. P. Barber, J. Casper, H. Clawson, M. Diekhans, C. Eisenhart, C. M. Fischer, D. Gibson, J. N. Gonzalez, L. Guruvadoo, M. Haeussler, S. Heitner, A. S. Hinrichs, D. Karolchik, B. T. Lee, C. M. Lee, P. Nejad, B. J. Raney, K. R. Rosenbloom, M. L. Speir, C. Villarreal, J. Vivian, A. S. Zweig, D. Haussler, R. M. Kuhn, and W. J. Kent. The UCSC Genome Browser database: 2017 update. *Nucleic Acids Res.*, 45(D1):D626–D634, Jan 2017. [PubMed Central:PMC5210591] [DOI:10.1093/nar/gkw1134] [PubMed:27899642].
- A. Varadarajan, R. K. Bradley, and I. H. Holmes. Tools for simulating evolution of aligned genomic regions with integrated parameter estimation. *Genome Biol.*, 9(10):R147, Oct 2008. [PubMed Central:PMC2760874] [DOI:10.1186/gb-2008-9-10-r147] [PubMed:18840304].

- S. Washietl and I. L. Hofacker. Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. *J. Mol. Biol.*, 342(1):19–30, Sep 2004. [DOI:10.1016/j.jmb.2004.07.018] [PubMed:15313604].
- S. Washietl, I. L. Hofacker, and P. F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. U.S.A.*, 102(7):2454–2459, Feb 2005. [PubMed Central:PMC548974] [DOI:10.1073/pnas.0409169102] [PubMed:15665081].
- S. Washietl, I. L. Hofacker, P. F. Stadler, and M. Kellis. RNA folding with soft constraints: reconciliation of probing data and thermodynamic secondary structure prediction. *Nucleic Acids Res.*, 40(10):4261–4272, May 2012. [PubMed Central:PMC3378861] [DOI:10.1093/nar/gks009] [PubMed:22287623].
- S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, 3(4):e65, Apr 2007. [PubMed Central:PMC1851984] [DOI:10.1371/journal.pcbi.0030065] [PubMed:17432929].
- S. Will, M. Yu, and B. Berger. Structure-based whole-genome realignment reveals many novel noncoding RNAs. *Genome Res.*, 23(6):1018–1027, Jun 2013. [PubMed Central:PMC3668356] [DOI:10.1101/gr.137091.111] [PubMed:23296921].
- C. Workman and A. Krogh. No evidence that mRNAs have lower folding free energies than random sequences with the same dinucleotide distribution. *Nucleic Acids Res.*, 27(24):4816–4822, Dec 1999. [PubMed Central:PMC148783] [PubMed:10572183].
- Y. Wu, B. Shi, X. Ding, T. Liu, X. Hu, K. Y. Yip, Z. R. Yang, D. H. Mathews, and Z. J. Lu. Improved prediction of RNA secondary structure by integrating the free energy model with restraints derived from experimental probing data. *Nucleic Acids Res.*, 43(15):7247–7259, Sep 2015. [PubMed Central:PMC4551937] [DOI:10.1093/nar/gkv706] [PubMed:26170232].
- J. Wuyts, G. Perriere, and Y. Van De Peer. The European ribosomal RNA database. *Nucleic Acids Res.*, 32(Database issue):D101–103, Jan 2004. [PubMed Central:PMC308799] [DOI:10.1093/nar/gkh065] [PubMed:14681368].
- Z. Yang. Estimating the pattern of nucleotide substitution. *J. Mol. Evol.*, 39(1):105–111, Jul 1994. [PubMed:8064867].
- Z. Yao, Z. Weinberg, and W. L. Ruzzo. CMfinder—a covariance model based RNA motif finding algorithm. *Bioinformatics*, 22(4):445–452, Feb 2006. [DOI:10.1093/bioinformatics/btk008] [PubMed:16357030].
- K. Zarrinhalam, M. M. Meyer, I. Dotu, J. H. Chuang, and P. Clote. Integrating chemical footprinting data into RNA secondary structure prediction. *PLoS ONE*, 7(10):e45160, 2012. [PubMed Central:PMC3473038] [DOI:10.1371/journal.pone.0045160] [PubMed:23091593].
- Michael Zuker and David Sankoff. Rna secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4), Jul 1984. ISSN 1522-9602. doi: 10.1007/BF02459506. URL <https://doi.org/10.1007/BF02459506>.