

Stefan Redl

July 27, 2025

## Contents

### 1 Abstract

### 2 Introduction

#### 2.1 Background

#### 2.2 Problem Statement

### 3 Methods

#### 3.1 SISSI

#### 3.2 SISSIZ

#### 3.3 RNAz

#### 3.4 ECSFinder

#### 3.5 Sship

### 4 Results

### 5 Discussion

### 6 SISSIZ Prediction

Checks whether the pipeline is still running anyway and whether the output is correct, i.e. no corrupt or incorrect files have been created. Unfortunately, several files with 0 bytes were created because I mistakenly worked on the wrong script and changed some code. I deleted all 0 bytes and restarted the pipeline.

### 7 RNAz Prediction

The pipeline is finished and has no false files.

## 8 Samples from SSIz, Multiperm and aln-Shuffle

Checked the data from the samples and they are fine. I made a script to convert the CLUSTAL files into FASTA files because the method SPOT-RNA needs a FASTA file to predict the RNA structure. I installed Biopython on my PC and also on the server.

Biopython version: 1.84

```
1 pip install biopython
```

Tested the script (convertCluToFasta.py) on my PC and it works fine.

```
1 python3 convertCluToFasta.py
```

The script is attached.

## 9 SPOT-RNA2

Tried to install SPOT-RNA2 as described at: <https://github.com/jaswindersingh2/SPOT-RNA2>

I was able to install SPOT-RNA2 on my PC, but was unable to run the program due to the requirement of the entire NCBI database ( 400GB), which caused issues as there is insufficient memory available on the server (only 15–30GB in my virtual environment). Therefore, this method will probably not work.

To create a link between the database and SPOT-RNA2 and simulate that the database is on the system.

## 10 SPOT-RNA

Since SPOT-RNA2 could not be used, I tried to install the previous version SPOT-RNA as described here: <https://github.com/jaswindersingh2/SPOT-RNA>

I was able to install it on my PC and it works perfectly.

## 11 RNA-MSM

Repository: <https://github.com/yikunpku/RNA-MSM?tab=readme-ov-file>

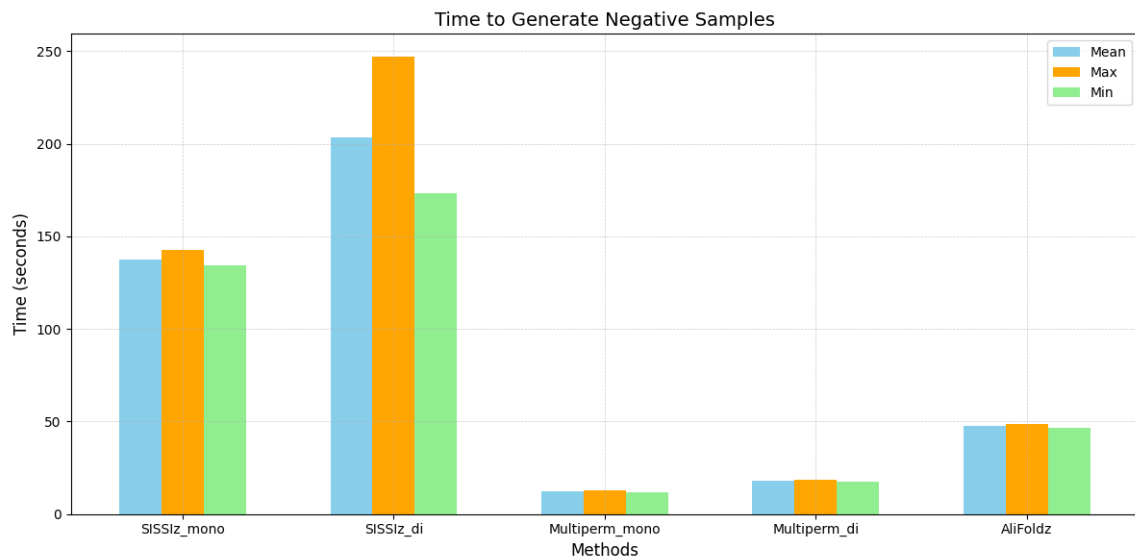
I'm just about to install it and try it out soon.

## 12 Questions

- Can I compare SPOT-RNA with SSIz? The output of a sample is attached.
- Should I use SPOT-RNA instead of SPOT-RNA2 or should I look for an alternative?

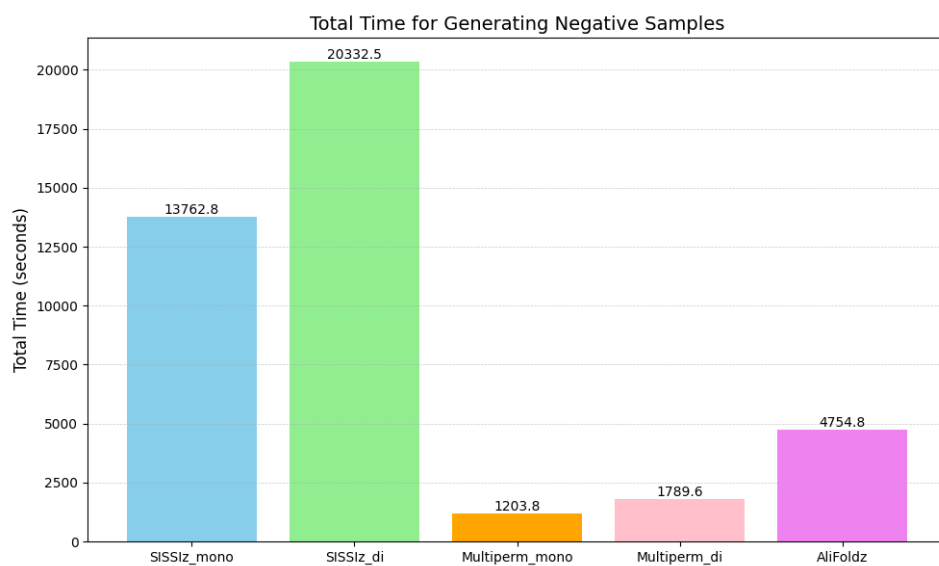
### 13 Benchmark of the Tools

This section presents the results of the randomization of the individual tools and compares their performance.



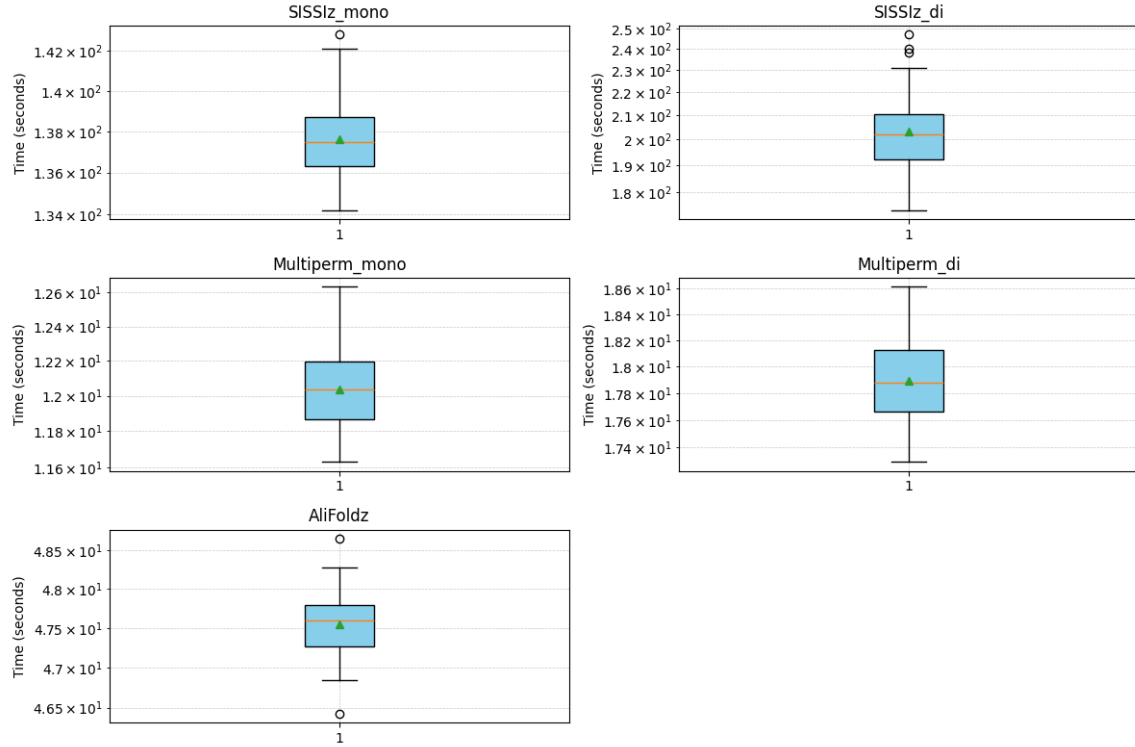
**Figure 1:** Display of the maximum, average and minimum time per 1000 samples

The SISSiz method requires significantly more time to simulate the randomized alignments. SISSiz mono requires an average of around 140 seconds per 1000 files, while SISSiz di is significantly slower at over 200 seconds. In contrast, the shuffle methods are 5 to 10 times faster, with Multiperm mono being the fastest method at only around 15 seconds per 1000 files.



**Figure 2:** Gesamtzeiten für jedes Werkzeug

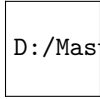
In terms of the total time for the simulation and shuffling of 100,000 alignments, SISSlZ requires the most time. The di variant takes 20,332.5 seconds, while the mono variant is slightly faster at 13,762.8 seconds. The shuffle methods, on the other hand, require less than 5,000 seconds.



**Figure 3:** Boxplot of the runtimes of the methods

## 14 RNAz Data analysis

Started the pipeline again to time how long RNAz takes to predict and then compare it to SISSlz and the other AI tools.



D:/Masterarbeit/2.Versuch/Result/RNAz/RNAz\_Boxplot\_pvalue.png

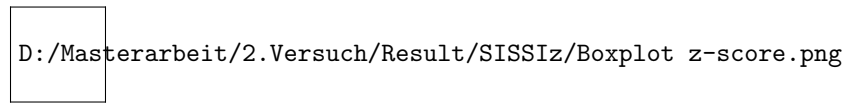
**Figure 4:** SVM RNA-class probability

The SVM RNA class probability does not clearly indicate that the RNA structure has been destroyed, as all measurable values are well above the threshold value of 0.9. The lowest value is only 0.987, which indicates an intact RNA structure.

Additionally, I wanted to create a plot with the z-score, but the mean z-score provided by RNAz only describes the mean value of the mean single sequence MFE, which cannot be compared to the z-score provided by SISSlz

## 15 SISSIZ Data analysis

This section presents the results of the SISSIZ data analysis.



**Figure 5:** Z-Scores

The z-scores confirm the previous results and show that SISSIZ has almost completely destroyed the RNA structure. The threshold values are between -4 and +4, where by everything below -4 is considered an almost intact RNA structure. This is also evident from the positive alignments of SISSI.

## 16 Tasks for KW5

- Calculation of the z-score of the RNA analysis in order to compare it with SSISSz
- Evaluation of the performance of RNAz and SSISSz
- Improve and adapt the pipeline diagram.
- Add the native alignments to the pipeline diagram.
- Consider which AI tools I use for the evaluation of the alignments

Summary KW5

## 17 Pipeline

In the first run, RNA sequences from *Bacillus subtilis* with 78 sequences and 401 alignments were used and subsequently randomized with SSISS. Subsequently, for each positive sample generated, 1,000 negative samples were created using shuffle and simulation methods in order to destroy the RNA secondary structure as far as possible. This was done using the following methods:

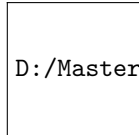
- MultiPerm v0.94 (mono- and dinucleotide conservation)
- aln-shuffle (mononucleotide conservation only)
- SSISSz v3.0 (mono- and dinucleotide conservation)

After the RNA secondary structure has been destroyed, it is checked whether it has been completely eliminated. I used the RNAz and SSISSz tools for this. The prediction is later extended with AI tools.

- RNAz v2.1.1
- SSISSz v3.0
- SPOT-RNA2 [\[1\]](#) [\[2\]](#)
- DeepFoldRNA [\[3\]](#) [\[4\]](#)
- RNA-MSM [\[5\]](#) [\[6\]](#)

The results were then visualized with Python, evaluated and compared with other methods.

After completion of the first run, an RNA sequence of *Bacillus subtilis* is extracted from the RFAM database. In this run, randomization with SSISS is omitted. Instead, the RNA secondary structure is destroyed directly and then analyzed with RNAz, SSISSz and the AI models.



D:/Masterarbeit/2.Versuch/Data/Pipeline Diagramm.png

**Figure 6:** Pipeline

## 18 Time capture from RNAz and SISSIz

I have integrated a time measurement into the scripts RNAz-prediction.py and SISSIz-prediction.py. An intermediate measurement is carried out after every 1,000 predictions. This enables a more precise evaluation of the performance of the prediction tools used. The test is carried out on a system with 24 cores to ensure a fair and comparable performance analysis.

## 19 Research about the p-value and mean-z-score in RNAz

I have noticed that the mean z-score is calculated differently in RNAz than in SISSIz. Therefore, I still need to check how I can adjust the calculation in RNAz to ensure direct comparability.

One possible solution would be to use RNAalifold. This could calculate the partial sequences from the Clustal files and then determine the mean value and the standard deviation.

## 20 Copy Data all Data to my Sytem

Because the FH get a new Server on Monday the 3th, i started to copy all necessary Folders into my System.

## 21 Tasks for KW6

- Extract an RNA sequence from the RFAM database to start the second run without SISSI
- Select two suitable AI tools and understand their process
- start a run from SISSI-prediction.py with the integrate time capturing
- as soon as the new fh server arrives, I will have to reinstall all the tools and recreate my setup



## 22 References SPOT-RNA2

- [1] <https://doi.org/10.1093/bioinformatics/btab165>
- [2] <https://github.com/jaswindersingh2/SPOT-RNA2>

## 23 References DeepFoldRNA

- [3] <https://doi.org/10.1101/2022.05.15.491755>
- [4] <https://github.com/robpearc/DeepFoldRNA>

## 24 References RNA-MSM

- [5] <https://doi.org/10.1093/nar/gkad1031>
- [6] <https://github.com/yikunpku/RNA-MSM>

Summary KW6 and KW7

## 25 Pipeline

## 26 Evaluate the timing of RNAz

Started to evaluate the timing from RNAz during the prediction. To compare it with SISSIz i started also a timing, but i must stop it, because the FH get a new Server. To be continued.

## 27 Searching for b.subtilis

I started to research for nativ b.subtilis alignments in the RFAM Database, but i cannot find the right alignments. After the Meeting with Tanja i restarted my research and try to translate the B.subtilis.ct2 file from SISSI into a fasta file. I entered the Fasta Sequence into the RFAM database and got 4 identical hits. To be continued

## 28 SPOT-RNA2[1]

### Title:

Improved RNA Secondary Structure and Tertiary Base Pairing Prediction Using Evolutionary Profiles, Mutational Coupling, and 2D Transfer Learning

### Authors:

Jaswinder Singh, Kuldeep Paliwal, Tongchuan Zhang, Jaspreet Singh, Thomas Litfin, Yaoqi Zhou

### Introduction:

The discovery of numerous non-coding RNAs, especially long non-coding RNAs, has revolutionized our understanding of their biological functions. The ability to accurately determine their secondary and tertiary structures is limited by the constraints of experimental techniques. Computational prediction methods have significantly improved in recent years with deep learning and transfer learning.

### Motivation:

Accurate RNA structure prediction is crucial for understanding its function. Traditional experimental methods such as X-ray crystallography and NMR are expensive and time-consuming, necessitating efficient computational methods.

### Methods:

The proposed method, SPOT-RNA2, extends previous approaches by incorporating evolutionary profiles and mutational coupling. Input data includes:

- One-hot encoding of the RNA sequence.
- Predicted base-pair probabilities from the single-sequence method LinearPartition.
- Position-Specific Scoring Matrix (PSSM) and two-dimensional Direct Coupling Analysis (DCA) information.
- The method employs an ensemble of deep neural networks, optimized from SPOT-RNA to improve prediction accuracy.

### Results:

SPOT-RNA2 demonstrates significant improvements in predicting canonical and non-canonical base pairs as well as tertiary interactions like pseudoknots. The method achieves an F1-score above 0.8 for 14 out of 16 tested RNAs with more than 1000 homologous sequences. The integration of artificial homologous sequences generated through deep

mutational scanning further improves accuracy.

### **Availability and Implementation:**

SPOT-RNA2 is available as standalone software and via a web server. The datasets used can be downloaded from GitHub and the web server.

### **Performance Evaluation:**

Performance is assessed using F1-score and Matthews correlation coefficient (MCC). SPOT-RNA2 outperforms existing RNA secondary structure prediction methods in various test scenarios, including pseudoknot and non-canonical base pair prediction.

### **Comparison with Existing Methods:**

SPOT-RNA2 is compared with other single-sequence and alignment-based prediction methods. Results show consistently higher accuracy, particularly for RNAs with complex base-pairing patterns.

### **Conclusion:**

SPOT-RNA2 is a powerful tool for RNA structure prediction, demonstrating the potential of evolutionary information to enhance prediction accuracy. The method is particularly beneficial for RNAs with a large number of homologous sequences, improving prediction accuracy.

### **Future Perspectives:**

Combining natural and artificial homologous sequences could further improve prediction accuracy. The development of more efficient algorithms for handling longer RNA sequences and reducing computational time is a future goal.

### **Contact Information:**

Jaswinder Singh: jaswinder.singh3@griffithuni.edu.au Yaoqi Zhou: yaoqi.zhou@griffith.edu.au

## 29 DeepFoldRNA[2]

### Title:

De Novo RNA Tertiary Structure Prediction at Atomic Resolution Using Geometric Potentials from Deep Learning

### Authors:

Robin Pearce, Gilbert S. Omenn, Yang Zhang

### Introduction:

RNA molecules play a crucial role in many biological processes, with their functions strongly dependent on their three-dimensional structures. There is a significant gap between the number of known RNA sequences and experimentally determined RNA structures. Effective computational methods for RNA structure prediction are urgently needed, particularly for non-coding RNAs.

### DeepFoldRNA:

DeepFoldRNA is a novel method combining deep self-attention neural networks and gradient-based folding simulations. The approach aims to determine the spatial position of each atom in an RNA molecule solely from its nucleotide sequence.

### Methods:

**Restraint Generation Module:** Multiple sequence alignments (MSAs) are created through iterative searches in RNA databases. Self-attention neural networks predict spatial constraints (restraints) such as pairwise distances and backbone torsion angles.

**Structure Construction Module:** Predicted geometric restraints are converted into composite potentials guiding L-BFGS folding simulations. These simulations enable rapid and precise RNA folding.

### Results:

DeepFoldRNA was tested on two independent benchmark datasets: one from Rfam families and one from RNA-Puzzles. The method achieved an average RMSD of 2.68 Å and a TM-score of 0.757, significantly improving over existing methods. DeepFoldRNA folds medium-sized RNAs in about one minute on average, 350-4000 times faster than leading Monte Carlo simulation approaches.

### Applications:

Its high speed and accuracy allow large-scale applications in atomic RNA structure mod-

eling. The method could be extended to RNA-protein interactions and RNA complexes to better understand molecular and cellular functions.

## 30 RNA-MSM[3]

### **Title:**

Multiple Sequence Alignment-based RNA Language Model and Its Application to Structural Inference

### **Authors:**

Yikun Zhang, Mei Lang, Jiuhong Jiang, et al.

### **Introduction:**

RNA and DNA are harder to interpret than proteins, as they have only four letters compared to twenty for proteins. Previous RNA language models based on BERT-like architectures fail to capture evolutionary information effectively. An unsupervised RNA language model based on multiple sequence alignments (MSA) could improve RNA structure and function prediction.

### **RNA-MSM:**

RNA-MSM was developed to utilize homologous RNA sequences from RNAcmap, generating more homologs than manually curated databases like Rfam. The model uses an unsupervised learning architecture producing two-dimensional attention maps and one-dimensional embeddings encoding structural information.

### **Results:**

RNA-MSM’s generated attention maps and embeddings accurately predict 2D base-pairing probabilities and 1D solvent accessibilities. The model outperforms techniques like SPOT-RNA2 and RNAsnap2 in base pair and solvent accessibility predictions. The study emphasizes challenges in RNA structure prediction compared to proteins and highlights the importance of evolutionary information.

### **Availability:**

Source codes, models, and datasets are publicly available on Zenodo.

### 31 Sean Eddy and Elena Rivas[\[4\]](#)

Sean Eddy and Elena Rivas, renowned computational biologists from the Janelia Research Campus, are joining the Department of Molecular and Cellular Biology (MCB) this fall. Eddy will serve as a professor, while Rivas will take on the role of senior research fellow and lecturer. Their work focuses on the significance of RNA in evolution, utilizing advanced computational tools to analyze RNA sequences' structure, function, and evolutionary history.

Eddy's journey into science began in rural Pennsylvania, inspired by his mother, who pursued biology while raising six children. He initially studied physics at Caltech but switched to biology, eventually earning a PhD in molecular biology at the University of Colorado. His research on self-splicing RNA and the movement of introns led him to develop algorithms for RNA analysis, culminating in the creation of HMMER and Infernal software.

Rivas, originally a particle physicist, transitioned to biology after realizing her passion for problem-solving in a more rewarding field. She joined Eddy's lab, where she applied her physics background to RNA research, developing programs to identify RNA genes and their evolutionary significance.

At Janelia, both Eddy and Rivas made significant contributions to RNA research and software development. Eddy focused on enhancing Infernal and exploring cell-type specific genomics, while Rivas worked on probabilistic models for genome sequence interpretation.

Now at MCB, they are excited to build a new lab, collaborate with diverse researchers, and tackle new scientific challenges. Eddy, who has a history with MCB, feels a strong connection to the department, viewing it as a second home. Together, they look forward to refining their current projects and exploring new avenues in RNA research and beyond.

## 32 Evolutionary conservation of RNA sequence and structure[5]

The article by Elena Rivas titled "Evolutionary conservation of RNA sequence and structure" provides a comprehensive analysis of the role of RNA in biological processes and the challenges associated with predicting its structural properties. In recent decades, scientific interest has increasingly shifted from DNA to RNA, driven by the recognition that RNA serves not only as an intermediary molecule in protein synthesis but also plays a crucial role in various regulatory and structural functions within the cell. In particular, non-coding RNAs, including long non-coding RNAs (lncRNAs), have been identified as essential for gene regulation and the maintenance of cellular homeostasis.

### Importance of RNA Structure

Rivas argues that predicting RNA structure from a single sequence is insufficient to establish the functional relevance of that structure. Many random RNA sequences can exhibit complex and plausible structures that are indistinguishable from those of functional RNAs. Therefore, it is critical to analyze the evolutionary signatures left by conserved RNAs to determine whether an RNA possesses a functionally relevant structure. The article emphasizes that identifying conserved RNA structures is not only vital for understanding RNA function but also for reconstructing the evolutionary history of organisms.

### Methods for Analyzing RNA Conservation

The article describes various statistical approaches for measuring the conservation of RNA structures, particularly the application of covariation analyses. These methods allow for the differentiation between structural RNA conservation and covariation arising from independent phylogenetic substitutions. Rivas highlights the necessity of identifying false positives that may arise from artifacts, such as the inclusion of pseudogenes in alignments, as these can significantly compromise the analysis of RNA structure conservation.

### Covariation and Variability

A central theme of the article is the analysis of sequence variations and their patterns to support or refute the existence of a conserved RNA structure. Rivas explains that a conserved RNA sequence does not necessarily imply a conserved structure. Instead, supporting evidence for a conserved structure requires a specific pattern of variations indicative of evolutionary preservation. The article underscores that there are also variation patterns that support the absence of a conserved structure, illustrating the complexity of RNA evolution.

### Challenges in Identifying Conserved RNA Structures

Rivas discusses the challenges associated with identifying new evolutionarily conserved



RNA structures. She emphasizes that the integration of sequence and structural information, along with the consideration of covariation and variability, is crucial for understanding the evolutionary significance of RNA. The article concludes by stating that developing methods for analyzing RNA structure and function, which incorporate both evolutionary and structural information, is of paramount importance for elucidating the mechanisms of RNA function and evolution.

## **Conclusion**

Overall, Rivas's article offers profound insights into the current challenges and advancements in RNA research, particularly regarding the identification and analysis of conserved RNA structures. The findings from this research could not only deepen the understanding of the biological functions of RNA but also pave the way for new approaches to investigate the evolution of RNA molecules and their roles in cellular regulation and function.

### 33 sincFold[6][7]

The article presents "sincFold," a novel end-to-end deep learning model designed for predicting RNA secondary structures by learning both short- and long-range interactions from RNA sequences. This research addresses the longstanding challenge of accurately predicting RNA secondary structures, which are crucial for understanding the functional roles of RNA molecules in biological processes.

#### Motivation and Background

RNA molecules, including both coding and non-coding RNAs, play vital roles in various biological functions. The ability to predict their secondary structures from sequences is essential for elucidating their functions and evolutionary significance. Traditional methods for RNA secondary structure prediction have relied on thermodynamic models and dynamic programming, which have limitations in performance. Recent advancements in deep learning have shown promise in improving prediction accuracy, yet there remains significant room for enhancement.

#### Methodology

sincFold employs a two-stage deep learning architecture that integrates both one-dimensional (1D) and two-dimensional (2D) residual neural networks. The model takes a one-hot encoded RNA sequence as input and predicts a contact matrix that represents nucleotide interactions. The first stage focuses on learning local patterns in the 1D representation of the sequence, while the second stage captures distant relationships through a 2D representation.

#### Key components of the sincFold architecture include

- **1D Residual Networks** These networks facilitate the learning of short-range interactions by stacking identity blocks that help propagate signals and mitigate vanishing gradient issues.
- **2D Residual Networks** After obtaining a 1D encoding, the model transitions to a 2D representation to learn long-range interactions, enhancing the model's ability to predict complex RNA structures.

#### Results

Extensive experiments were conducted using several benchmark datasets, including RNAs-tralign and ArchiveII, to evaluate the performance of sincFold. The model demonstrated superior accuracy in predicting RNA secondary structures compared to classical methods and other state-of-the-art deep learning approaches. The results indicate that sincFold

can effectively learn the intricate patterns of RNA interactions with minimal physical assumptions.

### **Performance Metrics**

The performance of sincFold was assessed using metrics such as the F1 score, which evaluates the accuracy of predicted base pairs against reference structures. The model achieved high F1 scores, particularly excelling in the prediction of both canonical and non-canonical base pairs.

### **Conclusion**

sincFold represents a significant advancement in RNA secondary structure prediction, leveraging deep learning techniques to capture both short- and long-range interactions effectively. The model’s end-to-end learning approach allows for accurate predictions without the need for multiple sequence alignments or extensive pre-processing. The authors have made the source code publicly available, facilitating further research and development in the field of RNA bioinformatics.

## 34 Tasks for KW8

- Revise the presentation
- Ask Frau Graf if she has already created a user for me so that I can run through my pipeline again.
- Reconfigure the server for my pipeline
- Research about native bacillus subtilis alignments
- Start with the Master's thesis (introduction, methods used, ...)
- Read following Papers and make a summary
  - <https://pubmed.ncbi.nlm.nih.gov/33845850/>
  - <https://pubmed.ncbi.nlm.nih.gov/36596869/>
  - <https://pubmed.ncbi.nlm.nih.gov/39526405/>
  - <https://pubmed.ncbi.nlm.nih.gov/39526405/>

## 35 References

- [1] <https://doi.org/10.1093/bioinformatics/btab165>
- [2] <https://doi.org/10.1101/2022.05.15.491755>
- [3] <https://doi.org/10.1093/nar/gkad1031>
- [4] <https://www.mcb.harvard.edu/departement/news/mcb-welcomes-sean-eddy-and-elena-riva>
- [5] <https://pubmed.ncbi.nlm.nih.gov/33754485/>
- [6] <https://pubmed.ncbi.nlm.nih.gov/38855913/>
- [7] <https://github.com/sinc-lab/sincFold>

Summary KW8 and KW9

## 36 Installation of the required software on the new server

All tools previously used on the old server were reinstalled and their functionality checked. In addition, the installation was adapted so that the programs are accessible user-wide and can be executed from any directory.

## 37 Adaptation of the presentation

The updated pipeline was added to the presentation. In addition, the results of the time measurements for the various tools used to generate the negative samples were integrated.

## **38 Performance measurement of the SSSIz predictions**

After successfully setting up the server, a new run with SSSIz was carried out. A time stamp was recorded at regular intervals (after every 1,000 predictions) and written to a log file. In addition, the number of CPU cores used was increased from 24 to 64, reducing the expected runtime to an average of 10 days.

## **39 Native Alignments from B.subtilis**

A search of the RFAM database for alignments of *Bacillus subtilis* identified 62 sequences, each comprising between 370 and 401 base pairs. The central question is whether these sequences can be used as a reference for a comparison with the alignments randomized by SSSI.

## 40 Comparative genomics identifies thousands of candidate structured RNAs in human microbiomes[\[1\]](#)

### **Industruction:**

In their study “Comparative genomics identifies thousands of candidate structured RNAs in human microbiomes”, Brayon J. Fremin and Ami S. Bhatt investigated the presence and diversity of structured RNAs within the human microbiome. Structured RNAs perform diverse bioregulatory functions in microbes. Although hundreds of such RNAs have been predicted using informative approaches, much of the metagenomic data of the human microbiome remained unexplored due to computational limitations.

### **Methods:**

The authors developed a pipeline to identify potential structured RNAs. First, genes in the Human Microbiome Project 2 (HMP2) were annotated with Prodigal. They then identified homologous intergenic regions using HS-BLASTN. Conserved regions were grouped and scored using RNAPhylo, R-scape and RNAcode. Known RNAs from the Rfam database were excluded, and remaining candidates were matched against HMP2 using cmsearch to ensure unique hits. Finally, these regions were matched against the nr database using BLASTx to ensure that they were not protein coding.

### **Results:**

Using this approach, the researchers identified 3,161 potential structured RNAs and 2,022 additional candidates with overlaps to the nr database. A large number of these RNAs, including tmRNAs, antitoxins and ribosomal protein leaders, were found in different taxa. Genomic neighborhood analysis revealed frequent associations with specific protein domains, indicating possible regulatory functions.

### **Conclusions:**

The developed pipeline enables conservative predictions of thousands of new potential structured RNAs in the human microbiome. These discoveries expand our understanding of RNA-based regulation in microbial communities and provide insights into the complex bioregulatory networks of the human microbiome.

## 41 Long non-coding RNAs: definitions, functions, challenges and recommendations[\[2\]](#)

### **Industruction:**

In the review article “Long non-coding RNAs: definitions, functions, challenges and recommendations”, the authors shed light on the diverse aspects of long non-coding RNAs (lncRNAs). These RNAs, which are over 200 nucleotides long and are not translated into proteins, make up a significant proportion of the genomes of complex organisms

### **Definition and classification:**

The term ‘lncRNAs’ encompasses a variety of transcripts, including those transcribed by RNA polymerase I, II, and III, as well as RNAs derived from processed introns. The variety of functions, isoforms and overlap with other genes complicates the classification and annotation of lncRNAs.

### **Functions of lncRNAs:**

lncRNAs play a critical role in regulating many aspects of cell differentiation, development, and other physiological processes. Many lncRNAs bind to chromatin-modifying complexes, are transcribed by enhancers and promote the formation of nuclear condensates, indicating a close link between lncRNA expression and the spatial control of gene expression during development.

### **Challenges in research:**

The rapid evolution of lncRNAs compared to protein-coding sequences, their cell-specific expression and the diversity of their functions pose considerable challenges for researchers. The identification of interaction partners and the determination of the specific functions of individual lncRNAs require comprehensive studies.

### **Recommendations for future research:**

The authors emphasize the need for standardized methods for annotation and functional characterization of lncRNAs. They recommend the development of new technologies and approaches to better understand the complex roles of lncRNAs in cell biology and disease.

## 42 RFAM 15: RNA families database in 2025[3]

### 43 Introduction

The Rfam database is a comprehensive collection of RNA families represented by multiple sequence alignments, consensus secondary structures and covariance models. With the release of version 15.0 in September 2024, several significant updates have been made.

#### **Expansion of Rfamseq:**

The number of genomes in the Rfamseq database was increased from 14,772 in version 14.0 to 26,106 in version 15.0, an increase of 76 percent. This expansion aims to better reflect the genomes currently available.

#### **Integration of 3D structures:**

In version 15.0, 25 additional RNA families have been reviewed and annotated with 3D information, bringing the total number of families containing such structural data to 65. This integration improves the accuracy of consensus secondary structures and annotations.

#### **Provision of a public MySQL database:**

Rfam now provides a publicly accessible, read-only MySQL database that is updated with each new version. This facilitates access to the latest data for the research community.

## 44 Tasks for KW10

- Make the changes to the presentation discussed in the meeting with Tanja
- Compare Time Prediction from RNAz and SISSIZ
- Research about native bacillus subtilis alignments
- Start with the Master's thesis (introduction, methods used, ...)

## 45 References

- [1] <https://pubmed.ncbi.nlm.nih.gov/33845850/>
- [2] <https://pubmed.ncbi.nlm.nih.gov/36596869/>
- [3] <https://pubmed.ncbi.nlm.nih.gov/39526405/>

Summary KW13 - KW16



## 46 Introduction to the ROC Curve Code

A function is defined here that compares two data sets (positive and negative), adds artificial noise and then evaluates them using a random forest classifier. Parameters such as noise scale (strength of the noise) or labels can also be adjusted.

```
1 def evaluate_classifier_with_noise(df_positive, df_negative, noise_scale=10,
    pos_label=1, neg_label=0, title_suffix=""):
```

The most important libraries for data analysis, visualization and machine learning are imported.

```
1     # import the necessary libraries
2     import numpy as np
3     import pandas as pd
4     import matplotlib.pyplot as plt
5     import seaborn as sns
6     from sklearn.ensemble import RandomForestClassifier
7     from sklearn.model_selection import cross_val_predict
8     from sklearn.metrics import confusion_matrix, classification_report,
        roc_curve, auc
```

The transferred data records are copied (so as not to change the original data). They are then given a label (1 for positive, 0 for negative), which is required later for training the classifier.

```
1     # Set the labels
2     df_positive = df_positive.copy()
3     df_negative = df_negative.copy()
4     df_positive['Label'] = pos_label
5     df_negative['Label'] = neg_label
```

Here, additional noise is added to each data set. This results in a deliberate overlap of the positive and negative data points. Without this overlap, the classification task would be too trivial and not realistic. This makes the classification more realistic and more challenging for the model (random forest).

```
1     # Add noise to the
2     df_positive['z-score calculated from 7. 8. and 9.'] += np.random.normal(loc
        =0, scale=noise_scale, size=len(df_positive))
3     df_negative['z-score calculated from 7. 8. and 9.'] += np.random.normal(loc
        =0, scale=noise_scale, size=len(df_negative))
```

The two data sets are merged into a common DataFrame, which simplifies subsequent

processing (training and visualization).

```
1 # Combine the datasets
2 data = pd.concat([df_positive, df_negative], ignore_index=True)
```

The comparison between the two datasets is visualized to illustrate the distribution of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

```
1 # Visualize the Comparison between the two Datasets to illustrate the TN,
  FN, TP and FP
2 sns.histplot(data=data, x='z-score calculated from 7. 8. and 9.', hue='
  Label', kde=True, bins=50)
3 plt.title(f"Distribution of z-scores with noise {title_suffix}")
4 plt.xlabel("z-Score")
5 plt.ylabel("Anzahl")
6 plt.show()
```

The feature matrix X and the target variable y are prepared. A random forest classifier with 100 decision trees is then created. `random_state = 42` ensures reproducibility.

```
1 # Preparation for the Randomforest Classifier
2 X = data[['z-score calculated from 7. 8. and 9.']]
3 y = data['Label']
4 model = RandomForestClassifier(n_estimators=100, random_state=42)
```

The model is evaluated with 5-fold cross-validation to avoid overfitting. Both the predicted classes ( $y_{pred}$ ) and the probabilities for the positive class ( $y_{proba}$ ) are calculated.

```
1 # Crossvalidation is added to avoid overfitting
2 y_pred = cross_val_predict(model, X, y, cv=5, method='predict')
3 y_proba = cross_val_predict(model, X, y, cv=5, method='predict_proba')[:,
  1]
```

The confusion matrix shows the number of correctly and incorrectly classified cases.

- True Negative = RNA structure was destroyed and correctly predicted
- True Positive = RNA structure was not destroyed and correctly predicted
- False Negative = RNA structure was not destroyed but not correctly predicted
- False Positive = The RNA structure was destroyed, but the model incorrectly predicted it as intact (false positive)

The classification report provides key figures such as Precision, Recall, F1-Score and Accuracy - these are important metrics for evaluating the classifier.

```
1      # Evaluating of the Confusion Matrix and the Classification
2      print(" Confusion Matrix:\n", confusion_matrix(y, y_pred))
3      print("\n Classification Report:\n", classification_report(y, y_pred))
```

The ROC curve compares the true positive rate against the false positive rate for different threshold values. The AUC value (Area Under Curve) measures the quality of the separation between the classes.

```
1      # Calculation of the AUC
2      fpr, tpr, thresholds = roc_curve(y, y_proba)
3      roc_auc = auc(fpr, tpr)
```

The ROC curve compares the true positive rate against the false positive rate for different threshold values. The AUC value (Area Under Curve) measures the quality of the separation between the classes.

```
1      # Plot ROC
2      plt.figure()
3      plt.plot(fpr, tpr, label=f"ROC (AUC = {roc_auc:.2f})", color='red')
4      plt.plot([0, 1], [0, 1], linestyle='--', color='gray')
5      plt.xlabel("False Positive Rate")
6      plt.ylabel("True Positive Rate")
7      plt.title(f"ROC-Curve {title_suffix}")
8      plt.legend(loc="lower right")
9      plt.grid(True)
10     plt.show()
```

Load the positive and negative Datasets

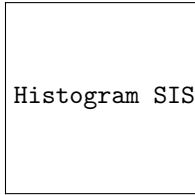
```
1      import pandas as pd
2
3      # Load the data
4      df_sissi = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data/
          SISSIz_Excel/sissi.xlsx", usecols=['z-score calculated from 7. 8. and 9.'])
5      df_sissiz_mono = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data/
          SISSIz_Excel/sissiz_mono.xlsx", usecols=['z-score calculated from 7. 8. and
          9.'])
6      df_sissiz_di = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data/
          SISSIz_Excel/sissiz_di.xlsx", usecols=['z-score calculated from 7. 8. and 9.
          '])
7      df_multiperm_mono = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data
          /SISSIz_Excel/multiperm_mono.xlsx", usecols=['z-score calculated from 7. 8.
          and 9.'])
8      df_multiperm_di = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data/
```

```
        SSISSiz_Excel/multiperm_di.xlsx", usecols=['z-score calculated from 7. 8. and  
        9.'])  
9 df_aln_shuffle = pd.read_excel("D:/Masterarbeit_programmieren/2.Versuch/Data/  
        SSISSiz_Excel/alifoldz.xlsx", usecols=['z-score calculated from 7. 8. and 9.'  
        ])  
10  
11 noise_scale_for_all=5
```

## 47 SISSI vs SISSIZ MONO

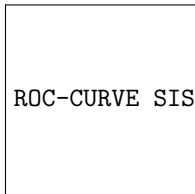
```
1 evaluate_classifier_with_noise(df_sissi, df_sissiz_mono, noise_scale=  
    noise_scale_for_all, title_suffix="SISSI vs SISSIZ_MONO")
```

```
1 Confusion Matrix:  
2 [[491  9]  
3  [ 7 493]]  
4  
5 Classification Report:  
6           precision    recall  f1-score   support  
7  
8      0           0.99       0.98       0.98         500  
9      1           0.98       0.99       0.98         500  
10  
11   accuracy                0.98         1000  
12   macro avg           0.98       0.98       0.98         1000  
13   weighted avg          0.98       0.98       0.98         1000
```



Histogram Sissi vs Sissiz\_mono.png

**Figure 7:** Histogram Sissi vs Sissiz-mono



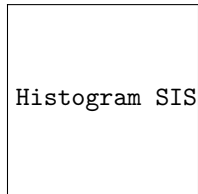
ROC-CURVE Sissi vs Sissiz\_mono.png

**Figure 8:** ROC-CURVE Sissi vs Sissiz-mono

## 48 SISSI vs SISSIZ DI

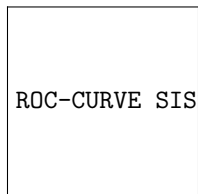
```
1 evaluate_classifier_with_noise(df_sissi, df_sissiz_di, noise_scale=  
    noise_scale_for_all, title_suffix="SISSI vs SISSIZ_DI")
```

```
1 Confusion Matrix:  
2 [[482  18]  
3  [ 16 484]]  
4  
5 Classification Report:  
6  
7      precision    recall  f1-score   support  
8  
9      0       0.97       0.96       0.97         500  
10     1       0.96       0.97       0.97         500  
11  
12    accuracy          0.97  
13    macro avg       0.97       0.97       0.97  
    weighted avg       0.97       0.97       0.97
```



Histogram SISSI vs SISSIZ-di.png

**Figure 9:** Histogram SISSI vs SISSIZ-di



ROC-CURVE SISSI vs SISSIZ-di.png

**Figure 10:** ROC-CURVE SISSI vs SISSIZ-di

## 49 SISSI vs MULTIPERM MONO

```
1 evaluate_classifier_with_noise(df_sissi, df_multiperm_mono, noise_scale=
    noise_scale_for_all, title_suffix="SISSI vs MULTIPERM_MONO")
2
3
4 \begin{lstlisting}
5 Confusion Matrix:
6 [[467  33]
7  [ 32 468]]
8
9 Classification Report:
10
11      precision    recall  f1-score   support
12
13     0       0.94      0.93      0.93        500
14     1       0.93      0.94      0.94        500
15
16    accuracy                0.94       1000
17    macro avg       0.94      0.94      0.93       1000
18    weighted avg       0.94      0.94      0.93       1000
```

Histogram SISSI vs MULTIPERM\_mono.png

**Figure 11:** Histogram SISSI vs MULTIPERM-mono

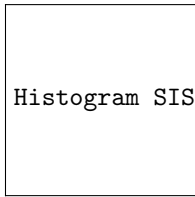
ROC-CURVE SISSI vs MULTIPERM\_mono.png

**Figure 12:** ROC-CURVE SISSI vs MULTIPERM-mono

## 50 SISSI vs MULTIPERM DI

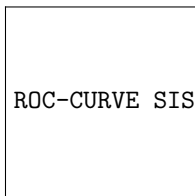
```
1 evaluate_classifier_with_noise(df_sissi, df_multiperm_di, noise_scale=  
    noise_scale_for_all, title_suffix="SISSI vs MULTIPERM_DI")
```

```
1 Confusion Matrix:  
2 [[475  25]  
3  [ 31 469]]  
4  
5 Classification Report:  
6           precision    recall  f1-score   support  
7  
8      0           0.94       0.95       0.94         500  
9      1           0.95       0.94       0.94         500  
10  
11   accuracy                0.94         1000  
12   macro avg           0.94       0.94       0.94         1000  
13   weighted avg           0.94       0.94       0.94         1000
```



Histogram SISSI vs MULTIPERM\_di.png

**Figure 13:** Histogram SISSI vs MULTIPERM-di



ROC-CURVE SISSI vs MULTIPERM\_di.png

**Figure 14:** ROC-CURVE SISSI vs MULTIPERM-di

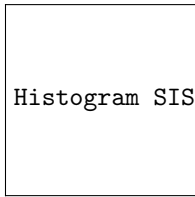


## 51 SISSI vs ALN-SHUFFLE

```
1 evaluate_classifier_with_noise(df_sissi, df_aln_shuffle, noise_scale=  
    noise_scale_for_all, title_suffix="SISSI vs ALN-SHUFFLE")
```

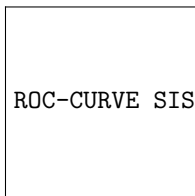
```
1 Confusion Matrix:  
2 [[399 101]  
3 [ 97 403]]  
4  
5 Classification Report:  
6  
7  
8  
9  
10  
11  
12  
13
```

		precision	recall	f1-score	support
	0	0.80	0.80	0.80	500
	1	0.80	0.81	0.80	500
	accuracy			0.80	1000
	macro avg	0.80	0.80	0.80	1000
	weighted avg	0.80	0.80	0.80	1000



Histogram SISSI vs ALN-SHUFFLE.png

**Figure 15:** Histogram SISSI vs ALN-SHUFFLE



ROC-CURVE SISSI vs ALN-SHUFFLE.png

**Figure 16:** ROC-CURVE SISSI vs ALN-SHUFFLE

## 52 Summary of the Results

The ROC curve compares the true positive rate against the false positive rate for different threshold values. The AUC value (Area Under Curve) measures the quality of the separation between the classes.

Die beste Klassifikationsleistung wurde bei der Gegenüberstellung von SISSI und SISSIz MONO erzielt. Das Modell erreichte eine Accuracy von 98 Prozent bei einem F1-Score von 0,98 für beide Klassen. Die Anzahl an Fehlklassifikationen war mit lediglich 9 False Positives und 7 False Negatives minimal, was auf eine sehr klare Trennbarkeit der beiden Datenquellen hinweist.

The ROC curve compares the true positive rate against the false positive rate for different threshold values. The AUC value (Area Under Curve) measures the quality of the separation between the classes.

The control groups based on MULTIPERM achieved an accuracy of 94 percent with a mean F1 score of 0,94. The MULTIPERM MONO and MULTIPERM DI configurations delivered very similar results, with MULTIPERM DI showing a minimally better recall value for the positive class. Overall, however, these values indicate that MULTIPERM-based shuffling is structurally closer to the SISSI data than the SISSIz controls.

A significantly lower classification quality was observed in the last comparison with ALN-SHUFFLE (Accuracy 80 percent, F1-Score 0,80). However, the high number of misclassifications (101 FP, 97 FN) suggests a strong structural similarity to the positive class, which made separation by the model considerably more difficult.

In summary, the SISSIz MONO method provides the clearest separation from the original structure and can therefore be identified as the most effective negative control method within this classification framework. It provides a well-distinguishable contrast image, which is ideally suited for further analysis or validation steps.

## 53 SPOT-RNA2

I am currently installing SPOT-RNA2 on the FH AI server but the databases are still missing and should be fully downloaded in 2 days.

## 54 DeepFoldRNA

I installed DeepFoldRNA on the FH AI server, but encountered an issue during execution: DeepFoldRNA only accepts single FASTA sequences. After reducing the input to individual sequences, the execution started successfully. However, the process took more than 12 hours and eventually stopped without producing meaningful results. Due to this, I have decided to replace DeepFoldRNA with MXfold2.

## 55 RNA secondary structure prediction using deep learning with thermodynamic integration[\[1\]](#)[\[2\]](#)

### Introduction

Accurate prediction of RNA secondary structure is essential for understanding the function of non-coding RNAs. Traditional models use free energy minimization, while recent approaches have applied deep learning. However, deep learning often overfits to the training data, limiting generalizability. The aim of this study is to integrate deep learning with thermodynamic models to achieve a more robust and generalizable RNA secondary structure predictor.

### Methods

The authors developed MXfold2, which computes folding scores using a deep neural network and integrates these with experimentally derived thermodynamic parameters from Turner’s model.

Key innovations include:

- **Thermodynamic Integration:** Combining deep learning–based scores with known free energy rules.
- **Thermodynamic Regularization:** A loss term that forces the neural network’s outputs to align with thermodynamic energy predictions. The model uses bidirectional LSTMs and convolutional neural networks to capture both local and long-range dependencies in RNA sequences.

### Results

MXfold2 was benchmarked against CONTRAfold, RNAfold, SPOT-RNA, and E2Efold using:

- Sequence-wise Cross-validation: MXfold2 achieved significantly higher accuracy.
- Family-wise Cross-validation: Unlike other models, MXfold2 maintained high performance and avoided overfitting.
- Independent Test Sets: The method outperformed competitors even on previously unseen RNA families. Moreover, MXfold2 was computationally efficient and scalable to longer sequences.

## Discussion

The authors emphasize that integrating data-driven learning with physics-based modeling yields synergistic benefits. The approach captures the adaptability of deep learning while maintaining consistency with physical principles. This duality enhances both accuracy and generalization. They suggest this framework could be extended to related problems like RNA tertiary structure prediction and RNA-protein interactions.

## Conclusion

MXfold2 represents a significant step forward in RNA bioinformatics. By combining thermodynamic modeling with deep learning, it delivers accurate and generalizable RNA secondary structure predictions—crucial for studying novel non-coding RNAs.

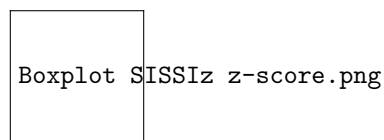
## 56 References

- [1] <https://doi.org/10.1038/s41467-021-21194-4>  
 [2] <https://github.com/mxfold/mxfold2?tab=readme-ov-file>

Summary KW18 - KW22

## 57 Results of SSSIz Prediction

### 57.1 SSSIz Boxplot



**Figure 17:** Boxplot SSSIz z-score

The boxplots clearly show the selectivity of the different models. In particular, SSSIz-mono and SSSIz-di stand out as the most effective in destroying the RNA secondary structure. In contrast, the other models showed little to no destructive effect. The worst result was achieved by ALN-Shuffle, which could hardly affect the structure of the RNA.



## 57.2 SISIz Histograms



**Figure 18:** Histograms from SISIz prediction

The histograms also confirm the results from the boxplots: The separability of the models is easily recognizable, and again *SISSIZ-mono* and *SISSIZ-di* show the best performance.





57.3 SISIz Confusion Matrix



Figure 19: Confusion Matrix from SISIz prediction

The evaluation of the confusion matrices illustrates the concrete separation performance of the respective models.

The SSSIz-mono and SSSIz-di models each achieved 0 false positives and 0 false negatives like in the Figure 3 (a) and (b), which indicates an almost perfect separation of the classes. These results indicate excellent separability between the original and the negatively controlled RNA structures.

In contrast, the multiperm-mono and multiperm-di models showed solid classification performance, but with some misclassifications:

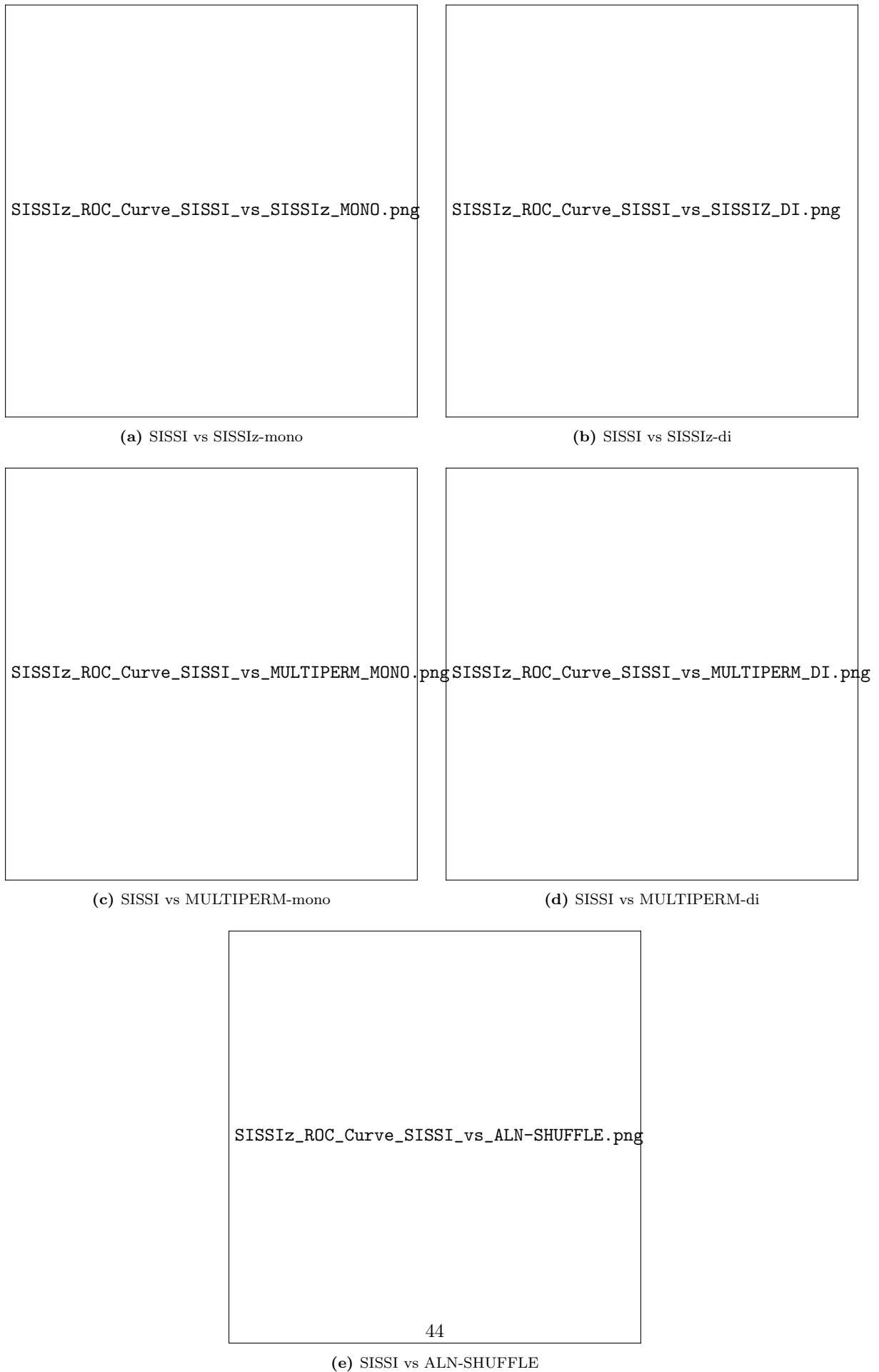
Multiperm-mono Figure 3 (c): 23 false positives, 29 false negatives

Multiperm-di Figure 3 (d): 17 false positives, 21 false negatives

The worst separability was found in the Aln-Shuffle Figure 3 (e) scenario, with 3877 false positives and 5327 false negatives. These high misclassification numbers illustrate the difficulty of reliably distinguishing this model from the original structure.



## 57.4 SISSIZ ROC-Kurven



**Figure 20:** ROC-curves from SISSIZ prediction

The ROC curve compares the true-positive rate with the false-positive rate for various threshold values. The AUC value (Area Under the Curve) measures the quality of the separation between the classes.

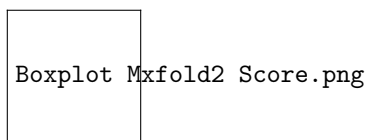
The analysis of the classification performance showed that the SISSI model achieved excellent results in almost all comparison scenarios. Particularly striking was the AUC value in the Aln-Shuffle scenario, which was exceptionally high at 0,99 - despite the overall poorer separability in this case.

SISSIZ-mono and SSISSIZ-di also showed very high AUC values, confirming their strong discriminatory power. The multiperm variants were slightly lower, but also showed useful classification results. In comparison, the Aln-Shuffle model presented the greatest challenge and, with an AUC of 0,95, achieved the lowest value of all the methods tested.

## 58 Results of MXfold2 Prediction

### 58.1 RNA secondary structure prediction using deep learning with thermodynamic integration[\[1\]](#)[\[2\]](#)

#### 58.2 MXfold2 Boxplot



**Figure 21:** Boxplot MXfold2 Score

The boxplot of MXfold2 clearly shows that the RNA secondary structure was successfully destroyed. However, MXfold2 shows difficulties in handling the simulated data from SISSI, which leads to an inaccurate prediction of whether the secondary structure was actually destroyed or not. This observation could also explain why the Multiperm-mono and Multiperm-di methods achieve almost equivalent results to SSSIz-mono and SSSIz-di, although the evaluations of RNAz and SSSIz consistently show that the SSSIz-mono and SSSIz-di methods generally perform better than the other approaches.



## 58.3 MXfold2 Histograms



**Figure 22:** Histograms from MXfold2 prediction



When analyzing the histograms of the individual models, it becomes apparent that the positive and negative data are strongly intertwined. This indicates that the predictive accuracy of the random forest model may be limited. One possible explanation for this could be that all five histograms show almost identical results, which indicates that the MXfold2 model has difficulties in dealing with simulated data.



58.4 MXfold2 Confusion Matrix



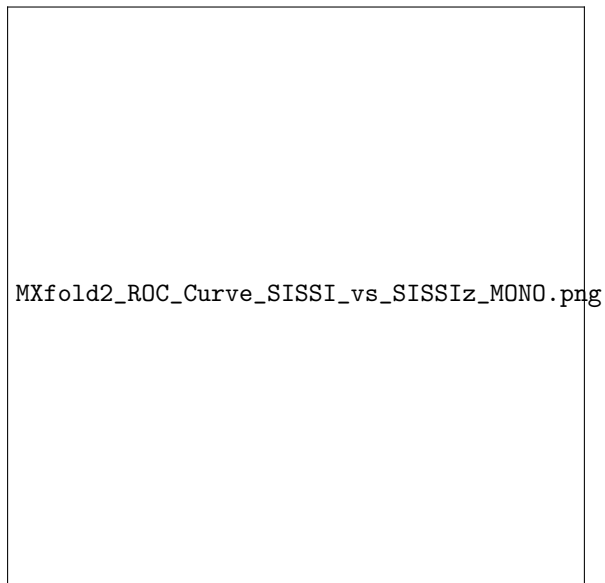
Figure 23: Confusion Matrix from MXfold2 prediction

The trends observed in the boxplot and histogram are further underpinned by the analysis of the confusion matrix. The separability of the models proves to be suboptimal. For the SISSIZ-mono model (Figure a), 15,535 false-positive and 28,569 false-negative results were determined. The SISSIZ-di model (Figure b) showed 17,546 false-positive and 34,018 false-negative results.

The Multiperm-mono model (Figure c) yielded 9,834 false-positive and 32,476 false-negative results, while the Multiperm-di model (Figure d) performed similarly with 10,583 false-positive and 31,270 false-negative results. The Aln-shuffle model (Figure e) showed the worst results with 12,691 false-positive and 44,002 false-negative results.

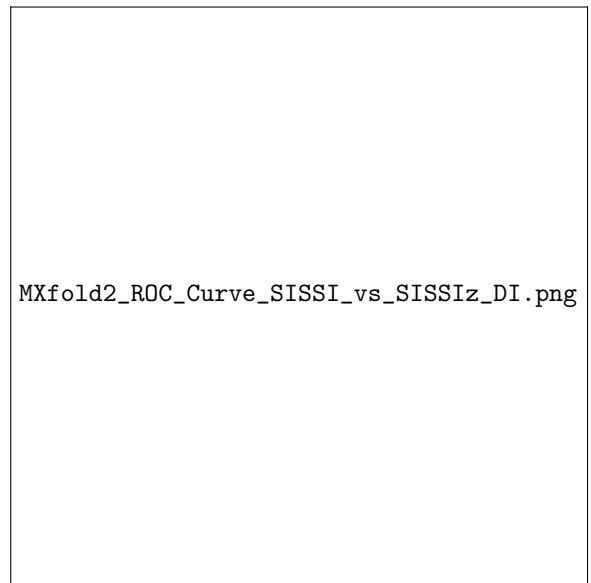


## 58.5 MXfold2 ROC-Kurven



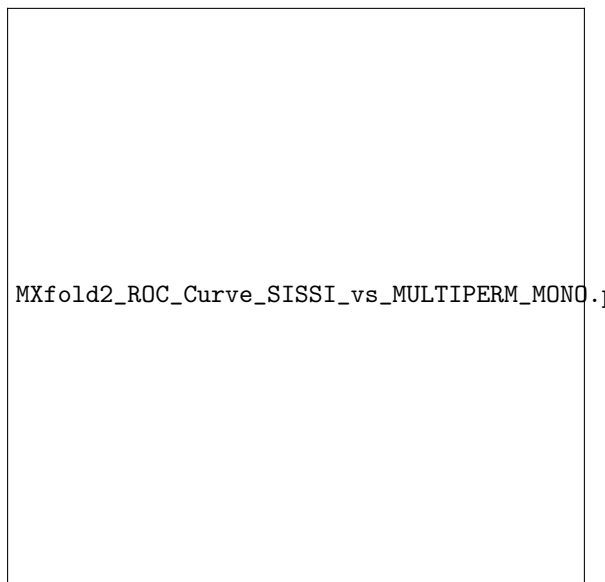
MXfold2\_ROC\_Curve\_SISSI\_vs\_SISSIZ\_MONO.png

(a) SISSI vs SISSIZ-mono



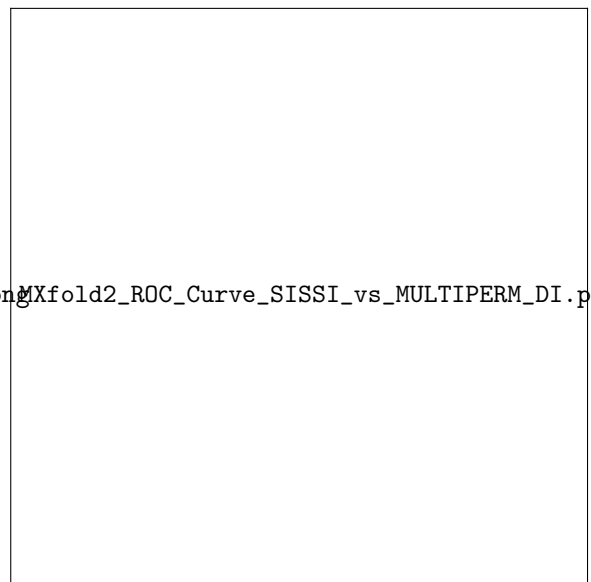
MXfold2\_ROC\_Curve\_SISSI\_vs\_SISSIZ\_DI.png

(b) SISSI vs SISSIZ-di



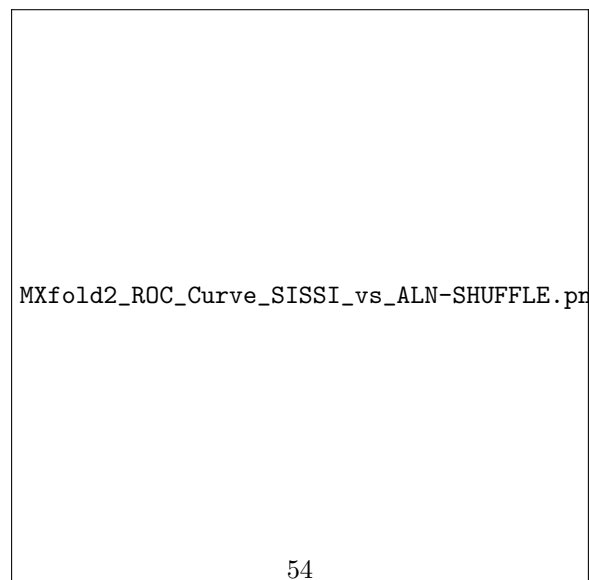
MXfold2\_ROC\_Curve\_SISSI\_vs\_MULTIPERM\_MONO.png

(c) SISSI vs MULTIPERM-mono



MXfold2\_ROC\_Curve\_SISSI\_vs\_MULTIPERM\_DI.png

(d) SISSI vs MULTIPERM-di



MXfold2\_ROC\_Curve\_SISSI\_vs\_ALN-SHUFFLE.png

54

(e) SISSI vs ALN-SHUFFLE

**Figure 24:** ROC-curves from MXfold2 prediction

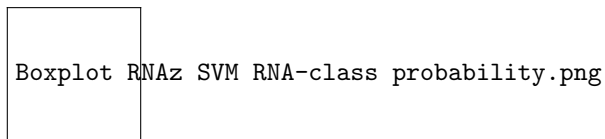
The evaluation of the ROC curves for the MXfold2 models shows that none of the tested models provides satisfactory results. This could indicate that MXfold2 has difficulties in dealing with the simulated data generated by SISSI, leading to inaccurate prediction of RNA structural integrity.

Overall, the Multiperm-mono, Multiperm-di and SISSIz-mono model performs best, as they have a AUC score of 0.85, followed by SISSIz-di with 0.81. The last one is as excepted the Aln-shuffle with 0.77

In view of these results, it would be of great interest to evaluate the models with real data in order to better assess their performance and applicability in practical scenarios. This could potentially lead to an improvement in the prediction accuracy and the general robustness of the models.

## 59 Results of RNAz Prediction

### 59.1 RNAz Boxplot



**Figure 25:** Boxplot MXfold2 Score

The boxplot illustrates the ability of different methods to destroy the RNA secondary structure. While the positive SSSI samples show very high RNAz probabilities as expected, the SSSIz-mono and SSSIz-di methods show significantly lower values and greater scatter. This indicates a successful destruction of the structure. In contrast, the RNAz values of the shuffling methods remain high, indicating insufficient structure destruction. Particularly striking is the Aln shuffling method, whose values are almost identical to the positive controls.





## 59.2 RNAz Histograms



**Figure 26:** Histograms from RNAz prediction

The histograms also confirm the results of the boxplot analysis: the SSSIz-mono and SSSIz-di methods lead to a significant shift in the probability distribution towards low RNAz values, indicating effective destruction of the RNA secondary structure. In contrast, the shuffling methods show hardly any change in the distribution compared to the positive control, indicating insufficient impairment of the structure.



59.3 RNAz Confusion Matrix



Figure 27: Confusion Matrix from RNAz prediction

An interesting result can be seen when analyzing the confusion matrices: The shuffle methods have lower overall values for false-positive (FP) and false-negative (FN) classifications. The best values are achieved by Multiperm-di with 70 FP and 717 FN, followed by Multiperm-mono with 83 FP and 706 FN. ALN-shuffle also achieves lower error rates than the SSSIz-based methods with 144 FP and 2052 FN.

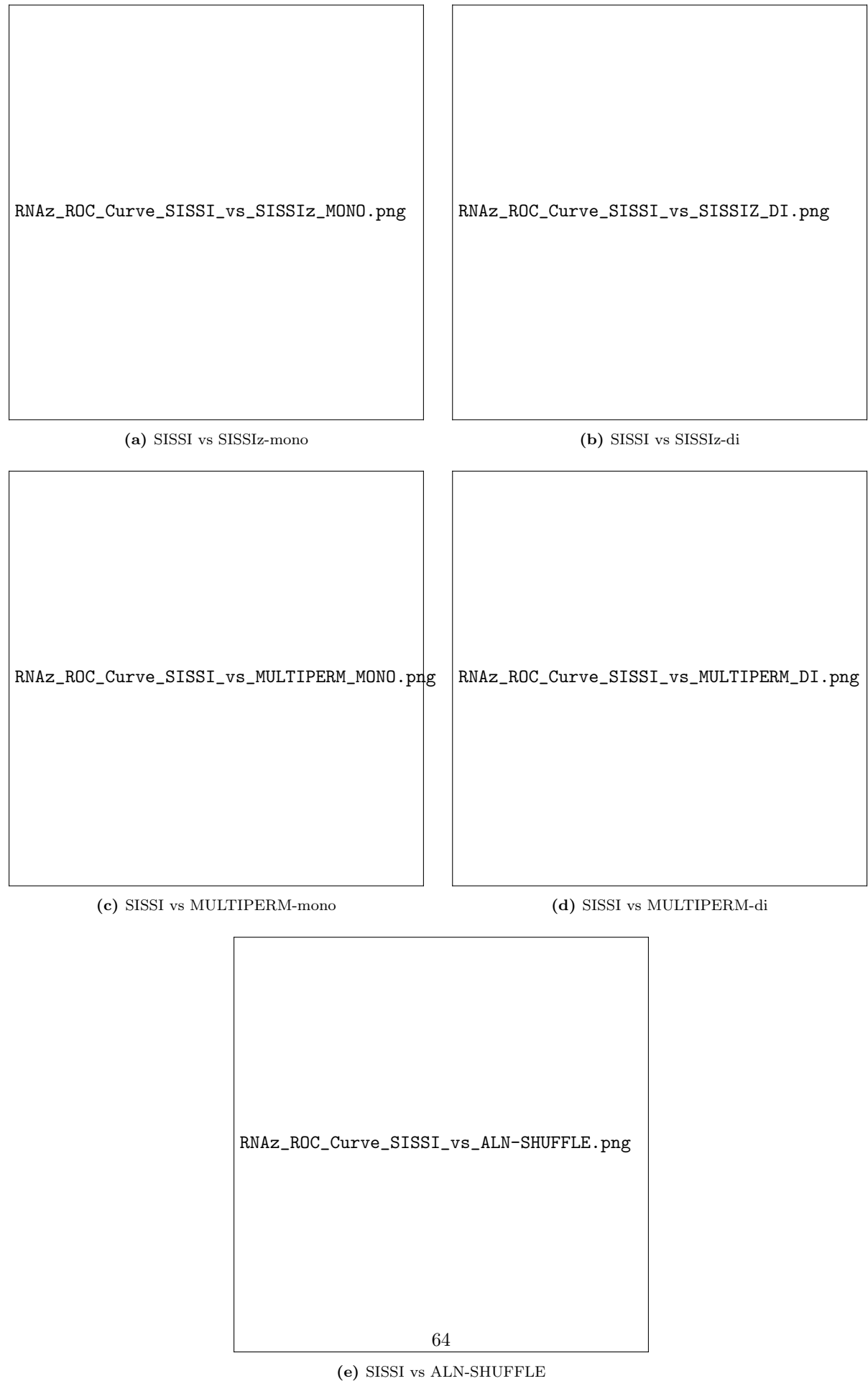
However, these seemingly positive values should be viewed critically. The reason for the low FP and FN values for the shuffle methods lies in the low discriminatory power of the classification: The p-values of the negative controls are very close to those of the positive sample (SSSI), so that the model can hardly distinguish between the classes.

In contrast, SSSIz-mono (2262 FP, 6885 FN) and SSSIz-di (3262 FP, 8145 FN) show a significantly higher misclassification rate. However, this results from an overall better separability.

In summary, it can be said that the SSSIz-mono and SSSIz-di methods cause a more effective destruction of the RNA secondary structure and thus enable an improved classification. Despite higher misclassifications numbers, they provide the most convincing results overall in terms of discriminatory power and classification performance.



## 59.4 RNAz ROC-Kurven



**Figure 28:** ROC-curves from RNAz prediction



The highest AUC values of 1.00 were achieved with the shuffle-based control methods (Multiperm-mono, Multiperm-di and ALN-shuffle). These values indicate an almost perfect differentiation of the classes in terms of score ranking.

However, this seemingly ideal classification must be viewed in a differentiated manner. Although a high AUC value means that positive instances tend to receive higher scores than negative ones, it says nothing about the absolute separability at a specific threshold value. The shuffle methods in particular show that, despite high AUC values, the actual class separation is less pronounced, which is reflected in the relatively low p-value differences between positive and negative examples.

In contrast, the SISSIZ-mono and SISSIZ-di models have slightly lower values with AUC values of 0.98, but show a clearer separation of the classes along the score spectrum. This stronger separation leads to improved classification performance in practical application, especially with fixed decision limits. This is demonstrated, among other things, by the clearly different score distributions and the results of the confusion matrices.

In summary, it can be stated that the AUC values of all models indicate a fundamentally high classification quality. However, despite slightly lower AUC values, the SISSIZ-mono and SISSIZ-di methods offer a more robust separation of classes that is easier to use in practice, making them particularly effective negative control methods in the given classification context.

## 60 SPOT-RNA2

For the prediction of RNA secondary structures, I downloaded the NCBI database `nt-prok. × .tar.gz` (26 compressed files in total). This should be sufficient to ensure broad coverage of prokaryotic sequences.

An initial problem was that SPOT-RNA2 did not recognize the database correctly. This could be fixed by creating a dummy file named `nt-prok` and adjusting the paths accordingly.

The script `run_spotrna2.sh` can be executed successfully. However, the runtime per file is around 3909 seconds. For a total data volume of 600,000 sequences, processing would theoretically take around 74 years.

The biggest bottleneck is the use of the `Infernal` tool, which can only be executed on the CPU. As the server used only has 48 CPU cores and these are fully utilized, it is not possible to parallelize the process.

### Possible solution approaches:

- (1) Skipping the `Infernal` step, but this can lead to a lower prediction accuracy.
- (2) Use of a smaller database, which could also negatively affect the prediction quality

If no solution is found for SPOT-RNA2 then I will look around for alternatives. (e.g.: UFold, E2EFold,...)

## 61 References

- [1] <https://doi.org/10.1038/s41467-021-21194-4>
- [2] <https://github.com/mxfold/mxfold2?tab=readme-ov-file>

Summary KW23 - KW24

## 62 Methods that i try to used

### 62.1 SPOT-RNA[1][2]

The installation of SPOT-RNA went smoothly and the initial results also looked promising. Unfortunately, the tool required around three days to predict 3000 samples when the available resources were fully utilized. Extrapolated to 600,000 samples, this would lead to an impractically long runtime. The use of SPOT-RNA was therefore discontinued and the focus shifted to alternative tools.

## 62.2 SPOT-RNA2[3][4]

SPOT-RNA2 also showed a very long prediction runtime. The computing time required per sample remained high despite the reduction in data volume, as SPOT-RNA and Infernal are very time-consuming. A significant acceleration could not be achieved.

## 62.3 UFold[5][6]

UFold was tested next. First issue was that the source code has some conflicts. After i solved the problems UFold could be used for the prediction. This tool does not support MSA files, only single-sequence fasta files. An attempt was therefore made to split the MSA files into individual sequences and generate suitable input files from them. This worked well and UFold provided fast predictions. One disadvantage, however, was that no real parallelization was possible, since UFold only ever works with a fixed input file ('input.txt'). Attempts to run multiple instances simultaneously failed. In addition, the output in dot-bracket notation was sometimes incorrect: brackets were missing or there were too many, which meant that no valid MFE calculation with RNAeval was possible.

## 62.4 E2Efold[7][8]

E2Efold follows a similar concept to UFold and is also its predecessor version. As it is no longer actively maintained, a functioning installation on the server was not possible.

## 62.5 AliNA[9][10]

AliNA is a relatively new tool for the prediction of RNA secondary structures. Installation and execution were straightforward, and the support of MSA data was a big advantage - no splitting of sequences was required. Unfortunately, AliNA only supports alignments up to a maximum length of 256 nucleotides. My alignment had a length of 401, and adjusting this limit in the code did not make sense as the model was only trained on sequences up to 256. An extension would have led to poor predictions. Without this limitation, AliNA would have been a suitable solution.

## 62.6 sincFold[11][12]

The installation and setup of sincFold also went smoothly. However, the tool delivered strange results when processing my MSA fasta files: Only one sequence was ever predicted - seemingly selected at random. The tool is therefore not suitable for MSA data. In addition, the prediction time was relatively long.

## 62.7 REDFold[13][14]

Finally, REDFold was tested. Although it only accepts single sequences as input, its speed and GPU support were convincing. The installation went smoothly and the results

in dot-bracket notation were formally correct. Based on these positive aspects, a full test run was started. The computing time for 234,000 fasta sequences was around 10 hours with sequential GPU use. Parallelization of the GPU processes could enable further optimizations here. However, it turned out that REDFold temporarily requires a lot of RAM - of the available 187.5 GB RAM, up to 141.5 GB was used at times. The content analysis of the results is still pending, but is being planned.

## 63 References

- [1] <https://doi.org/10.1038/s41467-019-13395-9>
- [2] <https://github.com/jaswindersingh2/SPOT-RNA>
- [3] <https://doi.org/10.1093/bioinformatics/btab165>
- [4] <https://github.com/jaswindersingh2/SPOT-RNA2>
- [5] <https://doi.org/10.1093/nar/gkab1074>
- [6] <https://github.com/uci-cbcl/UFold>
- [7] <https://doi.org/10.48550/arXiv.2002.05810>
- [8] <https://github.com/ml4bio/e2efold>
- [9] <https://doi.org/10.1002/minf.202300113>
- [10] <https://github.com/Arty40m/Alina>
- [11] <https://doi.org/10.1093/bib/bbae271>
- [12] <https://github.com/sinc-lab/sincFold>
- [13] <https://doi.org/10.1186/s12859-023-05238-8>
- [14] <https://github.com/aky3100/REDfold>

## 64 ECSFinder

Zusammenfassung 1 In dieser Studie wurde die Leistungsfähigkeit zweier etablierter Programme zur Vorhersage evolutionär konservierter RNA-Sekundärstrukturen – SISSIz und R-scape – systematisch verglichen. Beide Werkzeuge verfolgen unterschiedliche Ansätze: Während SISSIz auf thermodynamischer Modellierung und der Bewertung globaler Strukturstabilität basiert, konzentriert sich R-scape auf das Erkennen signifikanter Basenpaar-Covariation, also kompensatorischer Mutationen in evolutionär konservierten Helices. Um die Stärken und Schwächen beider Methoden zu bewerten, wurden zunächst mitochondriale Genome analysiert, da diese gut charakterisierte RNA-Strukturen enthalten.

Hierbei zeigte sich, dass SSISS eine hohe Sensitivität aufweist und breit gefächerte konservierte Strukturen erkennt – insbesondere in tRNAs –, jedoch auch eine erhöhte Rate an falsch-positiven Vorhersagen erzeugt. R-scape hingegen erzielte eine höhere Spezifität, übersah jedoch mitunter konservierte Strukturen, die keine ausgeprägten Covariationssignale zeigten. Die zusätzliche Nutzung des Helix-Scores in R-scape konnte diesen Nachteil teilweise ausgleichen.

Um die Vorhersageleistung beider Tools weiter zu testen, wurden experimentell validierte Rfam-RNA-Strukturen in simulierte, zufällig generierte Genomalignments eingebettet. Diese Benchmarking-Strategie ermöglichte eine kontrollierte Leistungsbewertung unter variierenden Sequenzähnlichkeiten (Multiple Sequence Identity, MPI). Dabei zeigte sich, dass SSISS insbesondere im Bereich mittlerer Ähnlichkeiten (60–80

ECSfinder kombiniert die Vorzüge von SSISS und R-scape mithilfe maschinellen Lernens. Hierzu wurden verschiedene Merkmale – darunter thermodynamische Stabilitätswerte und Z-Scores aus SSISS, signifikante Basenpaare und Helix-E-Werte aus R-scape sowie Pseudoenergie- und MFE-Werte aus RNALalifold – extrahiert und zur Merkmalsauswahl genutzt. Zwei Klassifizierungsmodelle, ein Generalisiertes Lineares Modell (GLM) und ein Random-Forest-Classifer (RF), wurden trainiert. In der unstrandspezifischen Analyse zeigte sich die Pseudoenergie von RNALalifold als besonders aussagekräftig, während die stranded Modelle vor allem vom Z-Score aus SSISS profitierten. Zusätzlich gewann bei stranded Vorhersagen auch die Standardabweichung des simulierten MFE an Bedeutung, was auf die Relevanz konsistenter Hintergrundmodellierung hinweist.

Der Random Forest erwies sich im Vergleich zum GLM als überlegen und übertraf in Benchmarks sowohl SSISS als auch R-scape hinsichtlich Genauigkeit, Sensitivität und Spezifität deutlich. Dies zeigte sich auch in der robusten Performance über verschiedene MPI-Bereiche hinweg sowie bei unterschiedlichen Anzahlen an verglichenen Spezies. Die Feature-Analyse verdeutlichte zudem, dass zwar Covariationsmerkmale besonders stark gewichtet wurden, thermodynamische Eigenschaften wie MFE oder die Standardabweichung des simulierten MFE jedoch ebenfalls essenzielle Prädiktoren darstellen – besonders bei stranded Vorhersagen.

Insgesamt zeigt ECSfinder, dass durch die Kombination komplementärer Vorhersagemethoden und den Einsatz interpretierbarer maschineller Lernverfahren eine präzisere Identifikation evolutionär konservierter RNA-Strukturen möglich ist. Die Ergebnisse legen nahe, dass weder rein thermodynamische noch ausschließlich covariationsbasierte Methoden ausreichen, um die volle Bandbreite funktioneller RNA-Strukturen zu erfassen. ECSfinder stellt damit ein robustes, skalierbares Werkzeug für vergleichende Genomik dar und kann künftig erweitert werden – etwa durch Einbezug experimenteller Daten wie SHAPE-MaP oder zusätzlicher Sequenzmerkmale. Damit leistet ECSfinder einen wichtigen Beitrag zur funktionellen Charakterisierung nicht-kodierender RNAs, insbesondere von regulatorisch aktiven lncRNAs, und eröffnet neue Möglichkeiten in der RNA-Biologie und genomweiten Strukturvorhersage.

**Zusammenfassung 2** In dieser Studie wurde die Leistungsfähigkeit zweier etablierter Werkzeuge zur Vorhersage evolutionär konservierter RNA-Sekundärstrukturen – SISIz und R-scape – untersucht und mit einem neu entwickelten Tool namens ECSfinder verglichen. Ziel war es, die Erkennungsgenauigkeit funktioneller RNA-Strukturen zu verbessern, indem die komplementären Stärken der bestehenden Methoden durch maschinelles Lernen kombiniert werden.

SISIz basiert auf thermodynamischer Modellierung und simuliert Zufallsverteilungen zur Ermittlung signifikanter Strukturmerkmale, insbesondere durch die Berechnung eines Z-Scores. Dieses Verfahren zeigte eine hohe Sensitivität, insbesondere bei mitochondrialen Genomen, indem es eine Vielzahl bekannter Strukturen wie tRNAs und rRNAs korrekt identifizierte. Allerdings ging diese Empfindlichkeit mit einer erhöhten Rate falsch-positiver Vorhersagen einher, insbesondere in Regionen, die möglicherweise funktionelle, aber nicht annotierte Strukturen enthalten. R-scape hingegen setzt auf die Erkennung signifikanter Basenpaar-Covariation und liefert damit eine höhere Spezifität. Dieses Werkzeug identifizierte konservierte Strukturen mit starker statistischer Evidenz, verpasste jedoch manchmal solche, bei denen die Covariationssignale weniger ausgeprägt waren – ein Nachteil, der durch den Einsatz des R-scape-internen Helix-Scores teilweise kompensiert werden konnte.

Um die individuellen Stärken beider Werkzeuge auszunutzen, wurde ECSfinder entwickelt. Dieses neue Verfahren integriert Merkmale aus SISIz, R-scape und RNALalifold in ein Machine-Learning-Modell, wobei sowohl ein generalisiertes lineares Modell (GLM) als auch ein Random Forest (RF) zum Einsatz kamen. Die Trainingsdaten basierten auf simulierten genomischen Alignments mit eingebetteten Rfam-Strukturen. Besonders effektiv war die Random-Forest-Methode, die alle anderen getesteten Verfahren hinsichtlich Klassifikationsgenauigkeit übertraf – sowohl in strand-unspezifischer als auch strand-spezifischer Vorhersage. Die Analyse der Merkmalsgewichte zeigte, dass der SISIz-Z-Score und die RNALalifold-Pseudoenergie zentrale Prädiktoren für korrekt konservierte Strukturen sind. Bei strand-spezifischen Vorhersagen war außerdem die Standardabweichung der Hintergrund-MFE aus SISIz ein besonders aussagekräftiges Merkmal.

Die Ergebnisse belegen, dass ECSfinder durch die Kombination thermodynamischer Stabilitätsmerkmale, statistischer Covariation und physikalischer RNA-Eigenschaften eine deutlich robustere und genauere Vorhersage evolutionär konservierter RNA-Strukturen ermöglicht. Während GLM eine gleichmäßigere Merkmalsverteilung aufweist, zeigt das Random-Forest-Modell insgesamt die höchste Klassifikationsleistung. Damit stellt ECSfinder eine effektive Erweiterung bisheriger Methoden dar und bietet neue Möglichkeiten zur funktionellen Annotation nicht-kodierender RNAs, insbesondere in groß angelegten vergleichenden Genomstudien. In Zukunft könnte ECSfinder um experimentelle Daten wie SHAPE-MaP erweitert werden, um die Genauigkeit weiter zu steigern und das Verständnis der Rolle konservierter RNA-Strukturen in Genregulation, Entwicklung und Krankheit zu vertiefen.

## 65 Svhip

### Zusammenfassung 1

Die Erkennung funktionaler RNA-Strukturen auf genomweiter Ebene ist ein zentrales Ziel der RNA-Bioinformatik, insbesondere im Kontext von nicht-kodierenden RNAs (ncRNAs), deren funktionale Bedeutung zunehmend erkannt wird. In ihrer Arbeit präsentieren Klapproth et al. ein flexibel einsetzbares, modular aufgebautes Machine-Learning-Framework, das es ermöglicht, maßgeschneiderte Klassifikatoren für die Detektion konservierter RNA-Strukturen aus genomischen Alignments zu trainieren. Dieses System wurde entwickelt, um die Einschränkungen klassischer Tools wie RNAz zu überwinden, die mit festen Modellen arbeiten und keine Anpassung an spezifische biologische Kontexte oder neue Trainingsdaten erlauben.

Das vorgestellte Framework — als Open-Source-Tool unter dem Namen Svhip verfügbar — ist vollständig in Python 3 implementiert und unterstützt Multiple-Sequence-Alignments im MAF-Format. Es erlaubt sowohl die Erkennung konservierter sekundärstrukturierter ncRNA-Regionen als auch die Identifikation protein-kodierender Sequenzen. Eine zentrale Stärke von Svhip liegt in der Möglichkeit, das zugrunde liegende Klassifikationsmodell mit benutzerdefinierten Daten neu zu trainieren. Anwender können ihre eigenen positiven und negativen Trainingsbeispiele bereitstellen, typischerweise als ClustalW-Alignments, um Modelle an bestimmte Arten, evolutionäre Distanzen oder Datenqualitäten anzupassen.

In der Methodik wurden verschiedene Features berücksichtigt, die zuvor bereits in Tools wie RNAz Anwendung fanden – z.B. Z-Score, Minimum Free Energy (MFE), Strukturkonservierung und Sequenzähnlichkeit. Neu ist jedoch die Möglichkeit, diese Features durch zusätzliche, frei wählbare Merkmale zu ergänzen, wie etwa Position-Specific Scoring Matrices oder kontextbezogene Sequenzdaten. Das Framework unterstützt unterschiedliche Machine-Learning-Verfahren, darunter generalisierte lineare Modelle (GLM) und Random Forests (RF). Die Modelle lassen sich exportieren und eigenständig oder integriert in andere Pipelines nutzen.

Ein besonderes Augenmerk lag auf der Vergleichbarkeit und Reproduzierbarkeit der Ergebnisse. Die Autoren betonen, dass Svhip bewusst nicht versucht, ein universell bestes Modell bereitzustellen, sondern stattdessen die Erstellung biologisch relevanter, anpassbarer Modelle fördert. Das ist vor allem für Anwendungen in Organismengruppen wichtig, deren genomische Eigenschaften (z.B. Basenkomposition oder Struktur motive) stark von Modellorganismen abweichen.

In Benchmarks auf dem Drosophila-Genom zeigte sich, dass Svhip-Modelle ähnlich leistungsfähig wie RNAz sind, mit vergleichbarer Sensitivität und Spezifität. Beim Test auf protein-kodierende Regionen erreichten die Modelle, abhängig von der Bewertungsmethode, einen Recall von bis zu 74 Prozent. Im nicht-kodierenden Bereich identifizierte Svhip eine Vielzahl bekannter konservierter RNA-Strukturen, wobei durch das flexible Modell-

training auch bislang nicht annotierte funktionale Regionen entdeckt wurden. Besonders hervorzuheben ist dabei die Fähigkeit, falsch-positive Vorhersagen zu reduzieren, indem auf strand-spezifische Merkmale und konsistente thermodynamische Profile geachtet wird. Zusätzlich diskutieren die Autoren die Kombination von Svhip mit bestehenden Programmen wie SSISS, RNAalifold und R-scape. In einem kombinierten Ansatz (wie z.B. in ECSfinder gezeigt), lässt sich die Vorhersagekraft weiter steigern, indem thermodynamische Stabilität (aus SSISS) mit Kovariationsanalysen (aus R-scape) und strukturorientierten Features (aus RNAalifold) kombiniert werden. Solche kombinierten Modelle, wie sie Svhip unterstützt, können je nach Anwendungsszenario individuell trainiert werden und übertreffen in den meisten Tests die isolierte Nutzung der Einzeltools.

Insgesamt bietet Svhip eine leistungsstarke und zukunftsweisende Plattform für die genomweite Identifikation funktionaler RNA-Strukturen, mit besonderem Fokus auf Erweiterbarkeit, Nachvollziehbarkeit und biologische Anpassbarkeit. Die Arbeit unterstreicht damit, wie maschinelles Lernen — wenn es gezielt und modular eingesetzt wird — helfen kann, die strukturelle RNA-Funktionalität im Genom besser zu verstehen und neue funktionale Elemente zu entdecken.