Multiscale Modeling of RNA Structures Using NMR Chemical Shifts

by

Kexin Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Chemistry and Scientific Computing)
in The University of Michigan
2020

Doctoral Committee:

Assistant Professor Aaron T. Frank, Chair
Assistant Professor Sarah Keane
Professor Ayyalusamy Ramamoorthy
Professor Kerby Shedden

Kexin Zhang

kexin@umich.edu

ORCID iD: 0000-0002-3273-2797

This thesis is dedicated to my family.

ACKNOWLEDGEMENTS

It has been a long journey but I can still remember the first day I came to Michigan. Now this journey is about to end and I would like to express my sincere gratitude to all the people that have helped me during these five years.

First I would like to thank my advisor, Dr. Aaron Frank, who has been an amazing advisor and mentor. When I first joined this lab, I have no background in computational structural biology. During these three years, Aaron has taught me so much and he has been so supportive and patient. I learnt a lot from him, not just skills and knowledge, but also how to be an independent and creative researcher.

I would like to thank all my committee members. I would like to thank Dr. Sarah Keane for all the help, support, and collaboration. I would also like to thank Dr. Ayyalusamy Ramamoorthy and Dr. Kerby Shedden for all the great feedbacks on my research. Your expertises have gave me many new perspectives. I would also like to thank our collaborator, Dr. Qi Zhang at UNC for the help on the CS-Fold project. I would also like to thank my previous advisor, Dr. Zhan Chen, for his understanding and for his help and mentor during the first two years of my graduate school.

Next I would like to thank everyone in the Frank lab. I want to thank Jingru Xie who gave me a lot of help when I first joined this lab and for our collaboration on the PyShifts project. I want to thank Ciara Witt for proofreading my papers and thesis and for all the fun she have brought to this group. I want to thank Dr. Indrajit Deb for all the help and advice he gave me on my research and career development. I want to thank Yichen Liu and Ziqiao Xu for the help you gave me during my final year. I

want to thank our alumnus, Dr. Sahil Chhabra, for his help when I first joined this lab.

I want to thank my friends here at Michigan and back home for all the fun and support you gave me during this long journey. I want to thank my parents for their unconditional love and support all my life. I want to thank my boyfriend, Yilai Li, for being my good friend and partner. Your trust, encouragement, and love is the reason I have come this far. Finally, I want to thank my two sweet cats, Cypress Zhang and Maple Zhang, for their company during all the hard time and for all the joy they have brought to my life.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Structure determination is an important step in understanding the mechanisms of functional non-coding ribonucleic acids (ncRNAs). Experimental observables in solution-state nuclear magnetic resonance (NMR) spectroscopy provide valuable information about the structural and dynamic properties of RNAs. In particular, NMR-derived chemical shifts are considered structural "fingerprints" of RNA conformational state(s). In my thesis, I have developed computational tools to model RNA structures (mainly secondary structures) using structural information extracted from NMR chemical shifts.

Inspired by methods that incorporate chemical-mapping data into RNA secondary structure prediction, I have developed a framework, CS-Fold, for using assigned chemical shift data to conditionally guide secondary structure folding algorithms. First, I developed neural network classifiers, CS2BPS (**C**hemical **S**hift to **B**ase **P**airing **S**tatus), that take assigned chemical shifts as input and output the predicted base pairing status of individual residues in an RNA. Then I used the base pairing status predictions as folding restraints to guide RNA secondary structure prediction. Extensive testing indicates that from assigned NMR chemical shifts, we could accurately predict the secondary structures of RNAs and map distinct conformational states of a single RNA.

Another way to utilize experimental data like NMR chemical shifts in structure modeling is probabilistic modeling, that is, using experimental data to recover native-like structure from a structural ensemble that contains a set of low energy structure models. I first developed a model, SS2CS (**S**econdary **S**tructure to **C**hemical **S**hift),

that takes secondary structure as input and predicts chemical shifts with high accuracies. Using Bayesian/maximum entropy (BME), I was able to reweight secondary structure models based on the agreement between the measured and reweighted ensemble-averaged chemical shifts. Results indicate that BME could identify the native or near-native structure from a set of low energy structure models as well as recover some of the non-canonical interactions in tertiary structures. We could also probe the conformational landscape by studying the weight pattern assigned by BME.

Finally, I explored RNA structural annotation using assigned NMR chemical shifts. Using multitask learning, eleven structural properties were annotated by classifying individual residues in terms of each structural property. The results indicate that our method, CS-Annotate, could predict the structural properties with reasonable accuracy. We believe that CS-Annotate could be used for assessing the quality of a structure model by comparing the structure derived structural properties with the CS-Annotate derived structural properties.

One major limitation of the tools developed is that they require assigned chemical shifts. And to assign chemical shifts, a secondary structure model is typically assumed. However, with the recent advances in singly labeled RNA synthesis, chemical shifts could be assigned without the assumption about the secondary structure. We envision that using the chemical shifts derived from singly labeled NMR experiments, CS-Fold could be used for modeling the secondary structure of RNA. We also believe that unassigned chemical shifts could be used for selecting structure models. Native-like structures could be recovered by comparing optimally assigned chemical shifts with computed chemical shifts (generated by SS2CS). Overall, the results presented in this thesis indicate we could extract crucial structural information of the residues in an RNA based on its NMR chemical shifts. Moreover, with the tools like CS-Fold, SS2CS, and CS-Annotate, we could accurately predict the secondary structure, model conformational landscape, and study structural properties of an RNA.

CHAPTER I

Introduction

1.1 Central dogma and non-coding RNAs

1.1.1 Overview

The central dogma of molecular biology[1] states that genetic information is stored in deoxyribonucleic acid (DNA) and passed to protein by ribonucleic acid (RNA). Protein then carries out the cellular functions encoded by genetic information from DNA. Thus, for a long time, RNA was considered to be the intermediate of genetic information. However, it was discovered that only 2% of the human genome is translated into proteins.[2] The remaining transcripts are thought to be functional non-coding RNAs (ncRNAs). An ncRNA is an RNA molecule that does not code for proteins. Examples of functional ncRNAs include transfer RNAs (tRNAs), ribosomal RNAs (rRNAs), small RNAs (such as microRNAs[3]), and long ncRNAs (lncRNAs)[4]).

Currently, for even the most thoroughly studied organisms, such as *Escherichia coli*, there are still many non-coding sequences waiting to be discovered. Intriguingly, in human genes, there are more lncRNAs than protein-coding RNAs, with comparable expression level.[5] By combining experimental detection methods (such as RNA sequencing[6]) and computational tools, new, functional ncRNAs are continually being identified and studied.

1.1.2 Examples of functional non-coding RNAs

Among these ncRNAs, the discovery of catalytic RNA in 1982, or **ribozyme**,[7] was especially surprising for the scientific community at that time because the prevailing scientific opinion was that proteins were the only catalytic molecules in cells. Though RNA base pairs are perfect for storing and carrying genetic information, RNAs have some structural features that appeared to preclude them from working as catalysts. For example, ribozymes, as catalytic RNAs, only have four unique nucleotides that can be used to facilitate reactions in their active sites.[8] By comparison, proteins can have twenty different amino acids. Also, RNAs lack positively charged functional groups, which can be used to neutralize negative charges of catalytic transitional states (at neutral pH)[9] or to stabilize a leaving group.

Another long-held assumption was that ribozymes were all metalloenzymes[8] that would require divalent metal ions (such as Mg^{2+}) as cofactors to function. However, this assumption is incorrect. Most self-cleaving ribozymes[10] do not require metal ions.[11] Their catalytic activity is attributed more directly to nucleotide bases. Hepatitis delta virus (HDV)-like, hammerhead, hairpin, *Neurospora* Varkud satellite (VS), glucosamine-6-phosphate synthase (glms), and twister ribozymes[12–17] are all self-cleaving ribozymes using a general acid-base catalysis model to function.

Another example of a class of functional RNAs is the **riboswitch**. Riboswitches are segments of messenger RNA (mRNA), located at the 5'-untranslated region (5'-UTR), that regulate gene expression by "switching" between different conformational states. The conformational change of a riboswitch can be triggered by specific cellular metabolites,[18] including single molecules (such as glycine[19]) and more complex compounds (such as vitamin B_{12}[20]). There are two structural domains in riboswitch: the aptamer domain and the expression platform (Figure 1.1). The aptamer domain is more structurally conserved while the expression platform is less conserved. The aptamer domain can sense and bind to specific ligands by forming a special binding

pocket. The ligand binding event triggers structural rearrangement, at the level of secondary structure and possibly tertiary structure, across the riboswitch that enables it to regulate downstream gene expression.

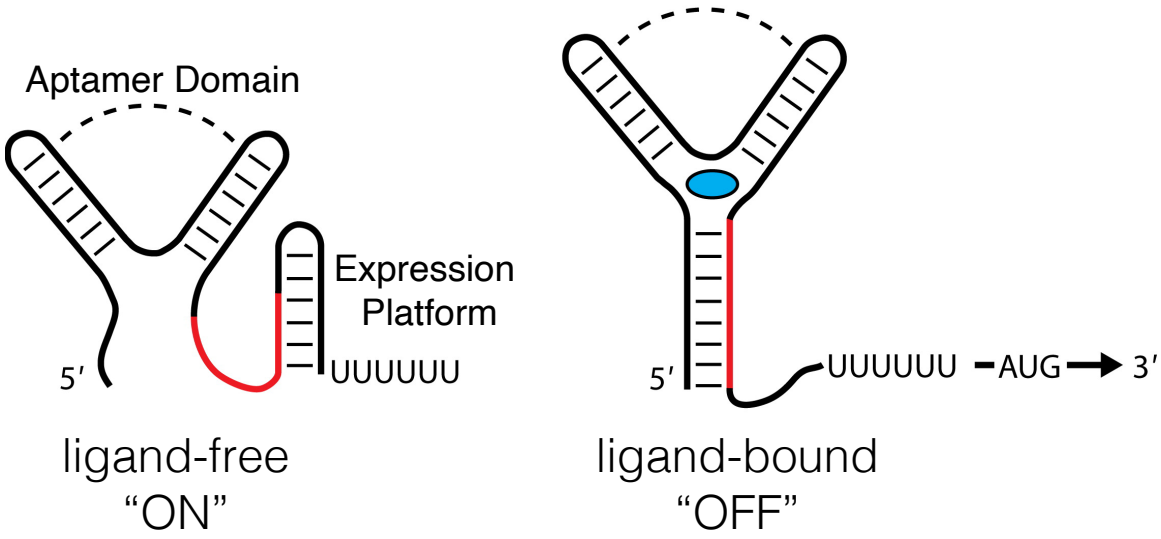

Figure 1.1: Transcriptional control of riboswitch. Adapted from reference.[21]

1.2 RNA structure determination

1.2.1 Overview of RNA structure

To understand the molecular mechanisms of functional ncRNAs, it is important to know their structures. For example, the function of ribozymes and riboswitches is dependent on their structures as structure dictates how they form catalytic domains and how they form “pockets” that accommodate their cognate ligands, respectively.

RNAs are polymers that consist of monomers called “nucleotides” (nts) or “residues”. A nucleotide contains a nucleobase, a ribose sugar ring, and a phosphate group. RNA can have two purine bases, adenine (**A**) and guanine (**G**), and two pyrimidine bases, uracil (**U**) and cytosine (**C**) (Figure 1.2).

There are three different levels of RNA structure:

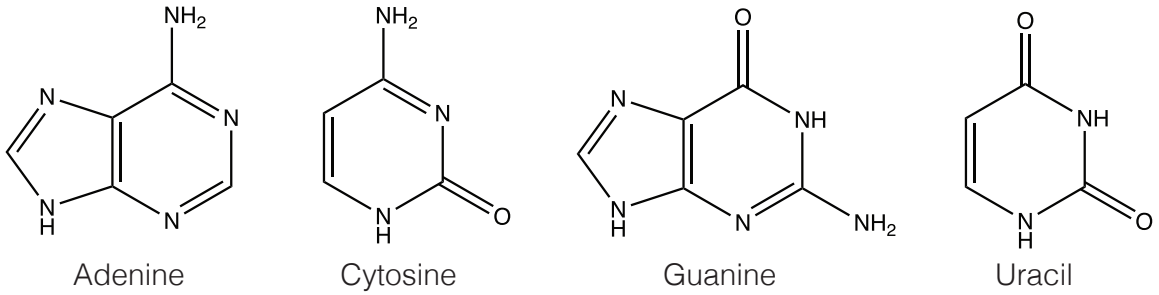

Figure 1.2: Four RNA bases.

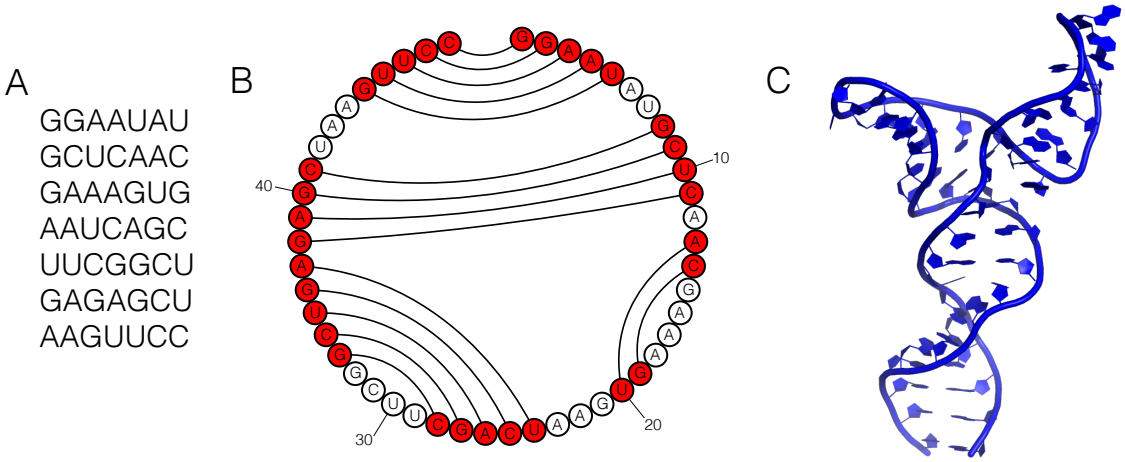

Figure 1.3: The three levels of structure for the group II intron Sc.ai5γ RNA (PDBID: 2LU0).

- the primary sequence or **1D** structure (Figure 1.3A), which is the string of nucleotides that make up the RNA molecule;
- the secondary structure or **2D** structure (Figure 1.3B), which is the connectivity between nucleotides: helical stem regions formed by base pairs and single-stranded regions (including hairpins, internal loops, and junctions).
- the tertiary structure or **3D** structure (Figure 1.3C), which is how an RNA chain folds in 3D space.

The function of an RNA depends largely on its tertiary structure.

1.2.2 Determining RNA structure using experimental techniques

To determine RNA structure experimentally, one accurate method is **X-ray crystallography**. The first crystal structure of RNA was published in 1947 for tRNA.[22] For a long time, this was the only available RNA tertiary structure.[23] Now, there are over 800 RNA-only structures solved by X-ray crystallography that have been deposited in the Protein Data Bank (`https://www.rcsb.org/`), with a median sequence length of 61 nts (Table 1.1). Despite efforts in biochemistry characterization that help grow better crystals, many X-ray structures still suffer from poor resolution (> 2.5Å)[24] since RNA molecules are relatively small, dynamic, and harder to crystallize compared to proteins.

One way to conquer the potential ambiguities in X-ray derived structures is to combine experimental data (electron density maps) with computational modeling. Chou *et* al.[24] incorporated electron density data into the energy function of *Rosetta*,[25–27] one of the most commonly used tools for RNA 3D modeling that is based on fragment assembly. This tool, referred to as ERRASER, was based on a three-step real-space refinement and was shown to improve structure resolution on a data set of 24 crystal structures of RNAs of different sizes.

Table 1.1: Size of RNA solved by different experimental methods

Residue count statistics	X-ray	Solution NMR	Other methods
mean	120	29	1040
min	6	8	17
25%	33	18	98
50%	61	24	244
75%	107	32	1533
max	2998	155	4446

RNA can adopt and interconvert between different conformational states to perform important biological functions. For example, riboswitch RNA regulates gene

expression by "switching" between ligand-bound state (*holo* state) and ligand-free state (*apo* state). To study the structure of an RNA to better understand its flexibility, another powerful and widely applied technique is **nuclear magnetic resonance (NMR) spectroscopy**.

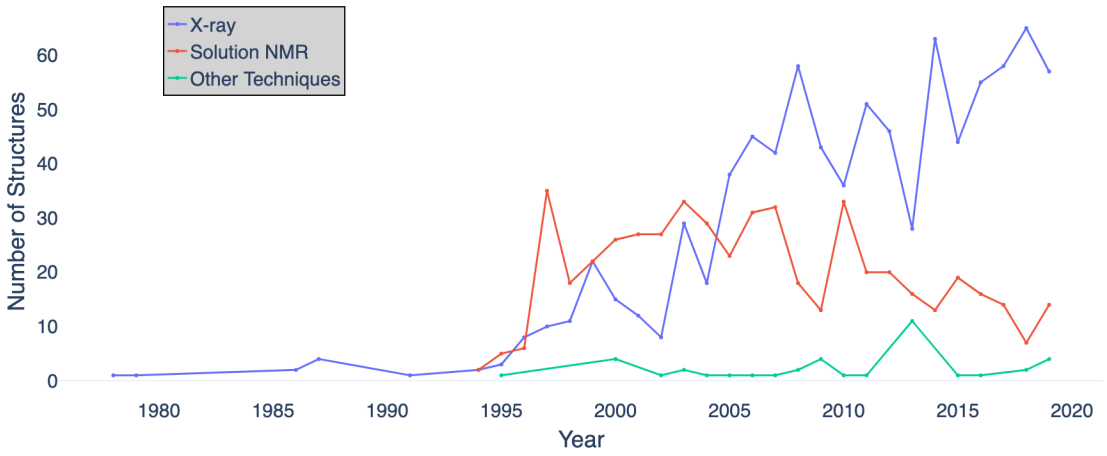

Figure 1.4: RNA structures deposited in the Protein Data Bank (PDB) per year before 2020.

As shown in Table 1.1, out of 1441 RNA structures deposited in the PDB as of March 2020, 883 structures were solved using X-ray crystallography, with sizes ranging from 6-2998 nts. In comparison, 519 structures were solved by solution NMR. Among these 519 structures, the largest RNA has 155 nts. Structures determined by solution NMR are often much smaller than structures solved by X-ray crystallography as it is more difficult to crystallize the smaller RNAs.

The reason that NMR is well-suited for studying the structures of these small and flexible RNAs is that NMR provides a set of experimental observables including chemical shifts and other structural restraints (such as nuclear Overhauser effect (NOE) distances and dihedral angles) which contain information about structural and dynamic properties of biomolecules. Among all the observables from NMR, **chemical shift** data are of the most significant value because they can provide information about conformational states that are accessible to a given RNA. Details of chemical

shifts and their use in guiding RNA structure modeling will be discussed in `Section 1.4`.

Although NMR provides valuable information about RNA conformational states, it suffers from limitations when studying large RNA molecules (>50 nts). Resonance assignment is complicated by severe spectral overlap and fast relaxation.[28] Fortunately, with recent developments such as selective labeling, advances in 2D spectra like NOESY, and utilization of experimental restraints such as residual dipolar couplings (RDCs), more precise and more accurate structures can be captured.[29–32]

1.2.3 Computational modeling

Solving RNA structures with NMR spectroscopy is often quite challenging, particularly for large RNAs.[33] Most of the RNA structures derived from NMR experiments that have been deposited in the PDB are small RNAs (with an average size ∼29 nts; Table 1.1). Besides, both X-ray crystallography and NMR spectroscopy experiments can be expensive, tedious, and time-consuming. Despite recent advances in other methods such as small-angle scattering (SAXS),[34] there is now much interest in developing and applying computational structure prediction methods (*in silico* methods), which can be used, in principle, to reduce the effort needed to "determine", or, more accurately, model the structure of RNAs.

It is a widely held view that RNA folding is hierarchical[35] in nature, meaning that secondary structure elements are formed first from which tertiary (3D) structure emerges. Thus, to mimic the hierarchical nature of RNA folding, many RNA modeling tools decompose the structure prediction into two discrete steps: first, the secondary structure is predicted from its sequence (i.e., its primary structure). Then, given the predicted secondary structure, tertiary models are built that are consistent with the predicted secondary structure. Recently, however, the hierarchical nature of RNA folding has been called into question. An NMR study[36] on the mechanism of the

fluoride riboswitch has provided evidence that even with identical tertiary structures, riboswitches may access conformational states with different secondary structures to perform transcription regulation functions. Despite the debate over whether RNA folding is concurrent or hierarchical, a well-predicted secondary structure is required for generating high resolution tertiary structure models.

1.2.3.1 Secondary structure prediction

RNA secondary structure describes the connectivity within an RNA sequence (Figure 1.3B): nucleotides can either form base pairs or be unpaired. Secondary structures can be evolutionarily conserved,[37] even more conserved than primary sequence, and are indicative of critical biological functions. Knowledge about these conserved regions helps RNA classification and phylogenetic study.[38]

When multiple RNA sequences are available, one common approach for secondary structure prediction is **comparative analysis**. The comparative approach scans sequences from the same family (homologous sequences) and identifies conserved secondary structures: these regions have the same connectivity but may contain different nucleotides. Nucleotides at covariant base pairs change together but ultimately maintain the same secondary structure. Comparative analysis has been successfully utilized to predict the secondary structure of large RNAs such as the 16s ribosomal RNA and 23s ribosomal RNA,[39] both with over a thousand nucleotides. There is also a keen interest in using multiple sequence alignment to study lncRNAs. lncRNAs may contain unidentified translated regions or transcriptional noise.[40] Thus, a fast quantitative tool is needed to locate possible conserved regions. Methods, such as R-scape,[37] perform covariation analysis and statistical tests to determine whether lncRNAs have functional, conserved structures.

Comparative analysis requires multiple sequences to make predictions, and the quality of the prediction is mainly dependent on existing structures. When a single

sequence is to be analyzed, the **thermodynamics approach** is used to predict its secondary structure. The principle of thermodynamics based methods is to identify the structure(s) with the lowest free energy. To do this, fragments of RNA sequences are evaluated in terms of their folding energy according to the nearest neighbor parameters which were determined using optical melting experiments.[41] The searching of the lowest energy structure is done through dynamic programming to save computation time. All possible base pairs and secondary structures are considered (implicitly) by combing the energy of shorter fragments.

However, free energy minimization sometimes does not provide the correct answer because: (1) energy is calculated based on a set of experimentally measured parameters which can be inaccurate; (2) RNA might not adopt the lowest energy structure. RNA may transition between different structures, sometimes higher energy structures, to function, especially when interacting with other molecules such as when binding with small ligands.

To improve prediction accuracy, experimental data can be added as restraints to guide the folding and free energy minimization. For example, the *RNAstructure* suite allows users to input chemical mapping data like SHAPE reactivities[42,43] to guide structure prediction.

Table 1.2: List of secondary structure prediction programs[44–48]

Tool	Description
MFold	free energy minimization, dynamic programming
RNAfold	statistical sampling using equilibrium probabilities
RNAstructure	free energy minimization, allows the use of experimental data (e.g. SHAPE reactivities)
RNAalifold	multiple sequence alignment
ContraFold	statistics based, conditional log-linear models
SPOT-RNA	deep learning, transfer learning

Recent advances in machine learning also inspired new prediction tools. For ex-

ample, SPOT-RNA[48] applied deep neural network models (that have been proven successful in predicting protein contact maps[49]) on a recently released large database of RNA sequences and their annotated secondary structures. It then applied the transfer learning techniques to improve structure prediction using a small set of high resolution structures.

1.2.3.2 Tertiary structure prediction

The biological function of RNAs largely depends on their complex tertiary (3D) structures. Given the challenges of solving structures experimentally, computational tools have been developed to model the 3D structures of RNA. Many of these methods were inspired by similar approaches developed for protein modeling.

Most methods for RNA 3D modeling can be grouped into two categories: **physics-based approaches** and **knowledge-based approaches**. Physics-based approaches use principles of physics and chemistry to explore free energy landscape and model biophysical events while knowledge-based approaches extract information from existing database such as sequences and known structures.

One example of physics-based modeling is molecular dynamics (MD) simulations. **MD simulations** are Molecular Mechanics (MM) method that is based on Newtonian mechanics. MD simulations calculate molecule's motion in a given time interval. The trajectories of particles and forces applied to particles are calculated by solving the Newton's equation. Due to its timescale, nanoseconds to microseconds, MD simulations are suitable for studying conformational changes such as ligand unbinding process,[50] and can be used to explore and construct the folding landscape of an RNA. However, MD simulations cannot deal with a biological process that involves chemical reactions. For example, the breaking of covalent bonds should be studied with Quantum Mechanics (QM) methods.

MD simulations have two major limitations: inaccurate force field and inefficient

sampling. The calculation of force and motion is dependent on the force field used. Two widely used force fields for RNAs are CHARMM and AMBER.[51,52] However, since MM makes some approximations compared to QM, the force field cannot depict all interactions involved. Advances[53,54] have been made to achieve better parametrization, but RNA force fields are still less accurate compared to protein force fields. The second limitation is the computational cost. Although MD simulations are less computationally expensive compared to QM methods, the timescale is still too short for many biological processes. For example, the timescale for the ligand unbinding process is significantly longer than regular MD simulations.[50] In addition, when the folding landscape is very "rugged", it is easy for the structure to be trapped in local minima for a long time. Recent advances, including a scaled-MD method,[50] have been made to accelerate such simulations.

In addition to physics-based approaches, knowledge-based approaches have been successful in protein prediction and have inspired new RNA prediction methods. One example is **Rosetta** and its analog in RNA prediction: fragment assembly of RNA (**FARNA**). FARNA is a *de novo* prediction method that does not rely on homologous structures, secondary structure predictions, or experimental data. It builds a structure library consisting of conformations of 3 nt fragments from the crystal structure of the large ribosomal subunit (PDBID: 1FFK) which has ∼2700 nts. It also uses a coarse-grained representation of RNA bases to speed up the simulation. The assembly of fragments is done through Monte Carlo simulation. This process starts from an extended chain and forms a native-like structure that is supposed to have the lowest free energy. The simulation is guided by an energy function that takes into account previously seen conformations in experimentally solved structure (including structural details like backbone preference and side chain preference). The authors, Das and Baker, later presented an optimized protocol, **FARNA/FARFAR**,[26] that was further optimized with a step of full-atom refinement by incorporating Rosetta

energy function. FARNA/FARFAR was validated to predict native structure with high resolution, including recovering noncanonical interactions.

Aside from MD simulations and fragment assembly, there are other programs available where different inputs (sequence or secondary structure), models (all-atom or coarse-grained) and simulation methods (MD or Monte Carlo) can be applied. Some commonly used programs are RNAComposer, NAST, SimRNA, and MC-sym.[55–58]

1.3 Functionally important RNA transient state

To carry out biological functions, some ncRNAs may sample different conformational states and fluctuate between a ground state (i.e., the lowest free energy state) and transient states (i.e., higher free energy states) contingent on environmental conditions. Transient states are local minima in the RNA free energy landscape.[59,60] The conformational transition between the ground state and transient states involves structural rearrangements such as pseudoknot (an RNA secondary structure in which half of one helical segment is intercalated between another helical segment) formation, base pair reshuffling, and base-flipping.[61] For example, while the hairpin ribozyme has a very simple structure compared to other ribozymes, it can undergo different docked states to carry out catalytic activities.[62] The fluoride riboswitch, on the other hand, can access an excited state where an important base pair is broken to trigger structural rearrangement and terminate downstream gene expression.[36]

The structures of these transient states provide significant information regarding RNA function. However, due to their low populations and short lifetimes,[61] these transient states are "invisible" to conventional experimental techniques, so the structural and dynamic properties of RNA transient states have yet to be thoroughly studied.

Increasingly, there is a keen interest in applying NMR spectroscopy to study the transient states of RNAs. Many features of NMR make it suitable for character-

izing less populated transient states:[63] it can be used to characterize structures of biomolecules at high resolution; it can provide different types of observables which contain structural and dynamic properties of different conformational states; it can be used to probe motions at different timescales ranging from picoseconds to seconds, or even longer, which is suitable for sparsely populated transient states. Although it is not currently possible to detect the NOE distances associated with these transient states, it is now possible to characterize the ^{1}H and ^{13}C chemical shift "signatures" of RNA transient states using techniques based on saturation transfer[36,64] and relaxation dispersion.[59,65]

1.4 NMR spectroscopy and its application to RNA structural and dynamics study

1.4.1 Basic theory of NMR spectroscopy

Spin, or spin angular momentum, is an intrinsic property of nuclei. The magnitude of spin angular momentum, L, is given by $L = \hbar\sqrt{S(S+1)}$ where S is the spin quantum number. S will be 0 if the nucleus has an even number of protons and an even number of neutrons. The magnetic dipole moment associated with the spin is $\mu = \gamma * \vec{S}$. Here, the constant γ is called the gyromagnetic ratio and is a characteristic that belongs to a specific nucleus.

When an external magnetic field (B_0) is applied, the nucleus' spin can either align parallel or anti-parallel to the external field. The parallel alignment is slightly more populated because of the lower energy. The population difference is characterized by the Boltzmann distribution: $N_{parallel}/N_{anti} = exp(\frac{\Delta E}{kT})$, where ΔE is the energy difference between two alignments. The energy difference between two states can be calculated with: $\Delta E = \gamma * \hbar * B_0$. Although parallel alignment is slightly more favorable in terms of energy and thus, causing a bulk magnetization (M_0) over all

identical nuclei, the energy difference between two states is still quite small. Therefore, to increase the signal to noise ratio, one of the instrument design options is to use stronger B_0 to increase the energy difference. The precession frequency, or *Larmor frequency* of the magnetization along B_0 is:

$$\omega_0 = 2\pi\nu = \gamma B_0, \tag{1.1}$$

where ν is the resonance frequency of the nucleus.

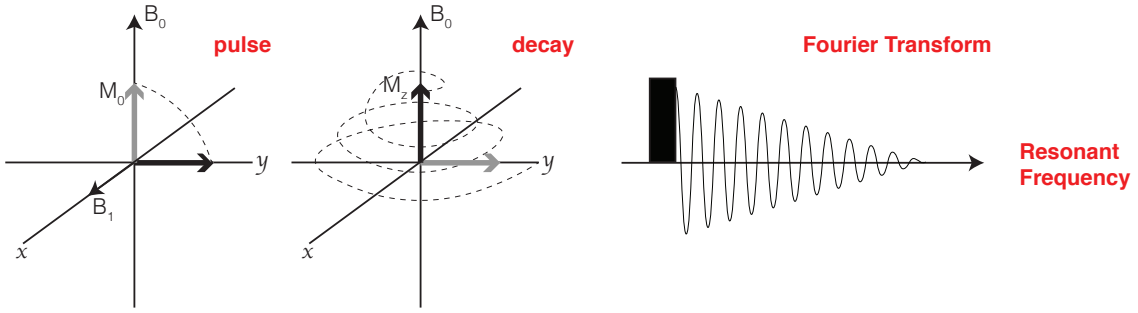

Figure 1.5: How NMR works (adapted from the web[66]).

A radio-frequency pulse (B_1) that is perpendicular to B_0 is then applied to the sample (Figure 1.5A), causing the equalization of populations between the two spin states. M_0, which was parallel to the external field, is now flipped to the transverse plane (Figure 1.5A). The non-equilibrium transversal magnetization (black arrow in Figure 1.5A) will relax back to the original equilibrium state (M_z) and the resultant decay in the transverse plane can be measured. The signal is then Fourier transformed into a characteristic resonant frequency that is unique for different types of nuclei.

1.4.2 Chemical shift

According to Eq. 1.1, the same nuclei should have the same Larmor frequency because γ is a characteristic that is dependent on the nucleus type. However, in the presence of an external magnetic field, the nucleus of an atom can either be

shielded or deshielded from the external field, depending on its chemical environment or electronic environment. We use the shielding constant σ to describe how much the resonance frequency is affected:

$$\nu = \frac{\gamma}{2\pi} B_0(1-\sigma). \tag{1.2}$$

To get a more direct picture of the physio-chemical environment around a nucleus, we define "**chemical shift**" as:

$$\delta = \frac{\nu - \nu_{ref}}{\nu_{ref}} 10^6, \tag{1.3}$$

with the unit parts per million, or ppm. ν and ν_{ref} are the resonance frequency of the nucleus and the reference nucleus.

The sensitivity of chemical shifts to the surrounding environment indicates that chemical shifts report on both the structural and dynamical properties of atoms, and as such, are considered "**structural fingerprints**".

1.4.3 Chemical shifts based modeling of RNA structure

Chemical shifts are the primary and most accurately measured NMR observables, and are, sometimes, the only observables that can be obtained from NMR spectroscopy. As alluded to above, they are also recognized as the structural fingerprints of biomolecules because they are sensitive to the environment (chemical and physical) surrounding a nucleus.

Chemical shift data have been used to guide and improve protein modeling, such as CS-ROSETTA,[67] but they are less explored in RNA structure modeling. Extracting structural information from chemical shift data is an urgent task because other structural restraints, such as NOE distances, which are the interproton distance, and RDCs, representing the relative orientation of nuclei, are more difficult to access and

interpret[68] for RNA due to its structural flexibility.

1.5 Machine learning

Chemical shifts are affected by multiple factors,[68] such as electronegativity, anisotropy, and hydrogen bonding. This complexity makes it challenging to develop an explicit function to describe the inherent structural information of chemical shifts. In order to extract useful patterns directly from available chemical shift data, we designed and developed novel computational tools using machine learning that can guide the structure prediction of RNA.

1.5.1 Overview of machine learning

Artificial intelligence (AI) has gained much attention in the last decade. It has changed the way we think and live: online advertising is optimized with recommendation systems; chatbots help customers navigate through websites; driving safety is improved with co-pilot technology, etc. AI has also changed the natural science community. With the help of machine learning algorithms like deep neural networks and the development of powerful supercomputers, scientists can rapidly parse through mountains of data to discern useful patterns or detect abnormal data points.

Machine learning (and deep learning) are central techniques of AI. Machine learning is about learning from data. Common types of machine learning are supervised learning, unsupervised learning, and reinforcement learning. In this thesis, the algorithms I applied all belong to **supervised learning**. In supervised learning, an algorithm learns to map input data X to output data y. Simply speaking, we have a set of *labeled* historical data, for example, experimental data like chemical shifts or other NMR restraints along with the NMR derived structures, and we build a predictive model based on these known data. In the previous example, chemical shifts, or other experimental observables, are variables or *features*, and NMR derived

structures are outcomes or *labels*. With the trained machine learning model, we are able to predict the label given a set of new features, that is, we can predict structure from NMR chemical shifts. In the past, the Frank lab has used machine learning tools to predict chemical shifts from the 3D coordinates of RNAs[69] and proteins,[70] as well as predict solvent accessible surface area from coarse-grained models of proteins.[71]

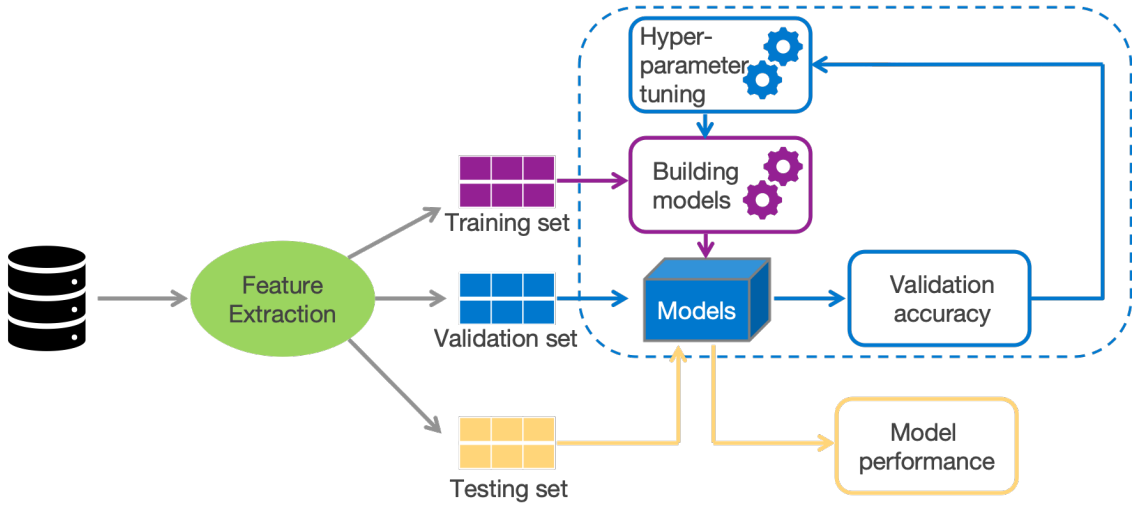

Figure 1.6: Workflow of developing a machine learning application (adapted from reference[72]).

Shown in Figure 1.6 is the general workflow of training a supervised learning model. Given a data set, the first step is to **extract features**. Data exploration and feature engineering are necessary to improve the quality of the model. Statistics, such as mean and variance, can be calculated to better understand features. Depending on algorithms used, imputation may be done before training to fill in missing data. In the case of image classification, data augmentation may also be necessary to ensure the model has enough training data points. Other feature engineering steps include dimension reduction, feature selection, and feature scaling.

Now that feature/label pairs have been constructed, the data set is split into three parts to **train a model**: *training set*, *validation set*, and *testing set*. An initial model will be built using the training set. The validation set is then fed into this model to evaluate the validation score. The training accuracy and validation

accuracy are used to guide hyperparameter tuning. Hyperparameters are parameters that determine model architecture and optimization that aren't learned in the model training process. They may affect how fast the training is done or whether the algorithm can converge or not, but, in principle, they should not affect how good the model is when predicting new data points. For example, in neural networks, the learning rate is a hyperparameter and should be defined before training the model.

After model training, the final step is to **evaluate** model performance. To do that, we feed the testing set into the optimized model and assess the model accuracy. The purpose of having this left-out testing set is to ensure the model performance is calculated on an unseen data set so that the performance measurement is unbiased.

One potential problem with machine learning models is *overfitting*, which occurs when the model does not generalize well so that it cannot predict the outcome of unseen data with similar accuracy compared to when it is applied on training data. This happens when the model is too complex. By reporting and comparing the training error and validation error, we can stop the training process at a certain point where the balance between training error and validation error is reached. Other techniques to avoid overfitting will be discussed in details in later chapters.

1.5.2 Artificial neural networks

Artificial neural networks, or ANNs, are inspired by biological nervous systems. With the recent advances in computing powers and algorithms design, neural networks have become deeper and deeper, making it possible to deal with complex data sets. Shown in Figure 1.7 is a very simple form of neural networks, with one input layer (*green*), one hidden layer (*blue*), and one output layer (*orange*). Input data (x_1, x_2, x_3, x_4) are fed into the hidden layer, each circle representing a *hidden neuron*. The input layer is connected with the hidden layer through a set of weights $(w_{11}^{(1)}, w_{12}^{(1)}, ...)$. The value of the hidden node can be calculated with $f(g(x))$

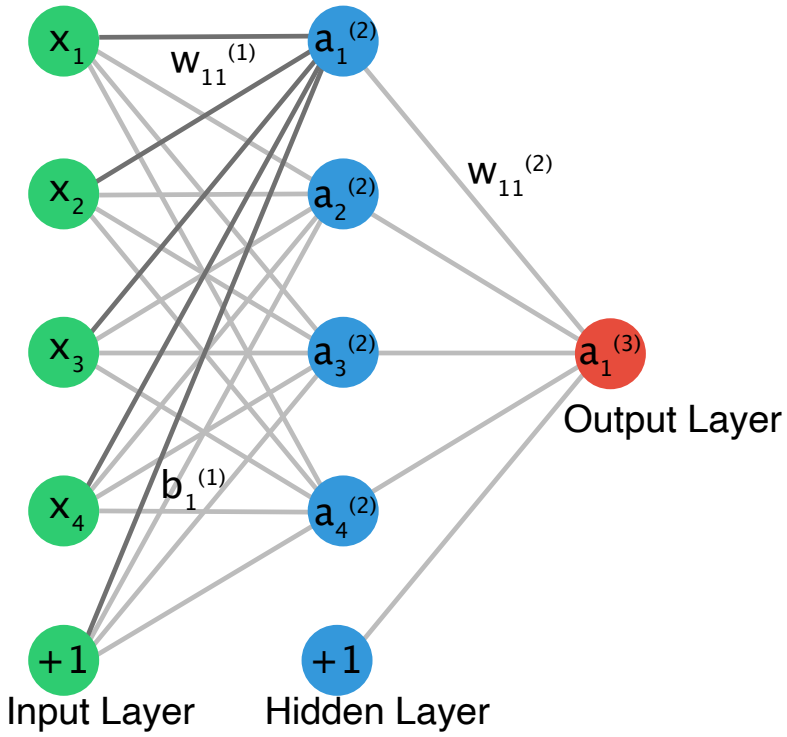

Figure 1.7: The architecture of neural network with one hidden layer. The ANN has one input layer (*green*), one hidden layer (*blue*), and one output layer (*orange*). The highlighted connections are weights associated with the first hidden neuron at Layer 2, or the first hidden layer.

where f is the activation function and $g(x)$ is a linear model of x, for example: $a_1^{(2)} = f(w_{11}^{(1)}x_1 + w_{12}^{(1)}x_2 + w_{13}^{(1)}x_3 + b_1^{(1)})$, where $w_{ij}^{(l)}$ represents the weight of the connection between the ith node on layer l and the jth node on layer $l+1$, and similarly, $b_j^{(l)}$ is the weight between the intercept node on layer l and the jth node on layer $l+1$. By calculating node values for all hidden layers and the output layer, the output value $h_{W,b}(x)$, or as in Figure 1.7, $a_1^{(3)}$, can also be calculated.

The next step is to train and minimize the *loss function* of the neural networks model:

$$\begin{aligned} J(W,b) &= \left[\frac{1}{m}\sum_{i=1}^{m} J(W,b;x^{(i)},y^{(i)})\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ij}^l)^2 \\ &= \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}(x^{(i)}) - y^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_l+1}(W_{ij}^l)^2, \end{aligned} \tag{1.4}$$

where $h_{W,b}(x^{(i)})$ is the predicted outcome of the ith data point, $y^{(i)}$ is the actual output of the ith data point, m is the number of training samples, λ is a penalty constant, n_l is the number of layers, s_l is the number of nodes at layer l, and $W_{ij}^{(l)}$ is the weight between the ith node at layer l and jth node at layer $l+1$.

The loss function has two terms: the mean squared error between predicted and actual output and the squared weights. The second term is a *regularization* term to ensure the model will not overfit.

To minimize this loss function (Eq. 1.4), gradient descent is used to update parameters, here, the weights:

$$\begin{aligned} W_{ij}^{(l)} &= W_{ij}^{(l)} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W,b) \\ b_j^{(l)} &= b_j^{(l)} - \alpha \frac{\partial}{\partial b_j^{(l)}} J(W,b), \end{aligned} \tag{1.5}$$

where α is the learning rate, one of the hyperparameters that can be tuned.

The calculation of the partial derivative of the loss function J with respect to each parameter $W_{ij}^{(1)}$ is done through **back propagation**. That is, the derivative for parameters at layer l is associated with the error at layer $l+1$. Thus, given the error at the final layer (since both predicted and actual output are already known), the derivative for previous layers can be calculated and weights can be updated via gradient descent:

$$\begin{aligned} \nabla_{W^{(l)}} J(W,b;x,y) &= \delta^{(l+1)} (a^{(l)})^T \\ \nabla_{b^{(l)}} J(W,b;x,y) &= \delta^{(l+1)}. \end{aligned} \tag{1.6}$$

Here $\delta^{(l+1)}$ is the error at layer $l+1$. Through iterations of forward and backward propagation, the weights of the neural networks can be optimized.

1.5.3 Random forest

Random forest[73] is an advanced predictive modeling technique based on decision trees.[74] Decision trees are tree-shaped graphs made of interconnected nodes that can be used to determine the probable outcome based on the data used to build it. At every node of a decision tree (except the terminal node, which is called a *leaf*), data points are split into branches, based on some features. This partitioning is done recursively until a stopping condition is met. A final decision can then be made based on the estimation of the outcome in each terminal node. The splitting is optimized through learning to minimize the training error (for regression) or minimize impurity (such as *Gini index*, for classification).

Decision trees are useful due to their simplicity, low bias, and high interpretability. However, they usually exhibit a high variance (overfitting issue), which means the model can change significantly if new training data are used to build the model, and are not robust. To overcome this limitation, random forest uses an approach called **bagging** where a "forest" of de-correlated decision trees are built instead of a single decision tree. Each tree in the forest will learn the model based on a subset of samples (**bootstrap** samples), and each tree will contribute to the final result. Random forest is, therefore, an example of **ensemble learning** method.

1.6 Thesis outline

Determining the structure of biomolecules is an important first step in understanding how they execute specific cellular functions. Recent advances in NMR, such as CEST NMR spectroscopy and NMR $R_{1\rho}$ relaxation dispersion spectroscopy,[64] make it possible to access chemical shift signatures associated with RNA transient states. In contrast, NOE-derived distances and other NMR observables, which are conventionally used to determine RNA structures, remain inaccessible for these transient states.

The goal of my thesis was to develop and apply computational tools for accurately modeling the structures (secondary structures in particular) of RNA conformational states, including sparsely populated transient states, based on their chemical shift signatures.

Inspired by methods that incorporate chemical mapping data as restraints in RNA secondary structure prediction, I developed a CS-Fold framework for conditional prediction of RNA secondary structures with NMR chemical shifts. First, I developed ANN-based classifiers that predict the base pairing status of individual residues in an RNA based on their assigned chemical shifts. Then I used these predictions as restraints to guide secondary structure folding of RNA. Extensive testing indicated that, from assigned NMR chemical shifts, we could accurately predict the secondary structures of RNAs and map distinct conformational states of a single RNA. The study on conditional prediction of RNA secondary structure is presented in `Chapter II`.

I then explored the probabilistic modeling of RNA secondary structures using chemical shifts in `Chapter III`. Given a simulated ensemble of structure models, I first developed a method that can predict chemical shifts from given secondary structures of RNA. Using Bayesian/maximum entropy (BME), I was able to reweight secondary structure models based on the agreement between the measured and predicted chemical shifts. The results indicated that using BME and predicted chemical shifts, we could recover native-like structures from a set of low energy structure models.

In `Chapter IV`, I further explored whether NMR chemical shifts can be used to annotate other structural features of RNAs. Structural features such as base pair, stacking interaction, *syn* or *anti* conformation, solvent accessibility, and sugar puckering modes are important for understanding RNA structure–function relationship. In this chapter, I applied multi-task learning with neural network classifiers to extract such structural information from NMR chemical shifts.

Finally, in the `Appendix`, a PyMOL plugin (*PyShifts*) that was implemented by my colleague Jingru Xie and me, is presented. PyShifts is designed for visualizing and analyzing structure ensembles of biomolecules. NMR-derived chemical shifts provide valuable information about the conformational state(s) accessible to a given biomolecule. Our examples show that PyShifts could be used to predict chemical shifts from 3D coordinates, visually detect referencing errors, identify the "best" structure model, and cluster structure ensembles into different conformational states.

1.7 References

(1) Crick, F. *Nature* **1970**, *227*, 561–563.

(2) Consortium, I. H. G. S. et al. *Nature* **2004**, *431*, 931.

(3) Bartel, D. P. *Cell* **2004**, *116*, 281–297.

(4) Perkel, J. M. Visiting "noncodarnia"., 2013.

(5) Djebali, S.; Davis, C. A.; Merkel, A.; Dobin, A.; Lassmann, T.; Mortazavi, A.; Tanzer, A.; Lagarde, J.; Lin, W.; Schlesinger, F., et al. *Nature* **2012**, *489*, 101–108.

(6) Ozsolak, F.; Milos, P. M. *Nat. Rev. Genet.* **2011**, *12*, 87–98.

(7) Kruger, K.; Grabowski, P. J.; Zaug, A. J.; Sands, J.; Gottschling, D. E.; Cech, T. R. *Cell* **1982**, *31*, 147–157.

(8) Fedor, M. J.; Williamson, J. R. *Nat. Rev. Mol. Cell Biol.* **2005**, *6*, 399–412.

(9) Pyle, A. M. *Science* **1993**, *261*, 709–714.

(10) Jimenez, R. M.; Polanco, J. A.; Lupták, A. *Trends Biochem. Sci.* **2015**, *40*, 648–661.

(11) Bevilacqua, P. C.; Yajima, R. *Curr. Opin. Chem. Biol.* **2006**, *10*, 455–464.

(12) Nakano, S.-i.; Chadalavada, D. M.; Bevilacqua, P. C. *Science* **2000**, *287*, 1493–1497.

(13) Han, J.; Burke, J. M. *Biochemistry* **2005**, *44*, 7864–7870.

(14) Bevilacqua, P. C. *Biochemistry* **2003**, *42*, 2259–2265.

(15) Wilson, T. J.; Li, N.-S.; Lu, J.; Frederiksen, J. K.; Piccirilli, J. A.; Lilley, D. M. *Proc. Natl. Acad. Sci. U. S. A.* **2010**, *107*, 11751–11756.

(16) Cochrane, J. C.; Lipchock, S. V.; Strobel, S. A. *Chem. Biol. (Oxford, U. K.)* **2007**, *14*, 97–105.

(17) Eiler, D.; Wang, J.; Steitz, T. A. *Proc. Natl. Acad. Sci. U. S. A.* **2014**, *111*, 13028–13033.

(18) Nudler, E.; Mironov, A. S. *Trends Biochem. Sci.* **2004**, *29*, 11–17.

(19) Mandal, M.; Lee, M.; Barrick, J. E.; Weinberg, Z.; Emilsson, G. M.; Ruzzo, W. L.; Breaker, R. R. *Science* **2004**, *306*, 275–279.

(20) Winkler, W. C. *Curr. Opin. Chem. Biol.* **2005**, *9*, 594–602.

(21) Jones, C. P.; Ferré-D'Amaré, A. R. *Annual review of biophysics* **2017**, *46*, 455–481.

(22) Kim, S.; Suddath, F.; Quigley, G.; McPherson, A; Sussman, J.; Wang, A.; Seeman, N.; Rich, A. *Science* **1974**, *185*, 435–440.

(23) Turner, D. H.; Sugimoto, N.; Freier, S. M. *Annu. Rev. Biophys. Biophys. Chem.* **1988**, *17*, 167–192.

(24) Chou, F.-C.; Sripakdeevong, P.; Dibrov, S. M.; Hermann, T.; Das, R. *Nat. Methods* **2013**, *10*, 74.

(25) Das, R.; Baker, D. *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104*, 14664–14669.

(26) Das, R.; Karanicolas, J.; Baker, D. *Nat. Methods* **2010**, *7*, 291.

(27) Sripakdeevong, P.; Kladwang, W.; Das, R. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 20573–20578.

(28) Barnwal, R. P.; Yang, F.; Varani, G. *Arch. Biochem. Biophys.* **2017**, *628*, 42–56.

(29) Tjandra, N.; Bax, A. *Science* **1997**, *278*, 1111–1114.

(30) Liu, Y.; Holmstrom, E.; Zhang, J.; Yu, P.; Wang, J.; Dyba, M. A.; Chen, D.; Ying, J.; Lockett, S.; Nesbitt, D. J., et al. *Nature* **2015**, *522*, 368–372.

(31) Keane, S. C.; Heng, X.; Lu, K.; Kharytonchyk, S.; Ramakrishnan, V.; Carter, G.; Barton, S.; Hosic, A.; Florwick, A.; Santos, J., et al. *Science* **2015**, *348*, 917–921.

(32) Getz, M.; Sun, X.; Casiano-Negroni, A.; Zhang, Q.; Al-Hashimi, H. M. *Biopolymers: Original Research on Biomolecules* **2007**, *86*, 384–402.

(33) Tolbert, B. S.; Miyazaki, Y.; Barton, S.; Kinde, B.; Starck, P.; Singh, R.; Bax, A.; Case, D. A.; Summers, M. F. *J. Biomol. NMR* **2010**, *47*, 205–219.

(34) Wang, J.; Zuo, X.; Yu, P.; Xu, H.; Starich, M. R.; Tiede, D. M.; Shapiro, B. A.; Schwieters, C. D.; Wang, Y.-X. *J. Mol. Biol.* **2009**, *393*, 717–734.

(35) Tinoco Jr, I.; Bustamante, C. *J. Mol. Biol.* **1999**, *293*, 271–281.

(36) Zhao, B.; Guffy, S. L.; Williams, B.; Zhang, Q. *Nat. Chem. Biol.* **2017**, *13*, 968.

(37) Rivas, E.; Clements, J.; Eddy, S. R. *Nat. Methods* **2017**, *14*, 45.

(38) Tahi, F.; Boucheham, A., et al. In *Promoter Associated RNA*; Springer: 2017, pp 145–168.

(39) Gutell, R. R.; Larsen, N.; Woese, C. R. *Microbiol. Mol. Biol. Rev.* **1994**, *58*, 10–26.

(40) Ji, Z.; Song, R.; Regev, A.; Struhl, K. *elife* **2015**, *4*, e08890.

(41) Mathews, D. H.; Turner, D. H. *Biochemistry* **2002**, *41*, 869–880.

(42) Deigan, K. E.; Li, T. W.; Mathews, D. H.; Weeks, K. M. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 97–102.

(43) Low, J. T.; Weeks, K. M. *Methods* **2010**, *52*, 150–158.

(44) Zuker, M. *Nucleic acids research* **2003**, *31*, 3406–3415.

(45) Denman, R. B. *Biotechniques* **1993**, *15*, 1090–1095.

(46) Bernhart, S. H.; Hofacker, I. L.; Will, S.; Gruber, A. R.; Stadler, P. F. *BMC bioinformatics* **2008**, *9*, 474.

(47) Do, C. B.; Woods, D. A.; Batzoglou, S. *Bioinformatics* **2006**, *22*, e90–e98.

(48) Singh, J.; Hanson, J.; Paliwal, K.; Zhou, Y. *Nat. Commun.* **2019**, *10*, 1–13.

(49) Hanson, J.; Paliwal, K.; Litfin, T.; Yang, Y.; Zhou, Y. *Bioinformatics* **2018**, *34*, 4039–4045.

(50) Deb, I.; Frank, A. T. *J. Chem. Theory Comput.* **2019**, *15*, 5817–5828.

(51) Brooks III, B.; Mackerell, C.; Nilsson, A.; Petrella, L; Roux, R.; Won, B; Archontis, Y; Bartels, G; Boresch, C; Caflisch, S, et al. *J. Comput. Chem* **2009**, *30*, 1545–1614.

(52) Cheatham III, T. E.; Case, D. A. *Biopolymers* **2013**, *99*, 969–977.

(53) Yildirim, I.; Stern, H. A.; Kennedy, S. D.; Tubbs, J. D.; Turner, D. H. *Journal of chemical theory and computation* **2010**, *6*, 1520–1531.

(54) Bergonzo, C.; Cheatham III, T. E. *Journal of chemical theory and computation* **2015**, *11*, 3969–3972.

(55) Popenda, M.; Szachniuk, M.; Antczak, M.; Purzycka, K. J.; Lukasiak, P.; Bartol, N.; Blazewicz, J.; Adamiak, R. W. *Nucleic Acids Res.* **2012**, *40*, e112–e112.

(56) Jonikas, M. A.; Radmer, R. J.; Laederach, A.; Das, R.; Pearlman, S.; Herschlag, D.; Altman, R. B. *RNA* **2009**, *15*, 189–199.

(57) Boniecki, M. J.; Lach, G.; Dawson, W. K.; Tomala, K.; Lukasz, P.; Soltysinski, T.; Rother, K. M.; Bujnicki, J. M. *Nucleic Acids Res.* **2016**, *44*, e63–e63.

(58) Parisien, M.; Major, F. *Nature* **2008**, *452*, 51–55.

(59) Dethoff, E. A.; Petzold, K.; Chugh, J.; Casiano-Negroni, A.; Al-Hashimi, H. M. *Nature* **2012**, *491*, 724–728.

(60) Mustoe, A. M.; Brooks, C. L.; Al-Hashimi, H. M. *Annu. Rev. Biochem.* **2014**, *83*, 441–466.

(61) Zhao, B.; Zhang, Q. *Curr. Opin. Struct. Biol.* **2015**, *30*, 134–146.

(62) Zhuang, X.; Kim, H.; Pereira, M. J.; Babcock, H. P.; Walter, N. G.; Chu, S. *Science* **2002**, *296*, 1473–1476.

(63) Bothe, J. R.; Nikolova, E. N.; Eichhorn, C. D.; Chugh, J.; Hansen, A. L.; Al-Hashimi, H. M. *Nat. Methods* **2011**, *8*, 919.

(64) Zhao, B.; Hansen, A. L.; Zhang, Q. *J. Am. Chem. Soc.* **2014**, *136*, 20–23.

(65) Blad, H.; Reiter, N. J.; Abildgaard, F.; Markley, J. L.; Butcher, S. E. *J. Mol. Biol.* **2005**, *353*, 540–555.

(66) .

(67) Shen, Y.; Lange, O.; Delaglio, F.; Rossi, P.; Aramini, J. M.; Liu, G.; Eletsky, A.; Wu, Y.; Singarapu, K. K.; Lemak, A., et al. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 4685–4690.

(68) Cavalli, A.; Salvatella, X.; Dobson, C. M.; Vendruscolo, M. *Proc. Natl. Acad. Sci. U. S. A.* **2007**, *104*, 9615–9620.

(69) Frank, A. T.; Law, S. M.; Brooks III, C. L. *J. Phys. Chem. B* **2014**, *118*, 12168–12175.

(70) Frank, A. T.; Law, S. M.; Ahlstrom, L. S.; Brooks III, C. L. *J. Chem. Theory Comput.* **2015**, *11*, 325–331.

(71) Wei, S.; Brooks III, C. L.; Frank, A. T. *J. Comput. Chem.* **2017**, *38*, 1270–1274.

(72) Karim, M. R., *Scala Machine Learning Projects: Build real-world machine learning and deep learning projects with Scala*; Packt Publishing Ltd: 2018.

(73) Breiman, L. *Machine learning* **1996**, *24*, 123–140.

(74) Quinlan, J *Machine Learning*, *1*.

CHAPTER II

Conditional Prediction of RNA Secondary Structures Using NMR Chemical Shifts

The contents of this chapter were published in the following reference:

Kexin Zhang, and Aaron T. Frank. "Conditional Prediction of Ribonucleic Acid Secondary Structure Using Chemical Shifts." *The Journal of Physical Chemistry B 124.3* (**2019**): 470-478.

Inspired by methods that utilize chemical-mapping data to guide secondary structure prediction, we sought to develop a framework for using assigned chemical shift data to guide RNA secondary structure prediction. We first used machine learning to develop classifiers which predict the base pairing status of individual residues in an RNA based on their assigned chemical shifts. Then, we used these base pairing status predictions as restraints to guide RNA folding algorithms. Our results showed that we could recover the correct secondary fold of most of the 108 RNAs in our data set with remarkable accuracy. Finally, we tested whether we could use the base pairing status predictions that we obtained from assigned chemical shift data to conditionally predict the secondary structure of RNA. To achieve this, we attempted to model two distinct conformational states of the microRNA-20b (miR-

20b) and the fluoride riboswitch using assigned chemical shifts that were available for both conformational states of each of these test RNAs. For both test cases, we found that by using the base pairing status predictions that we obtained from assigned chemical shift data as folding restraints, we could generate structures that closely resembled the known structure of the two distinct states. A command-line tool for **C**hemical **S**hifts to **B**ase-**P**airing **S**tatus (CS2BPS) predictions in RNA has been incorporated into our CS2Structure Git repository and can be accessed via: `https://github.com/atfrank/CS2Structure`.

2.1 Introduction

Like proteins, ribonucleic acids (or RNAs), play critical functional roles within cells, and like proteins, RNA function is determined by its structure.[1–3] RNAs, however, do not necessarily adopt a single structure. Instead, RNAs can adopt and interconvert between distinct conformational states that are kinetically linked to form a complex network of accessible states. Such networks enable RNAs to, for example, function as regulatory switches by responding to environmental stimuli such as changes in temperature or changes in ligand or metabolite concentration.[4–8] Mapping the conformational landscape of RNAs,[4,9] which consists of the set of conformational states accessible to that RNA, is crucial in unraveling the complex relationships between their sequence, their structure, and their function.

Determining the secondary structure of an RNA, under a specific set of physiochemical conditions, is a crucial first step in uncovering links between its sequence, its structure, and its function.[10] *In silico* methods can be used to predict the secondary structure of an RNA from sequence by identifying the secondary structure that minimizes its folding free energy.[11–13] However, the structure that an RNA adopts depends not only on its sequence but also on the physiochemical environment in which the RNA "resides". Urgently needed are methods that can predict the RNA structure from sequence, *conditioned* on the physiochemical environment of an RNA.

Herein, we implemented and tested a framework for *conditionally* predicting the secondary structure of RNAs based on *assigned* chemical shift data.[14,15] Specifically, we trained a set of machine learning classifiers that take as input the *assigned* chemical shifts of individual residues in an RNA and then predict the base pairing status of each residue. We then used these base pairing predictions as restraints to guide the folding of the target RNA. We discovered that the secondary structures generated using our chemical shift derived base pairing status predictions as restraints were more consistent with NMR-derived secondary structure models than the models generated

without these restraints. Moreover, for the microRNA-20b (miR-20b) and the fluoride riboswitch, we were able to accurately predict their secondary structure *conditioned* on the available chemical shift data for two distinct conformational states.

Collectively, our results demonstrate that the information content in *assigned* chemical shift data can be leveraged to *conditionally* predict the secondary structure of an RNA by combining machine learning tools and existing structure prediction algorithms. With access to the *assigned* chemical shift fingerprints of individual conformational states of an RNA, the hybrid modeling approach described in this study could be used to generate a hypothetical map of its conformational landscape, which will be particularly powerful when some of these states correspond to difficult-to-characterize transient states.[16,17]

2.2 Methods

2.2.1 Data preparation

2.2.1.1 Structure and chemical shift data set

For 115 RNAs, atomic NMR structures and NMR chemical shifts were downloaded from the Protein Data Bank (PDB: `http://www.pdb.org`) and the Biological Magnetic Resonance Data Bank (BMRB: `http://www.bmrb.wisc.edu/`), respectively. Next, for each RNA, LARMOR^{D}[18] was used to predict chemical shifts using the coordinates of the first model in the NMR bundle. Because ^{13}C chemical shifts frequently contain systematic referencing errors,[19] a structure-based approach was used to identify systematic referencing errors and (if necessary) correct ^{13}C data for each RNA. Briefly, a Bayesian inference approach was used to identify systematic offsets in the difference between experimental chemical shifts and chemical shifts computed using LARMOR^{D}. For each type of non-exchangeable ^{13}C nuclei (namely, C1', C2', C3', C4', C5', C2, C5, C6, C8), the corresponding chemical shift data

were assumed to contain a systematic error if the mean estimated offset (μ_{error}) was > 2 ppm and the ratio of the mean estimated offset and the standard deviation ($\mu_{\text{error}}/\sigma_{\text{error}}$) was > 5. This approach[19] was able to reproduce experimentally validated referencing errors previously identified by Aeschbacher *et* al. The R code used to detect and correct the chemical shifts and the corrected data set are available at: `https://github.com/atfrank/CS2Structure`.

After correcting (if necessary) the chemical shift data for each of the 115 RNAs in our initial data set, we determined the weighted (or reduced) mean absolute error (wMAE) between the corrected experimental chemical shifts and chemical shifts computed from the first model of each of the NMR bundles. RNAs that exhibited ^{1}H or ^{13}C wMAE that was $> 1.5 \times$ IQR (the interquartile range), were considered outliers and removed from our data set. The PDBIDs of the RNAs that were removed are: (1) 1S9S, (2) 1TJZ, (3) 2LC8, (4) 2AHT, (5) 2MQT, (6) 2M24, and (7) 5KMZ. For the remaining 108 RNAs (Table B.1), the secondary structures were retrieved using the program DSSR from the 3DNA suite.[20,21] From the secondary structure model, the base pairing status of each residue in the RNA was determined. In this work, we currently only included canonical base pairing interactions (that is, GC and AU Waston–Crick base pairs and GU wobble base pairs) and ignored noncanonical base pairs due to their overall under-representation in our data set.

2.2.1.2 Chemical Shift Imputation

To impute missing chemical shift data, we used the R package MICE (multivariate imputation by chained equations).[22] MICE assumes the probability of a data point being missing depends only on the observed data. To impute chemical shifts using MICE, separate regression models are built for the chemical shifts of each nucleus type based on the chemical shifts of other nucleus types. Next, multiple cycles of imputation are carried out using predictive mean matching (pmm). The pmm method

first identifies a set of observed chemical shifts whose regression-predicted values are closest to the regression-predicted value of the missing data point and then randomly selects one of those observed values to fill in the missing point.

For each of the 108 RNA systems in our data set, the imputed non-exchangeable (namely, C1', C2', C3', C4', C5', C2, C5, C6, C8, H1', H2', H3', H4', H2, H5, H5', H5", H6, H8) chemical shifts, along with the residue types and base pairing status of individual residues were combined to produce a single data structure (See Figure 2.1B) in which the rows corresponded to individual residues from each of the 108 RNAs and the columns corresponded to different nucleus types.

2.2.2 CS2BPS Classifiers

To predict the base pairing status of individual residues in an RNA based on the observed chemical shifts of atoms in those residues, we constructed artificial neural network (ANN) classifiers.[23] In an ANN, input features and output labels are connected through one or more layers of hidden neurons (Figure 2.1A). The neurons on adjacent layers are connected to each other through a linear transformation followed by an activation function. When training neural network, gradient descent is performed to update network weights through back-propagation until a tolerable loss is achieved.

Here, we built a chemical shift-based ANN classifier (referred to hereafter as, **C**hemical **S**hift to **B**ase-**P**airing **S**tatus classifier, CS2BPS) consisting of two dense layers and one dropout layer (the dashed lines between the second hidden layer and the output layer in Figure 2.1A; worked as regularization to avoid potential overfitting). For a given residue i, the chemical shift data of residues i, $i-1$, and $i+1$, along with the residue types of i, $i-1$, and $i+1$ were fed into the CS2BPS classifier and the classifier output the probability of residue i being unpaired (Figure 2.1C).

Hyperparameters, namely, the loss function, learning rate, dropout rate, batch

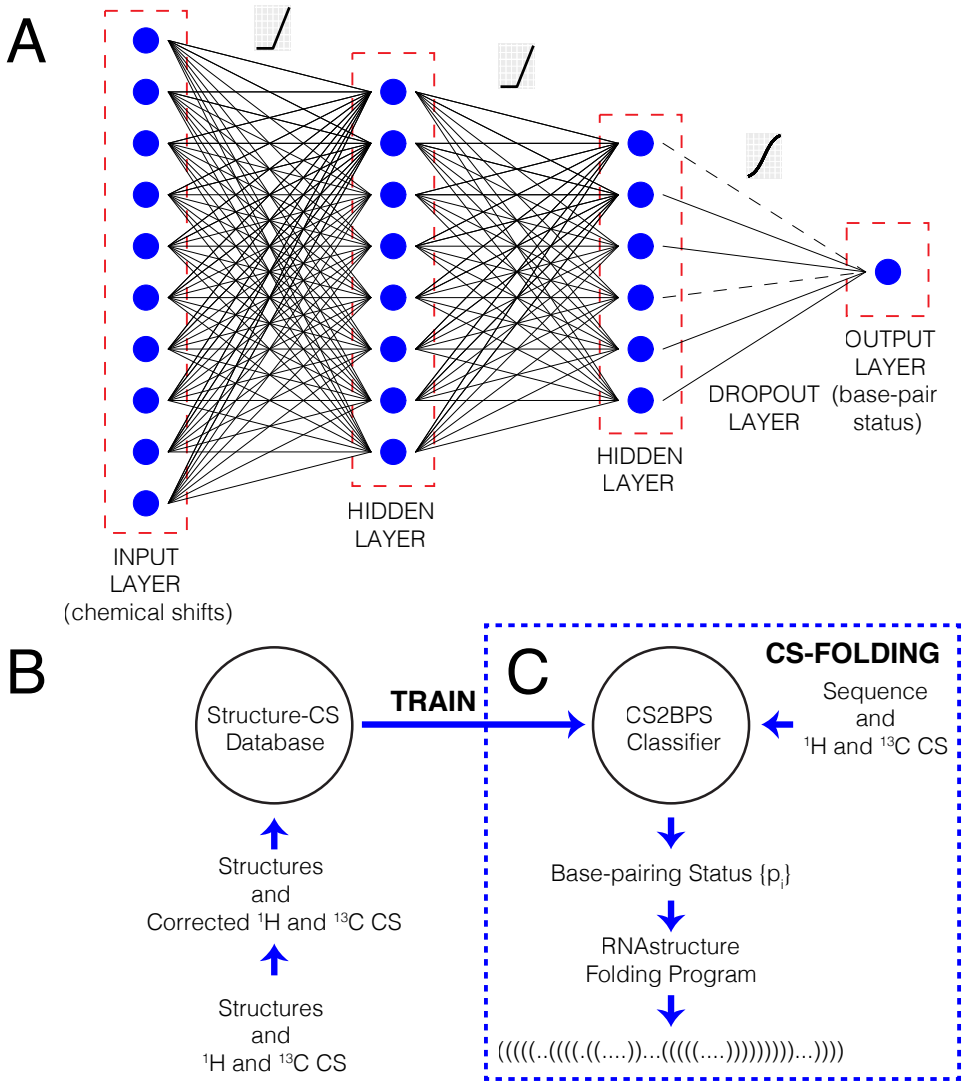

Figure 2.1: (A) Illustration of the artificial neural networks (ANNs) that we used to train our **C**hemical **S**hifts to **B**ase-**P**airing **S**tatus (**CS2BPS**) classifiers. The ANNs take as input (through the input layer) chemical shifts associated with an RNA residue and return (through the output layer) the probability of that residue being unpaired. (B) When developing the CS2BPS classifiers, we first obtained a data set containing NMR chemical shifts and NMR-derived secondary structures for 108 RNAs. Using a leave-one-RNA-out approach, we then trained a collection of independent CS2BPS classifiers. (C) Illustration of what we refer to as the CS-Fold framework, in which the optimized CS2BPS classifiers were used to predict the base pairing status of individual residues of a given RNA from its chemical shift data. The CS2BPS-derived base pairing status predictions were then used as restraints in RNA folding simulations to predict the secondary structure of the RNA.

size, optimization method and number of epochs were optimized through grid search cross validation. The CS2BPS base pairing status classifier was trained using Keras[24] with a TensorFlow backend.[25] For each CS2BPS classifier, we used a network containing two hidden layers followed by a dropout layer. The adjacent input and hidden layers were connected through ReLU activation function. A sigmoid activation was used on the output layer (Figure 2.1A).

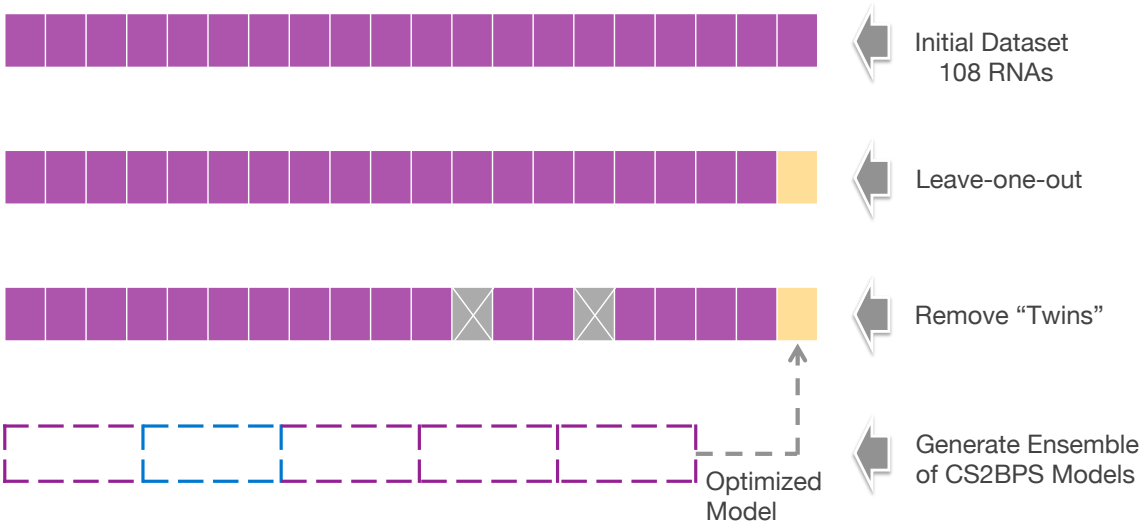

Figure 2.2: Leave-one-RNA-out Cross Validation.

Using a leave-one-RNA-out approach, independent sets of CS2BPS classifiers were generated for each RNA in our data set (Figure 2.2). For each RNA, RNAs in the training set with high sequence similarity ($\geq$ 80%), determined using the tool vsearch,[26] were first removed from the training set to avoid the "twinning" effect. Next, a set of independent CS2BPS classifiers were generated and then used to predict the base pairing status of individual residues in the left-out (testing) RNA. Here, we generated an ensemble of classifiers (six CS2BPS classifiers) for each left-out RNA.

To quantify the accuracy of the classifiers, we computed the sensitivity or the true positive rate (TPR) (defined as the probability that a base paired residue is predicted to be base paired) and the specificity or the true negative rate (TNR)

(defined as the probability that an unpaired residue is predicted to be unpaired) and the overall accuracy (defined as the fraction of residues in an RNA whose base pairing status are correctly predicted), for each testing RNA. Here, the predicted base pairing status corresponded to the average base pairing probability calculated from those six CS2BPS classifiers (see above). We reported the best-tuned parameters of the first set of classifiers for all 108 RNAs in Table B.2.

2.2.3 Assess the use of CS2BPS classifiers to guide secondary structure prediction

To assess the use of the CS2BPS classifiers to guide secondary structure prediction, we implemented a **CS-Fold framework**. Within this CS-Fold framework, the base pairing status predictions, which were calculated by averaging the results of six independent CS2BPS classifiers, were used as restraints in RNA folding simulations (Figure 2.1B). Similar to the approach used to incorporate SHAPE-derived restraints[27–30] into RNA folding simulations, we utilized a restraint term of the form

$$\Delta G = \Delta G_{\text{thermo}} + \Delta G_{\text{cs}}, \tag{2.1}$$

where ΔG is the folding free energy, ΔG_{thermo} is thermodynamic free energy, and ΔG_{cs} is the base pairing restraint term which has the functional form

$$\Delta G_{\text{cs}} = m \sum_{i}^{N} \left[\ln \left(p_i + 1 \right) + b \right]. \tag{2.2}$$

Here m and b are restraint parameters that influence the magnitude of the restraint free energy relative to the thermodynamic free energy, and p_i is the probability of a residue being unpaired (Figure 2.1C) that is calculated by averaging the predictions from the ensemble of six CS2BPS classifiers. In the CS-Fold framework, we first used the folding algorithms from the *RNAstructure* modeling suite[31] to generate a

set of possible secondary structure models of each of the 108 RNAs in our data set, incorporating the CS2BPS predictions as folding restraints (Figure 2.1C).

For each of the 108 RNAs in our data set, CS-Fold simulations were carried out using the folding algorithms Fold,[12] MaxExpect,[32] and ProbKnot[33] from the *RNAstructure* modeling suite. Fold predicts secondary structures by free energy minimization. MaxExpect generates secondary structure models that contain highly probable base pairs. And ProbKnot, like MaxExpect, generates secondary structure models that contain highly probable base pairs, but allows pseudoknots. In all CS-Fold simulations, m and b were set to 1.8 kcal/mol and 0.6 kcal/mol, respectively. These values corresponded to the default values used to incorporate normalized SHAPE reactivities into RNA folding simulations.[34] To select the best model among structures generated by Fold, MaxExpect, and ProbKnot algorithms, with and without CS2BPS-derived predictions as restraints, we defined the consistency score as the fraction of $\{s_{\text{fold}}\}$ that is identical to $\{s_{\text{CS2BPS}}\}$, where $\{s_{\text{fold}}\}$ is the base pairing status of individual residues from Fold, MaxExpect, and ProbKnot generated structures and $\{s_{\text{CS2BPS}}\}$ is the base pairing status derived from the CS2BPS classifiers. Among these six possible structures, the one that has the highest consistency score with CS2BPS base pairing status predictions was selected as the final predicted secondary structure model for a given RNA.

To assess the accuracy of the CS-Fold generated secondary structure models, we used the program scorer,[11] from the *RNAstructure* suite. Given a reference secondary structure model and a comparison structure, scorer calculates the sensitivity or true positive rate (TPR), defined as the fraction of base pairs in the comparison structure that also appeared in the reference structure, and positive predictive value (PPV), defined as the fraction of reference base pairs that also appeared in the comparison structure. In our case, the reference structure is the native NMR-derived structure and the comparison structure is the CS-Fold predicted structure.

2.3 Data analysis and results

2.3.1 Base pairing status from chemical shifts

We began our study by training a set of machine learning classifiers to predict the base pairing status of individual residues in an RNA from the ^{1}H and ^{13}C chemical shift "fingerprints" of each residue. To develop these classifiers, we utilized the artificial neural network machine learning technique (Figure 2.1A). Briefly, an artificial neural network (ANN) takes in information through an *input layer*, then passes it through one or more *hidden layers* and finally passes it through an *output layer*. Our ANN classifiers, which we refer to as **C**hemical **S**hift to **B**ase-**B**airing **S**tatus (**CS2BPS**) classifiers, are fed the chemical shifts for individual residues in an RNA as well as the chemical shifts of the residues before and after it. The network outputs the base pairing status of each residue (i.e., the probability that a given residue is base paired to some other residue). We note that for residues that are predicted to be base paired, our CS2BPS classifiers *do not identify their base pairing partner*.

Using a data set containing NMR chemical shifts and NMR-derived secondary structures for 108 RNAs, we built six independent CS2BPS classifiers for each RNA in our data set using a leave-one-RNA-out approach (Figure 2.2). Briefly, to build each classifier, data associated with one of the 108 RNAs (the left-out RNA) were removed from the data set. Then six CS2BPS classifiers were trained using data from the other RNAs (i.e., the training set). The resulting CS2BPS classifiers were then tested on the left-out RNA. To mitigate bias due to the "twinning" effect in which data in training set closely resemble the left-out data, when building each CS2BPS classifier, we also excluded from the training set data associated with any RNA(s) that exhibited a high sequence similarity ($\geq 80\%$) to the left-out RNA (see Methods).

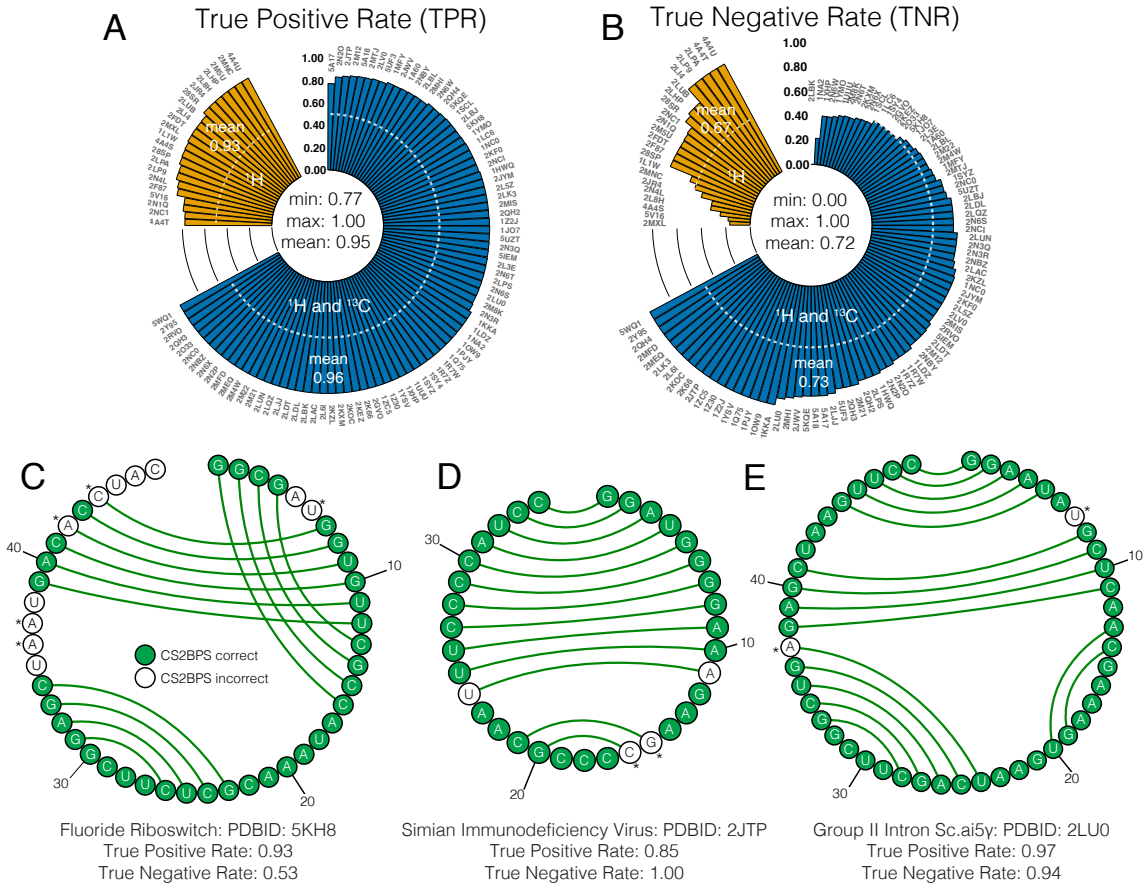

Figure 2.3: (A and B) CS2BPS Classification Accuracy. Shown are circular bar plots of (A) the sensitivity or true positive rate (TPR) and (B) the specificity or true negative rate (TNR). Accuracy statistics are based on a leave-one-RNA-out analysis. As a guide, the 0.5 accuracy levels are shown in white dashed lines. In the plots, bars are grouped based on whether only ^{1}H (*orange*) or whether both ^{1}H and ^{13}C (*blue*) non-exchangeable chemical shifts were available in the corresponding RNA systems. (C-E) Representative examples of CS2BPS predictions. Shown are the CS2BPS predictions projected onto the native structures of (C) the fluoride riboswitch (PDBID: 5KH8), (D) the simian immunodeficiency virus (SIV) RNA (PDBID: 2JTP), and (E) the group II intron Sc.ai5γ RNA (PDBID: 2LU0). Green circles indicate that our CS2BPS predictions were consistent with the base pairing status in the native structure, whereas white circles indicate that our CS2BPS predictions were incorrect. Residues labeled with '*' exhibited high variance (see Table B.3, B.4, and B.5) in their base pairing classification across six independent CS2BPS classifiers.

2.3.1.1 Overall Accuracy

Figure 2.3 summarizes the accuracy of the 108 CS2BPS classifiers. Reported are the true positive rate (TPR) and the true negative rate (TNR). Here a residue is base paired if its mean classification probability was ≥ 0.4. This classification threshold was chosen so as to maximize the overall classification accuracy (that is, the fraction of residues whose base pairing status were correctly predicted).

The mean TPR and the mean TNR of the CS2BPS classifiers were 0.95 and 0.72, respectively; TPR values ranged between 0.77 and 1.00 (Figure 2.3A) and TNR values ranged between 0.00 and 1.00 (Figure 2.3B). These results indicate that our CS2BPS classifiers were better at identifying base paired residues than unpaired residues. The comparatively lower TNR of our CS2BPS classifiers can be attributed to an imbalance in our data set; the total number of base paired and unpaired residues contained in our data set were 2329 and 1023, respectively (Table 2.1). This imbalance in our data set might also explain why the classification threshold that maximized the overall classification accuracy was 0.4, rather than 0.5.[35]

Table 2.1: Residue base pairing status in data set

Residue type	Number of paired residues	Number of unpaired residues
adenine (A)	386	363
guanine (G)	779	206
cytosine (C)	686	173
uracil (U)	478	281
Total	2329	1023

Out of the 108 systems in our data set, 22 corresponded to systems for which only ^{1}H chemical shifts were available. For these 22 systems with only ^{1}H chemical shifts, and for which the corresponding ^{13}C chemical shifts had to be imputed (see Methods), the mean TPR value was 0.93 (Figure 2.3A; *orange*); by comparison, the mean TPR value for systems for which both ^{1}H and ^{13}C chemical shifts were available was 0.96

(Figure 2.3A; *blue*). On the other hand, for systems for which only ^{1}H chemical shifts were available, the mean TNR value was 0.67 (Figure 2.3B; *orange*), compared to 0.73 for systems for which both ^{1}H and ^{13}C chemical shifts were available (Figure 2.3B; *blue*). As such, the CS2BPS classifiers exhibited lower TPR and TNR when only ^{1}H chemical shifts were available (Figure 2.3A and 2.3B); however, the reduction in performance was modest.

2.3.1.2 Accuracy by residue types

Table 2.2: CS2BPS TPR and TNR by residue types

Residue type	TPR	TNR	Instances
A	0.90	0.80	749
G	0.96	0.66	985
C	0.97	0.56	859
U	0.92	0.64	759

Shown in Table 2.2 is a breakdown of the CS2BPS performance for individual residue types, namely A (adenine), G (guanine), C (cytosine), and U (uracil). The TPRs ranged between 0.90 and 0.97. For G and C residues, the TPRs were 0.96 and 0.97, respectively. By comparison, the TPRs for A and U residues were 0.90 and 0.92. The TNRs for individual residue types ranged between 0.56 and 0.80. For G and C residues, the TNRs were 0.66 and 0.56, respectively. By comparison, the TNRs for A and U residues were slightly higher: the values were 0.80 and 0.64, respectively.

2.3.1.3 Accuracy by base pair types

Though our CS2BPS classifiers cannot predict the base pairing partners for residues that are estimated to have a high probability of being base paired, we were nonetheless interested in exploring whether our CS2BPS classifiers were able to correctly predict the base pairing status of both residues in individual base pairs. To explore

this, all of the GC (or CG), AU (or UA), and GU (or UG) canonical base pairs were identified, and a TPR score was calculated for each base pair type.

Table 2.3: CS2BPS TPR and TNR by base pair types

base pair type	TPR[1]	TNR	Instances
GC	0.93	N/A	1374
AU	0.88	N/A	772
GU	0.75	N/A	183

[1] Here, TPR or "sensitivity" is defined as the probability that both residues in a base pair are correctly predicted to be base paired.

In this case, the TPR was defined as the probability that both residues in a base pair were correctly predicted to be base paired. We found that for GC, AU, and GU base pairs, the TPR values were 0.93, 0.88, and 0.75, respectively (Table 2.3). Examining the number of instances of each base pair type in our database indicates these differences in TPR is most likely a result of an imbalance among the different types of base pairs in our data set: the total instance of GC base pairs was 1374 compared to only 772 and 183 for AU and GU, respectively (Table 2.3).

2.3.1.4 Examples from leave-one-RNA-out cross validation

Shown in Figure 2.3C-E are detailed comparisons between actual and predicted base pairing status for three representative RNAs in our data set. For the first example, the 47-nt fluoride riboswitch RNA (PDBID: 5KH8),[36] our CS2BPS predictions exhibited TPR and TNR values of 0.93 and 0.53, respectively (Figure 2.3C). This was one of the four structures in our data set that contained pseudoknot interactions and whose overall prediction accuracy was only around the 9th percentile. Interestingly, for this RNA, most of the residues that participated in long-range tertiary contacts were correctly predicted to be base paired (namely, residues 7-12 and 39-44) (Figure 2.3C). For this RNA, the major source of error was the misclassification of residues

35-38 and 45-47. Interestingly, for 5 misclassified residues (namely, residue 5, 36, 37, 42, and 44), we found significant variance in their classification across the six independent CS2BPS classifiers (Figure 2.3C; Table B.3).

The second example, the 34-nt simian immunodeficiency virus (SIV) RNA (PDBID: 2JTP)[37] was selected because its classification accuracy was identical to the median classification accuracy (0.88) across the entire data set. For this RNA, our CS2BPS predictions exhibited TPR and TNR values of 0.85 and 1.00, respectively (Figure 2.3D). The errors were due to residues 11, 15, 16, and 24 being misclassified as unpaired. As was the case for some of the misclassified residues in the fluoride riboswitch, two of the misclassified residues of the SIV RNA also exhibited high variance in their CS2BPS classification (Figure 2.3D; Table B.4).

For the third example, the 49-nt group II Intron Sc.ai5γ RNA (PDBID: 2LU0),[38] our CS2BPS predictions exhibited TPR and TNR values of 0.97 and 0.94, respectively (Figure 2.3E). The only two residues that were misclassified, residue 7 and 37, also exhibited high variance in their CS2BPS classification (Figure 2.3E; Table B.5).

In general, we discovered that the TNR was significantly lower than the TPR for CS2BPS classifiers. In some cases, we found that a fraction of residues that were misclassified exhibited high variance in their base pairing predictions (See Table B.3, B.4, and B.5). It should be noted that not all residues with high prediction variance were misclassified. Collectively, however, these results show that, given a set of *assigned* ^{1}H and ^{1}3C chemical shifts for a given RNA, our CS2BPS classifiers could be used to predict the base pairing status of individual residues.

2.3.1.5 Chemical shifts error analysis

To assess the sensitivity of our CS2BPS predictions to non-systematic errors in the chemical shift data, we simulated the presence of errors by adding "noise" to the measured chemical shifts of each RNA system in our training set and then used

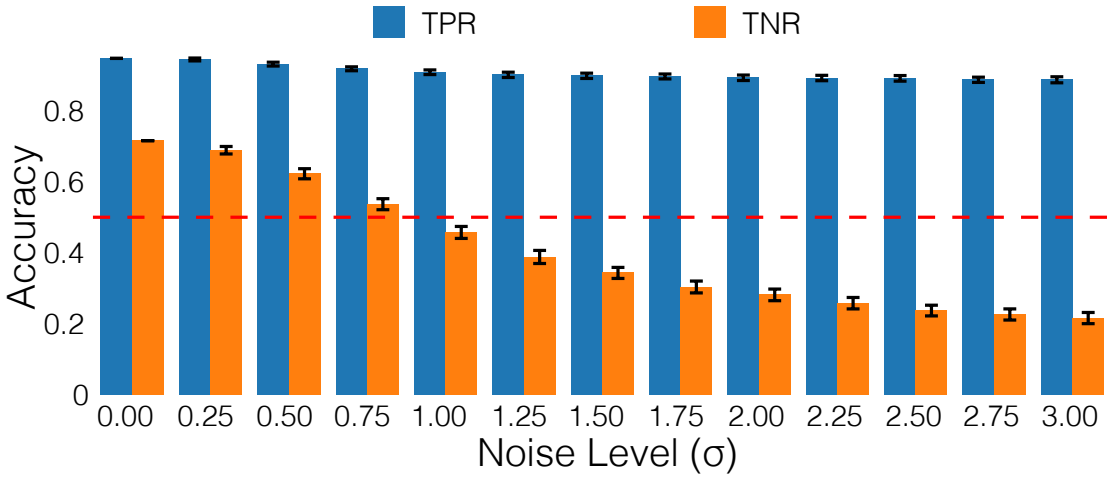

Figure 2.4: The sensitivity of CS2BPS classifiers to errors in chemical shifts data. Here, σ is the standard deviation calculated from published experimental chemical shifts of specific residue and nucleus type.

the "noisy" chemical shifts as inputs to our CS2BPS classifiers. Shown in Figure 2.4 are the TPR and TNR values for noise-levels ranging between 0.0 and 3.0σ, respectively, where σ is the standard deviation of experimental RNA chemical shifts deposited in BMRB. As expected, both the mean TPR and TNR decreased as the noise-levels increased, with the TNR exhibiting greater sensitivity to the added noise. For example, at the 1σ, 2σ, and 3σ noise-levels, TPR values were 0.91, 0.89, and 0.89 whereas the TNR values were 0.46, 0.28, and 0.22 (Figure 2.4). By comparison, the TPR and TNR values that we observed when using the original noise-free data were 0.95 and 0.72 (Figure 2.4), respectively. These results indicate that the performance of the predictions is indeed sensitive to the presence of errors in the chemical shift data. For unpaired residues in particular, for the prediction to be better than random (TNR > 0.5), errors in the chemical shifts, assuming that normally distributed, must be $< 0.75\sigma$. These results, coupled with the observation that we could achieve reasonable accuracy when predicting base pairing status for residues in RNA for which all the ^{13}C had to be imputed, strongly suggest that the errors introduced by imputation were mostly likely less than 0.75σ.

2.3.2 Guiding RNA secondary structure prediction

Next, we examined whether the residue-wise base pairing probabilities that were predicted using our CS2BPS classifiers could be used to guide RNA secondary structure modeling. Given that an RNA can adopt distinct conformational states, each with a set of distinct chemical shift fingerprints, we sought to develop an approach that allowed us to predict the secondary structure of an RNA *conditioned* on a set of *assigned* chemical shift data. Such a method would be useful in mapping the structure landscape of an RNA from available chemical shift data.

2.3.2.1 Overall Accuracy

To predict RNA secondary structure *conditioned* on a set of chemical shifts, we implemented a CS-Fold framework in which CS2BPS-derived pairing predictions were used as restraints in RNA folding simulations (Figure 2.1C). Within this modeling framework, chemical shifts were taken as inputs and fed into a CS2BPS classifier to predict the base pairing status of individual residues. These predictions were then used as restraints to guide RNA folding to produce a secondary structure model. This modeling approach closely resembles the approach used to guide modeling using chemical mapping data.[29,30]

Briefly, for each of the 108 RNAs in our data set, we predicted its secondary structure using the Fold, ProbKnot, and MaxExpect algorithms in *RNAstructure* suite, both with and without the single residue pairing restraints derived from the corresponding CS2BPS classifiers (Eq. 2.1, 2.2). Among these six predicted structures, the structure that was most consistent with the CS2BPS base pairing predictions was selected and then compared to the NMR-derived reference secondary structure model. In the cases where more than one structure had the same consistency, the structure with the lowest folding free energy was chosen.

Shown in Figure 2.5 are circular bar plots of the TPRs (which is defined as the

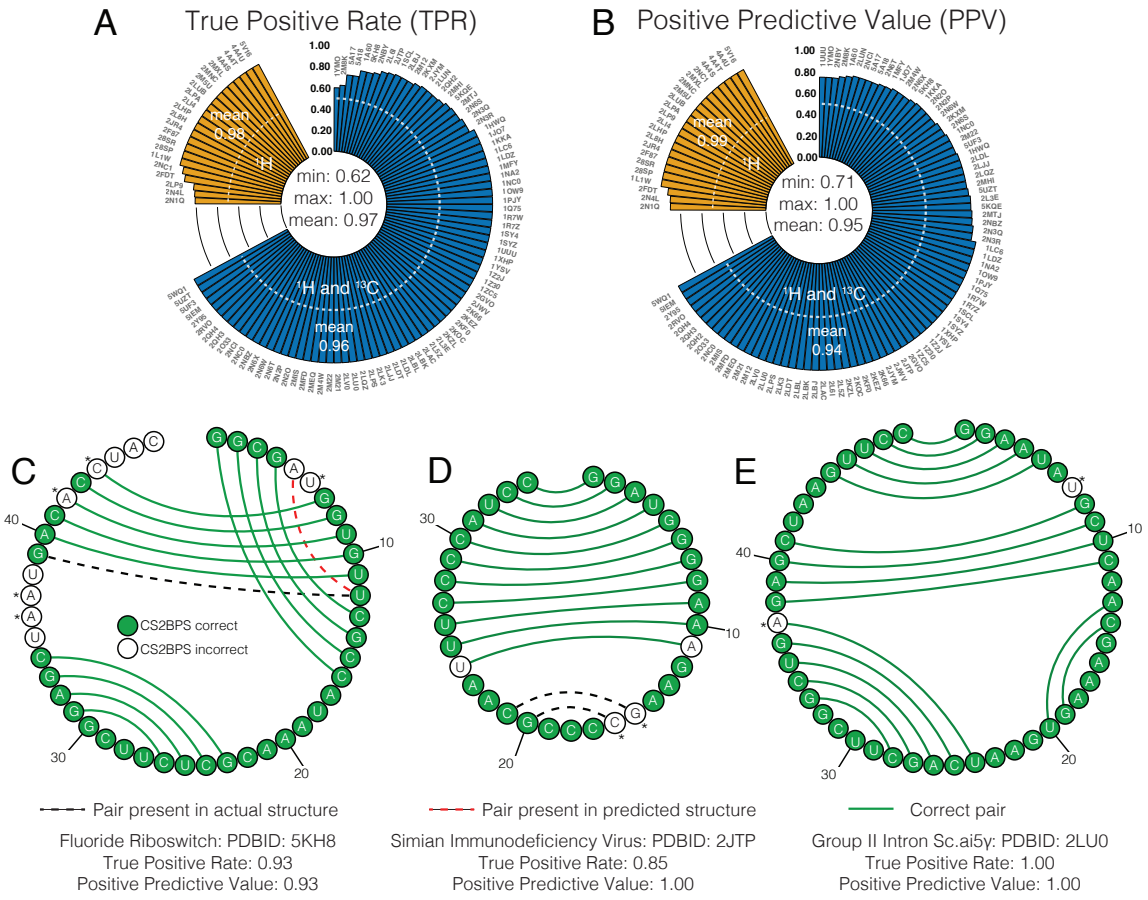

Figure 2.5: (A and B) CS-Fold Accuracy. Shown are circular bar plots of (A) the TPR and (B) the PPV values obtained when comparing the reference NMR secondary structure of each RNA to the model obtained from folding the RNA using CS2BPS-derived base pairing probabilities as folding restraints. As a guide, the 0.5 accuracy levels are shown in white dashed lines. (C-E) CS-Fold results. Shown are the comparison between CS-Fold predicted structures and secondary structure models derived from NMR bundle for (C) the fluoride riboswitch (PDBID: 5KH8), (D) the simian immunodeficiency virus (SIV) RNA (PDBID: 2JTP), and (E) the group II intron Sc.ai5γ RNA (PDBID: 2LU0). base pairs that are shown as green lines were present in both the CS-Fold structure and the NMR structure, whereas base pairs that are shown as red dashed lines were only present in the CS-Fold structure and base pairs that are shown as black dashed lines were only present in the NMR structure.

fraction of base pairs in the predicted structure that also appeared in the NMR-derived structure) and PPVs (which is defined as the fraction of base pairs in the NMR-derived structure that also appeared in the predicted structure) for the CS-Fold results of the 108 RNAs in our data set. In general, the CS-Fold generated structures exhibited high TPR (0.97) and high PPV (0.95) values. For RNAs for which only ^{1}H chemical shifts were available, the prediction TPR and PPV were 0.98 and 0.99, respectively (Figure 2.5A and 2.5B). In comparison, for RNAs for which both ^{1}H and ^{13}C chemical shifts were available, the TPR and PPV values were 0.96 and 0.94, respectively (Figure 2.5A and 2.5B).

2.3.2.2 Accuracy by base pair types

Shown in Table 2.4 are the TPRs for the recovery of GC, AU, and GU base pairs in the CS-Fold generated structures. In general, CS-Fold framework was able to recover all three base pair types with high TPRs, 0.97, 0.98 and 0.90 for GC, AU and GU base pairs, respectively. The slightly lower TPR for GU base pairs was most likely due to the lower instances of GU base pairs in the 108 RNAs data set. Our data set contained 184 of GU base pairs compared to 772 AU and 1378 GC base pairs, respectively.

Table 2.4: CS-Fold TPR and TNR by base pair types

base pair type	TPR[1]	TNR	Instances
GC	0.97	N/A	1378*
AU	0.98	N/A	772
GU	0.90	N/A	184*

* Here, TPR is defined as the fraction of a certain base pair type that is correctly recovered in CS-Fold generated structures. Note that the instances of GC and GU base pairs differ from those in Table 2.3 (labeled with '*') because there were some residues for which chemical shift data were not available and were not included in Table 2.3.

2.3.2.3 Representative examples

Shown in Figure 2.5C-E are detailed comparisons between the native secondary structures and the CS-Fold results for the fluoride riboswitch RNA (PDBID: 5KH8) (Figure 2.5C), the SIV RNA (PDBID: 2JTP) (Figure 2.5D), and the group II intron Sc.ai5γ RNA (PDBID: 2LU0) (Figure 2.5E), respectively. For each structure, only the canonical base pairs are shown.

For the fluoride riboswitch RNA, the TPR was 0.93 (Figure 2.5C). The predicted CS-Fold structure recovered 5 out of the 6 pseudoknotted base pairs as well as all of the non-pseudoknotted base pairs. Interestingly, for the pseudoknotted U12-G39 base pair that was missing in the CS-Fold structure, our CS2BPS classifier predicted both of these residues to be base paired (Figure 2.5C). For this RNA, the PPV was also high (0.93); the CS-Fold structure only contained a single extraneous A5-U12 base pair (Figure 2.5C).

For the SIV RNA and the group II intron Sc.ai5γ RNA, the majority of the base pairs in the reference NMR models were correctly recovered, 11 out of 13 (TPR=0.85)(Figure 2.5D) and 16 out of 16 (TPR=1.00)(Figure 2.5E), respectively. In both cases, no extraneous base pairs were found in the CS-Fold structures.

2.3.2.4 Application of CS-Fold to the microRNA-20b pre-element

To test whether the CS-Fold framework could be used to *conditionally* predict the secondary structure of an RNA, we first applied it to the 23-nt long microRNA-20b (miR-20b) pre-element, for which two distinct conformational states have recently been characterized using NMR spectroscopy: an unbound (*apo*) state (Figure 2.6A) and a Rbfox RRM protein-bound (*holo*) state (Figure 2.6B).[39] When interacting with the conserved Rbfox RRM protein, two canonical base pairs that are present in the *apo* state are disrupted (Figure 2.6A and B), enabling the protein and the RNA to interact in a sequence-specific manner. In addition to atomic structures, the *assigned*

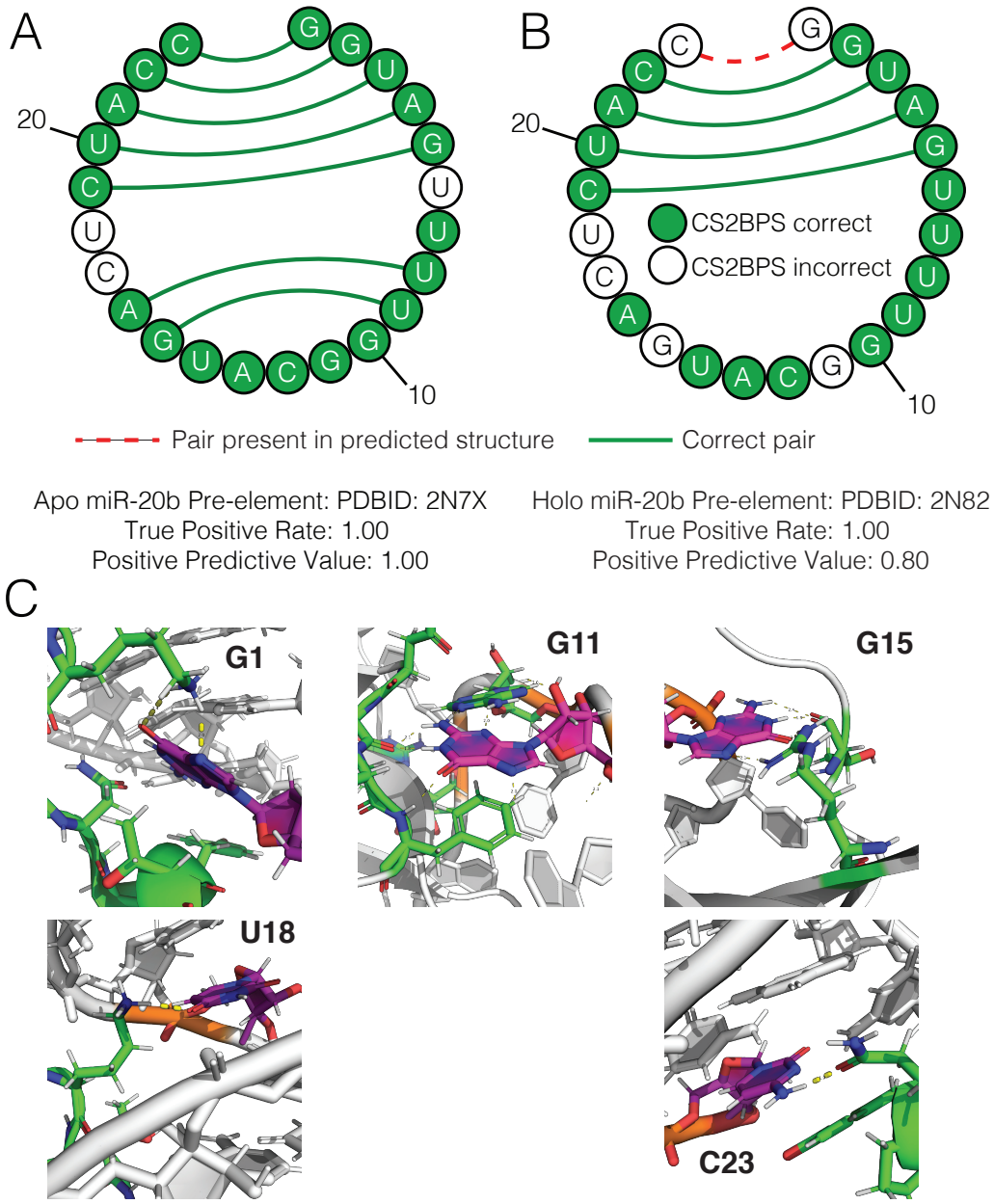

Figure 2.6: CS-Fold results for (A) the *apo* state and (B) the *holo* state of miR-20b RNA. Shown in (C) are residues G1, G11, G15, U18, and C23, which, based on the secondary structure of the *holo* state of miR-20b were initially thought to be "misclassfied" as being base paired, but upon closer examination of the 3D structure of the miR-20b-Rbfox complex were revealed to be hydrogen bonded to Rbfox RRM protein.

chemical shift data corresponding to the *apo* and *holo* conformational states were available, enabling us to test whether we could use the two sets of *assigned* chemical shifts to *conditionally* predict structures of the miR-20b RNA. Using the *apo* and *holo* chemical shifts, we predicted the base pairing status of each residue in miR-20b using our CS2BPS classifiers and then used the CS-Fold framework to predict their secondary structures.

Shown in Figure 2.6A and 2.6B are the detailed comparisons between the native secondary structures and the predicted structures. In the case of the *apo* state, the TPR and PPV between native and predicted CS-Fold structures were both 1.00 (Figure 2.6A), indicating that we were able recover the native secondary structure. Similarly, for the *holo* state, the TPR and PPV between native and predicted CS-Fold structures were 1.00 and 0.80, respectively (Figure 2.6B). The only error in the CS-Fold generated structure of the *holo* state was an extraneous base pair between residues G1 and C23. These results indicate that by biasing the folding algorithms using CS2BPS predictions, we were able to *conditionally* predict the two distinct conformational states of the miR-20b RNA.

Though the CS-Fold structures closely resembled the reference NMR structures of *apo* and *holo* states, respectively, the CS2BPS predictions which we used as folding restraints to predict their structures, contained what appeared, initially, to be several inconsistencies. For example, in the *apo* state, residues U6, C17, and U18 were "misclassified" as being base paired (Figure 2.6A). Closer examination of the structure of the *apo* state (PDBID: 2N7X) revealed that these residues were, however, involved in noncanonical base pairs, which we ignored in the study, because of their under-representation in our data set and due to the fact that Fold, MaxExpect, and ProbKnot algorithms currently only predict canonical base pairs. Similarly, in the *holo* state, residues G1, G11, G15, C17, U18, and C23 were all "misclassified" as being base paired, on the basis of the *holo* state secondary structure of the miR-20b RNA

(Figure 2.6B). Closer examination of the structure of the *holo* state (PDBID: 2N82) (including the Rbfox RRM protein) revealed that with the exception of C17, these residues were involved in hydrogen bond interactions with the Rbfox RRM protein (Figure 2.6C).

2.3.2.5 Application of CS-Fold to the Fluoride Riboswitch

As a second test of whether the CS-Fold framework could be used to *conditionally* predict the secondary structure of an RNA, we next applied it to model two distinct states of the fluoride riboswitch. The first state, referred as the Mg^{2+}-free state, corresponds to the riboswitch RNA in the absence of Mg^{2+} ions and its cognate fluoride ion and the second state, the *apo* state, corresponds to the riboswitch in the presence of Mg^{2+} ions but also in the absence of fluoride. Bo and Zhang recently used NMR spectroscopy to build a secondary structure model of the free state, and from their study, a set of *assigned* C1′/H1′ and C8/H8 chemical shift data were available for the free state.[40] Bo and Zhang also used NMR spectroscopy to determine the atomic structure of the *apo* state, and from this study, a nearly complete set of *assigned* chemical shift data were available for the *apo* state.[40] In the absence of Mg^{2+} ions, the fluoride riboswitch exists predominantly as a pair of non-nested hairpin loops (Figure 2.7A) whereas in the presence Mg^{2+} ions, it exists predominantly as a pseudoknotted structure (Figure 2.7B). With access to experimentally-validated secondary structure models, along with a set of *assigned* chemical shifts, we applied our CS-Fold framework to model the structure of the Mg^{2+}-free and *apo* states of the fluoride riboswitch.

Because only the C1′/H1′ and C8/H8 for guanine residues were available for the Mg^{2+}-free state, for consistency, we carried out CS-Fold for both states with only C1′/H1′ and C8/H8 chemical shift data for guanine residues in the fluoride riboswitch. Remarkably, despite the sparsity of the *assigned* chemical shift data we utilized, we

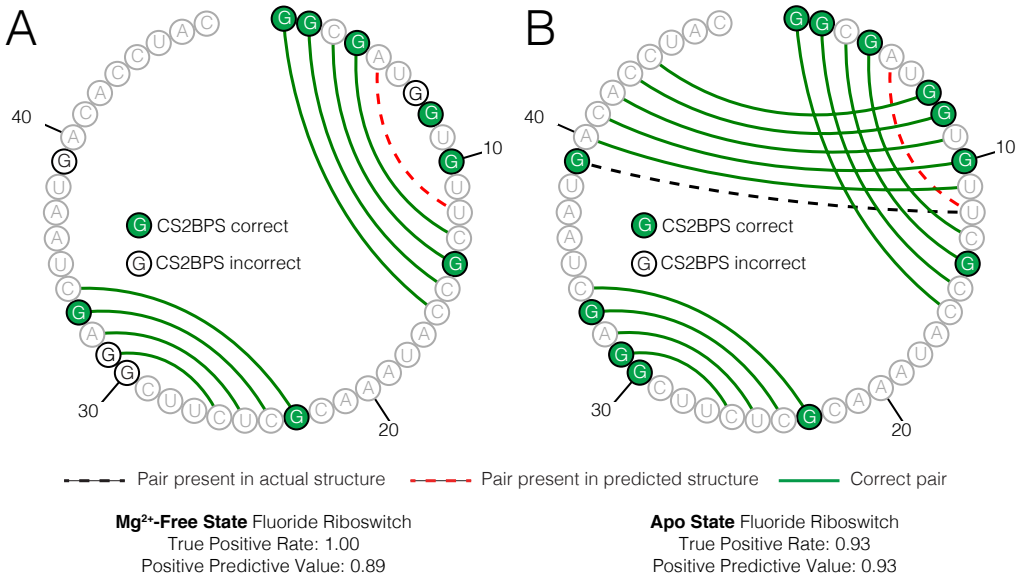

Figure 2.7: CS-Fold results for the fluoride riboswitch. Shown are comparisons between the experimentally validated secondary structure and the CS-Fold results of (A) the Mg^{2+}-free and (B) the *apo* state of the fluoride riboswitch. Here, CS-Fold was carried out using *assigned* C1'/H1' and C8/H8 chemical shift data for guanine residues only.

were able to build reasonably accurate models for both the Mg^{2+}-free and *apo* states using the CS-Fold framework. For instance, for the free state the TPR was 1.00 and PPV was 0.89 (Figure 2.7A), indicating that the CS-Fold structure closely resembled the experimentally-validated secondary structure model. For the *apo* state, the TPR and PPV were 0.93 and 0.93, respectively (Figure 2.7B). Interestingly, we achieved the same accuracy using complete chemical shifts of the *apo* state (Figure 2.5).

2.4 Discussion

In this study, we generated a set of artificial neural network classifiers that were capable of predicting the base pairing status of individual residues in RNAs directly from their non-exchangeable ^{1}H and ^{13}C chemical shift signature. These classifiers, which we referred to as CS2BPS classifiers, were able to identify base paired residues with relatively high true positive rate (TPR), regardless of the residue and base pair

types (Table 2.2 and 2.3). In cases where only ^{1}H chemical shifts were available, we found that the base pairing status of residues could still be accurately predicted (Figure 2.3A and 2.3B). Indirectly, this observation suggests that the errors introduced by the MICE imputation, which we estimated to be 1.25 ppm for ^{13}C nuclei (see Table B.7), was below the noise-level that would significantly reduce our prediction accuracy (see Figure 2.4).

Heavily inspired by previous work in which single nucleotide SHAPE reactivities were used to guide RNA folding algorithms,[27–30] we also explored whether the base pairing status predictions derived from our CS2BPS classifiers could be used to guide RNA secondary structure folding algorithms. Within what we refer to as a CS-Fold framework, in which the CS2BPS-derived base pairing status predictions (represented as single-residue pairing probabilities) were used as folding restraints (Eq. 2.1 and 2.2), we found that we could recover the correct fold of most of the 108 RNAs in our data set with remarkable accuracy. When guiding the Fold, Probknot, and MaxExpect algorithms from *RNAstructure* suite[31] with our CS2BPS-derived predictions and then identifying the structure with the highest base pairing status consistency with our CS2BPS predictions, we were able to achieve mean TPR and PPV values of 0.97 and 0.95, respectively (Figure 2.5). By comparison, the TPR and PPV values were 0.94 and 0.93, 0.95 and 0.92, and 0.94 and 0.93, respectively (Table B.6), when using Fold, ProbKnot, and MaxExpect by themselves (that is, not restrained using our CS2BPS predictions).

To test whether we could *conditionally* predict the secondary structure of an RNA, we applied our CS-Fold approach to microRNA-20b (miR-20b). For this RNA, two distinct conformational states, the free (*apo*) state and the protein bound (*holo*) state, were recently characterized using NMR spectroscopy.[39] Access to the structures and chemical shift data associated with both states of miR-20b enabled us to test whether we could use the CS-Fold framework to *conditionally* predict its secondary structure.

We discovered that we could recover, with high TPR and PPV the canonical base pairs for the *apo* and those for the *holo* conformational states of miR-20b, respectively (Figure 2.6A and 2.6B). Similar results were obtained when we applied the CS-Fold framework to the fluoride riboswitch (Figure 2.7). Collectively, these results suggest that given the chemical shifts for individual conformational states of an RNA, the CS-Fold modeling approach might be a viable technique for predicting the structure of each conformational state.

One significant limitation of our method is that it requires *assigned* chemical shifts. Indeed, when chemical shifts have been *assigned* for an RNA, base pairing interactions within the RNA (and thus the secondary structure) can be directly determined using NOESY NMR experiments.[41] Moreover, to assign chemical shifts, a secondary structure model is typically assumed. The CS-Fold does not, therefore, provide a significant advantage over conventional NMR methods for determining the secondary structure of RNAs. Immediately, we envision that the CS-Fold framework we demonstrated in this work can, however, be used as a tool to *independently* validate NOESY-derived secondary structural models of RNAs. With recent advances in singly-labeled RNA synthesis,[42] the chemical shifts of spin-active nuclei on individual residues in an RNA can, in principle, be unambiguously *assigned*, without any assumptions about the secondary structure that is adopted by the RNA. In such cases, we envision that the CS-Fold framework we described here will be an indispensable tool for objectively modeling the secondary structure of RNA based on chemical shifts derived from sets of singly-labeled NMR experiments.

Increasingly, there is keen interest in characterizing the transient states of RNAs. Unfortunately, it is not currently possible to detect the NOEs associated with these transient states. As such, conventional methods cannot be used to infer the secondary structure associated with the transient state or states of an RNA. Fortunately, it is now possible to characterize the ^{1}H and ^{13}C chemical shift signature of RNA transient

states using techniques based on saturation transfer[36,40] and relaxation dispersion.[16,43] The results we presented for the miR-20b RNA and the fluoride riboswitch, suggest that with access to these ^{1}H and ^{13}C chemical shifts, a CS-Fold framework, which utilizes predictions derived from CS2BPS classifiers like the ones we developed in this work, could be used to generate putative models for the transient states of RNAs.

To facilitate the community-wide use of our CS2BPS classifiers, we make available to the academic community a command-line tool with which users can predict the base pairing status of individual residues in an RNA from their *assigned* chemical shifts. These CS2BPS predictions can then be used to guide RNA secondary structure prediction using external tools like Fold, ProbKnot, and MaxExpect (from the *RNAstructure* suite) or other RNA folding tools that accept and incorporate single residue pairing probabilities as folding restraints. The command-line tool has been incorporated into our CS2Structure repository and can be accessed via: `https://github.com/atfrank/CS2Structure`. The input file of chemical shifts can be downloaded from BMRB and prepared using the script in our repository.

2.5 References

(1) Guerrier-Takada, C.; Gardiner, K.; Marsh, T.; Pace, N.; Altman, S. *Cell* **1983**, *35*, 849–857.

(2) Sharp, P. A. *Cell* **2009**, *136*, 577–580.

(3) Ponting, C. P.; Oliver, P. L.; Reik, W. *Cell* **2009**, *136*, 629–641.

(4) Dethoff, E. A.; Chugh, J.; Mustoe, A. M.; Al-Hashimi, H. M. *Nature* **2012**, *482*, 322.

(5) Gesteland, R. F.; Cech, T. R.; Atkins, J. F., *The RNA World*; Cold Spring Harbor Lab: 1999.

(6) Reining, A.; Nozinovic, S.; Schlepckow, K.; Buhr, F.; Fürtig, B.; Schwalbe, H. *Nature* **2013**, *499*, 355.

(7) Haller, A.; Rieder, U.; Aigner, M.; Blanchard, S. C.; Micura, R. *Nat. Chem. Biol.* **2011**, *7*, 393.

(8) Chen, S.-C.; Olsthoorn, R. C. *J. Virol.* **2010**, *84*, 1423–1429.

(9) Cruz, J. A.; Westhof, E. *Cell* **2009**, *136*, 604–609.

(10) Sim, A. Y.; Levitt, M. *Proc. Natl. Acad. Sci. U. S. A.* **2011**, *108*, 3590–3595.

(11) Mathews, D. H.; Sabina, J.; Zuker, M.; Turner, D. H. *J. Mol. Biol.* **1999**, *288*, 911–940.

(12) Mathews, D. H.; Disney, M. D.; Childs, J. L.; Schroeder, S. J.; Zuker, M.; Turner, D. H. *Proc. Natl. Acad. Sci. U. S. A.* **2004**, *101*, 7287–7292.

(13) Mathews, D. H.; Turner, D. H. *Curr. Opin. Struct. Biol.* **2006**, *16*, 270–278.

(14) Farès, C.; Amata, I.; Carlomagno, T. *J. Am. Chem. Soc.* **2007**, *129*, 15814–15823.

(15) Ohlenschläger, O.; Haumann, S.; Ramachandran, R.; Görlach, M. *J. Biomol. NMR* **2008**, *42*, 139–142.

(16) Blad, H.; Reiter, N. J.; Abildgaard, F.; Markley, J. L.; Butcher, S. E. *J. Mol. Biol.* **2005**, *353*, 540–555.

(17) Zhao, B.; Zhang, Q. *Curr. Opin. Struct. Biol.* **2015**, *30*, 134–146.

(18) Frank, A. T.; Law, S. M.; Brooks III, C. L. *J. Phys. Chem. B* **2014**, *118*, 12168–12175.

(19) Aeschbacher, T.; Schubert, M.; Allain, F. H.-T. *J. Biomol. NMR* **2012**, *52*, 179–190.

(20) Lu, X.-J.; Olson, W. K. *Nat. Protoc.* **2008**, *3*, 1213.

(21) Lu, X.-J.; Bussemaker, H. J.; Olson, W. K. *Nucleic Acids Res.* **2015**, *43*, e142–e142.

(22) Buuren, S. v.; Groothuis-Oudshoorn, K. *Journal of Statistical Software* **2010**, 1–68.

(23) Schneider, G.; Wrede, P. *Prog. Biophys. Mol. Biol.* **1998**, *70*, 175–222.

(24) Chollet, F. Keras., `https://github.com/fchollet/keras`, 2015.

(25) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems., Software available from tensorflow.org, 2015.

(26) Rognes, T.; Flouri, T.; Nichols, B.; Quince, C.; Mahé, F. *PeerJ* **2016**, *4*, e2584.

(27) Merino, E. J.; Wilkinson, K. A.; Coughlan, J. L.; Weeks, K. M. *J. Am. Chem. Soc.* **2005**, *127*, 4223–4231.

(28) Wilkinson, K. A.; Merino, E. J.; Weeks, K. M. *Nat. Protoc.* **2006**, *1*, 1610.

(29) Deigan, K. E.; Li, T. W.; Mathews, D. H.; Weeks, K. M. *Proc. Natl. Acad. Sci. U. S. A.* **2009**, *106*, 97–102.

(30) Low, J. T.; Weeks, K. M. *Methods* **2010**, *52*, 150–158.

(31) Bellaousov, S.; Reuter, J. S.; Seetin, M. G.; Mathews, D. H. *Nucleic Acids Res.* **2013**, *41*, W471–W474.

(32) Lu, Z. J.; Gloor, J. W.; Mathews, D. H. *RNA* **2009**, *15*, 1805–1813.

(33) Bellaousov, S.; Mathews, D. H. *RNA* **2010**, *16*, 1870–1880.

(34) Hajdin, C. E.; Bellaousov, S.; Huggins, W.; Leonard, C. W.; Mathews, D. H.; Weeks, K. M. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110*, 5498–5503.

(35) Zou, Q.; Xie, S.; Lin, Z.; Wu, M.; Ju, Y. *Big Data Research* **2016**, *5*, 2–8.

(36) Zhao, B.; Guffy, S. L.; Williams, B.; Zhang, Q. *Nat. Chem. Biol.* **2017**, *13*, 968.

(37) Marcheschi, R. J.; Staple, D. W.; Butcher, S. E. *J. Mol. Biol.* **2007**, *373*, 652–663.

(38) Donghi, D.; Pechlaner, M.; Finazzo, C.; Knobloch, B.; Sigel, R. K. *Nucleic Acids Res.* **2012**, *41*, 2489–2504.

(39) Chen, Y.; Zubovic, L.; Yang, F.; Godin, K.; Pavelitz, T.; Castellanos, J.; Macchi, P.; Varani, G. *Nucleic Acids Res.* **2016**, *44*, 4381–4395.

(40) Zhao, B.; Hansen, A. L.; Zhang, Q. *J. Am. Chem. Soc.* **2013**, *136*, 20–23.

(41) Wu, M.; Tinoco, I. *Proc. Natl. Acad. Sci. U. S. A.* **1998**, *95*, 11555–11560.

(42) Liu, Y.; Holmstrom, E.; Zhang, J.; Yu, P.; Wang, J.; Dyba, M. A.; Chen, D.; Ying, J.; Lockett, S.; Nesbitt, D. J., et al. *Nature* **2015**, *522*, 368–372.

(43) Dethoff, E. A.; Petzold, K.; Chugh, J.; Casiano-Negroni, A.; Al-Hashimi, H. M. *Nature* **2012**, *491*, 724.

CHAPTER III

Probabilistic Modeling of RNA Structure Ensembles Using NMR Chemical Shifts

In the previous chapter, I have developed a framework for conditional prediction of RNA secondary structures with available experimental data like NMR derived chemical shifts. This framework enabled us to model the distinct conformational states of miR-20b RNA and fluoride riboswitch RNA based on chemical shifts associated with each conformational state.

In this chapter, I will discuss an alternative way of modeling RNA secondary structures. As mentioned, one may generate secondary structure models using free energy minimization available in many RNA modeling programs. However, the identification of the "best" structure model among low energy structure models can be challenging. In this chapter, I have used probabilistic modeling, or Bayesian/maximum entropy (BME) approach, to be more specific, to reweight structural ensemble using experimental data like chemical shifts, and identify the structure model that best agrees with available experimental data. Our results indicate that chemical shifts have the resolving power to separate native-like structures from non-native structures.

3.1 Introduction

The determination of RNA structures has always been a challenging task due to the dynamic nature of RNA, especially when sparsely populated transient states are involved. Frequently used structure determination techniques like X-ray and NMR provide ensemble-averaged observations and cannot be used to study biological interactions that are faster than the measuring time.[1] Recent advances in NMR, such as relaxation dispersion and saturation transfer, made it possible to study these previously "invisible" but functionally important conformational states.[2,3] For example, with techniques like CEST NMR, it is now possible to access chemical shifts (although sparse) for some RNA transient states.

On the other hand, computational methods like RNAstructure (for 2D structure prediction)[4] and FARFAR (for 3D structure prediction)[5] has been successful in modeling RNA secondary and tertiary structures. Molecular dynamics (MD) simulations can also be used to study biological interactions associated with RNA, such as the ligand unbinding process.[6] However, the inaccuracies of the physics and chemistry principles used to guide the simulations may lead to the disagreement between experimental measurements and simulated data.

Sometimes neither experimental techniques nor computational tools alone could generate satisfactory structure models that are capable of describing all related biological properties and functions of an RNA. Thus, it is natural to combine these two and develop a method that could lead to structure(s) that satisfy experimental observations as much as possible.

3.2 Bayesian/maximum entropy

There are two ways of combining experimental observations with simulated data. The first method is to use experimental data as restraints during simulation, similar

to how we implemented the *CS-Fold* framework discussed in `Chapter II` in which base pairing status predictions were used as restraints during the secondary structure folding. The other method is to generate a structural ensemble of low energy structure models and then reweight them to match experimental data. The advantage of the second method is that one can use different tools to back-calculate experimental data, or perform simulations, thus leading to a more accurate result; moreover, it would be computationally expensive to include experimental restraints during complex simulation.

Maximum entropy is one of the major methods that can be used for combing experimental measurements and simulated data. The central idea is to optimize the weights assigned to each member in the initial simulated structural ensemble (*a priori*) so that the agreement between the reweighted ensemble-averaged properties and the experimental observations can be maximized. Tools developed based on maximum entropy reweighting have been successfully applied to studying protein structural ensembles. For example, the ENSEMBLE program has been implemented to study the unfolded state of the N-terminal SH3 domain of drk. It also revealed that the unfolded ensemble is more compact than previously thought, with many native-like contacts.[7] The maximum entropy principle has also been applied to RNA structure determination and force field refinement. For example, MD simulation was combined with measurements from solution NMR experiments such as ^{3}J scalar couplings and NOE distances to improve the Amber force field used for RNA tertiary structure modeling and study the conformational ensemble of RNA tetranucleotides.[8,9]

Here in this chapter, we applied a Bayesian/maximum entropy (BME)[10] approach where the error or uncertainty of the experimental data is also taken into account. The goal of BME is to find a new distribution of conformations, or a set of weights that can be assigned to each member in the structural ensemble, so that the reweighted ensemble will maximally agree with the experimental data. Based on this, the new

distribution P should satisfy the following conditions:[10]

- the relative entropy between the new distribution P and the initial distribution P^0 is maximized:

$$S_{rel}(P\|P^0) = D_{KL}(P\|P^0) \int dx P(x) ln[\frac{P(x)}{P^0(x)}]; \tag{3.1}$$

- the ensemble-averaged, back-calculated experimental quantities should agree with the measured quantities within a tolerance:

$$< F_i^{pred} + \epsilon_i >= F_i^{exp}, \tag{3.2}$$

  where $i = 1, ..., m$, m is the total number of experimental observables, and ϵ_i is the error or uncertainty of the ith measurement;

- the new distribution P is normalized.

From previous studies,[11–13] it can be shown that the optimal weights minimize the following function:

$$\mathcal{L}(w_1 \dots w_n) = \frac{m}{2}\chi^2(w_1 \dots w_n) - \theta S_{rel}(w_1 \dots w_n), \tag{3.3}$$

in which:

$$\chi^2(w_1 \dots w_n) = \frac{1}{m}\sum_i^m \frac{(\sum_j^n w_j F(x_j) - F_i^{exp})^2}{\sigma_i^2} \tag{3.4}$$

$$S_{rel} = -\sum_i^n w_j ln(\frac{w_j}{w_j^0}). \tag{3.5}$$

Here, w_j and w_j^0 is the new and initial weight of the jth member in the ensemble, n is the population of the ensemble, σ_i is the uncertainty of the measurement F_i^{exp}, and $F(x_j)$ is the back-calculated property from the jth member in the ensemble.

In the loss function $\mathcal{L}$ (Eq. 3.3), the first term (χ^2) describes the agreement between the experimental data and the back-calculated properties from structure models and the second term (S_{rel}) describes the deviation of the new weights from the initial weights. In our case, the initial weights should be $1/n$ if there are n members in the ensemble.

It was shown that the optimal weights could also be calculated through *Maximum A Posteriori* (MAP) estimation[9] by minimizing the negative log-likelihood of the posterior distribution. This method is called Bayesian ensemble refinement, and it is mathematically equivalent to the Maximum entropy with error or the BME approach described above in terms of our application.

3.3 Probabilistic modeling of RNA secondary structures

Current computational programs for RNA secondary structure prediction are largely focused on free energy minimization[4,14–17] in which free energy of RNA motifs are evaluated using a set of nearest neighbor parameters which can be measured with optical melting experiments.[18] Some programs, like *RNAstructure*, allows the incorporation of chemical mapping data, such as SHAPE reactivities,[19,20] to more accurately predict the secondary structures.

When the primary sequence for an RNA is available, programs such as *AllSub*[21,22] (within the RNAstructure modeling suite), or *MC-Fold*,[23] could generate possible low free energy structures, including the lowest energy structure and sub-optimal structures, for a given sequence.

The goal of this chapter was to develop a method to identify the "best" structure model from the structural ensemble. Obviously, it would be impossible to do so if no other information is available. However, with experimental data, such as NMR chemical shifts, we believe that it is possible to identify the structure that is most consistent with the experimental data using inherent structural information. In

this section, I have applied Bayesian/maximum entropy (BME) method (explained in `Section 3.2`) to RNA secondary structure prediction in which simulated data are the structural ensemble generated from programs like *AllSub* and *MC-Fold*, and experimental observables are NMR chemical shifts associated with the target RNA.

3.3.1 SS2CS: Predicting chemical shifts from RNA secondary structures

To use BME to reweight structural ensemble and to identify the "best" structure model, a method is required to predict or back-calculate NMR chemical shifts from a given RNA secondary structure. Thus, in this chapter, I have explored the simulation of NMR chemical shifts, directly from secondary structure models of RNA. If we can reproduce NMR chemical shifts from the 2D model, then we may be able to determine the secondary structure of RNAs by generating models, simulating their chemical shifts, and then identifying the most consistent model with experimental NMR chemical shifts. Here, we created a tool which we referred to as, **S**econdary **S**tructure to **C**hemical **S**hifts (**SS2CS**), that could take RNA secondary structure as input and output the predicted chemical shifts for different nucleus types.

The machine learning model I applied to develop SS2CS is random forest, a supervised ensemble learning method. Our testing results show that our tool could predict carbon and proton chemical shifts with high accuracy: the mean absolute errors (MAEs) were 0.84 ppm for carbon and 0.11 ppm for proton, respectively. The chemical shifts prediction accuracy of SS2CS is about the same level as when tertiary structures are used as input in other programs.[24]

3.3.1.1 Data sets and featurization

For 108 RNAs (the same RNAs we used for training CS2BPS in `Chapter II`), the secondary structures were retrieved using the program DSSR from the 3DNA suite.[25] The output secondary structure, a '.ct' file, contains for each nucleotide, or

each residue, the residue name, residue id, residue ids before and after the current residue, and the residue id that the current residue is base paired to. It should be noted that the secondary structure file output from DSSR only contains canonical base pairing interactions.

The NMR chemical shifts were downloaded from the Biological Magnetic Resonance Data Bank (BMRB: `http://www.bmrb.wisc.edu/`). We applied the same protocol as discussed in `Chapter II` to correct (if necessary) ^{13}C data for each RNA because ^{13}C chemical shifts often contain systematic referencing errors.[26] For 5 RNAs in our data set, whose PDBIDs are 1R7Z, 1R7W, 1Z30, 2LK3, and 2LU0, we found different measured chemical shifts for the same atom. Thus, these 5 RNAs were removed from the training set.

In order to predict the nonexchangeable chemical shifts of, namely, C1', C2', C3', C4', C5', C2, C5, C6, C8, H1', H2', H3', H4', H2, H5, H5', H5", H6, and H8, we first constructed data set for individual nucleus types. Briefly, for each nucleus type, the chemical shifts associated with this nucleus type from all the RNAs, along with the secondary structure *features* of each residue, were combined into a large data set. The secondary structural features we encoded from the input structure file include (for residue i):

- length of the RNA
- residue type of residues i, $i-1$, and $i+1$
- residue type of residue i's pairing partner j, if exists
- residue types of the pairing partner of residues $i-1$ and $i+1$, if exist
- residue types of residues $j-1$ and $j+1$, if exist
- residue types of the pairing partner of residues $j-1$ and $j+1$, if exist

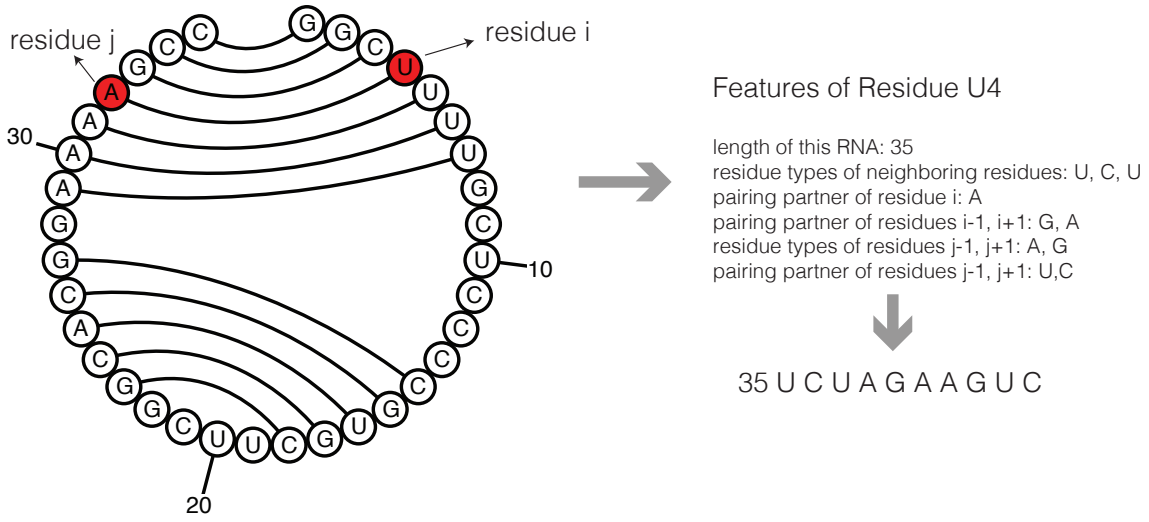

Figure 3.1: Feature extraction example for the human telomerase RNA (PDBID: 2L3E) in SS2CS.

For example, we show the extracted secondary structural features for the human telomerase RNA (PDBID: 2L3E) in Figure 3.1. For the highlighted residue i, U4, the features should be:

$$35 \quad U \quad C \quad U \quad A \quad G \quad A \quad A \quad G \quad U \quad C.$$

We repeated this featurization for all nucleus types and constructed individual training sets. The sample size of these data sets are shown below in Table 3.1.

Table 3.1: Sample size of individual nucleus types

Nucleus type	Sample size	Nucleus type	Sample size	Nucleus type	Sample size
C1’	1880	H1’	3152	C6	1022
C2’	1655	H2’	2961	H5	1475
C3’	1547	H3’	2509	C8	1109
C4’	1515	H4’	2069	H6	1545
C5’	1271	H5’	1662	H8	1644
C2	583	H5”	1631	H2	706
C5	997				

For most of the nucleus types, the sample size is around 1000 to 2000. But for C2, C5, and H2, we only have fewer than 1000 samples.

3.3.1.2 Model selection

Next, I constructed a machine learning pipeline that takes as input the secondary structure features for each residue, and outputs the predicted chemical shifts for individual nucleus types for that residue. To do that, I have trained separate machine learning models for each nucleus type. For model selection, I have tested six classic regression models, namely, linear regression, ridge regression, support vector machine regression, random forest regression, extra randomized tree regression, and gradient boosting regression. *Linear regression* uses a simple linear model to fit parameters of features and makes predictions; *ridge regression* is a linear model with an extra $L2$ regularization term to prevent *overfitting* problems; *support vector regression* (with radial basis function (rbf) kernel) trains a model that achieves maximal flatness while restraining all data points to be within an error margin; *random forest*, as discussed in `Introduction`, builds an ensemble of decision trees to avoid *overfitting*; *extra randomized trees*, similar to random forest, but adds another layer of randomness by randomly selecting a splitting threshold at each node; and finally, *gradient boosting regression*, which is also an ensemble learning technique based on decision trees, uses boosting to optimize weak learners.

The model selection was performed on the base models, meaning that we used all default parameters and did not do any hyperparameter tuning. Each secondary structure–chemical shifts data set was randomly split into training and testing sets with a 80%/20% ratio. A 10-fold cross validation was then performed on the training set to calculate train and validation scores. Figure 3.2 shows the learning curves of random forest regression and linear regression models for H1' and C1' chemical shifts prediction.

Plotting learning curve is an important procedure when training a machine learning model. It tells us how the model learns with more experience (either by training time or the number of trained samples), through the training score and validation

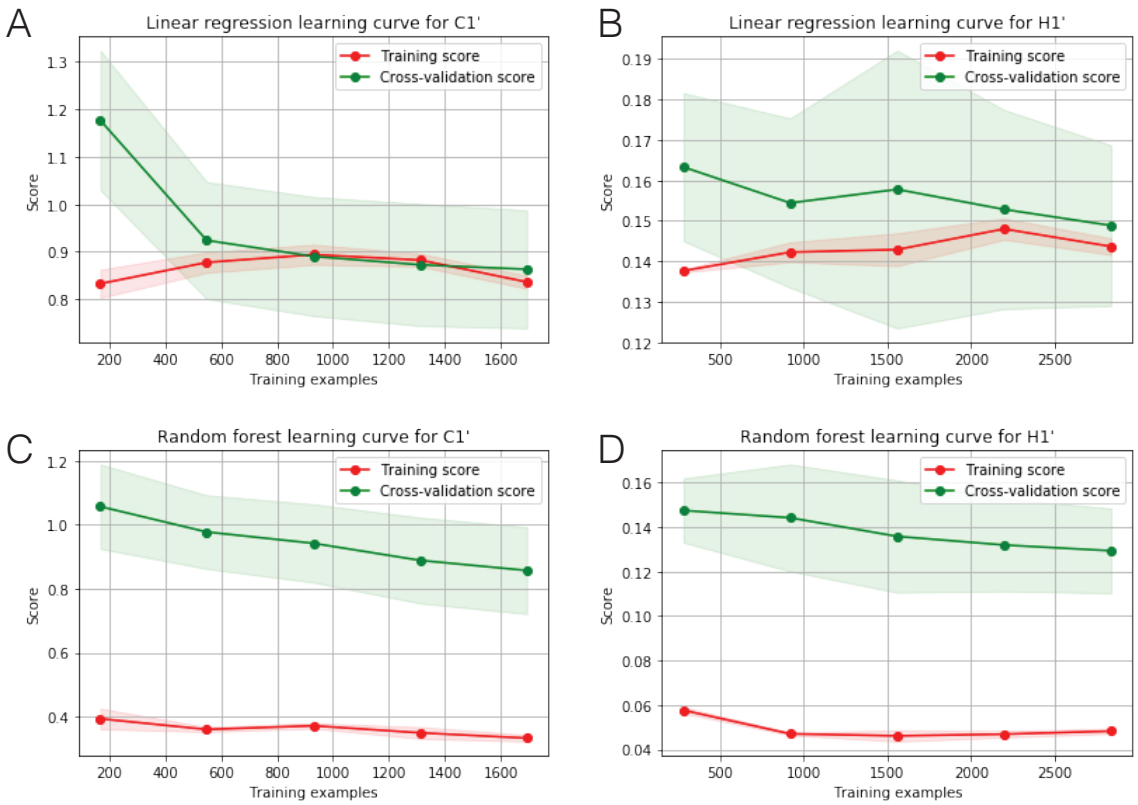

Figure 3.2: Learning curves of 10-fold cross validation. (A-B) Learning curves of training mean absolute error (MAE) and validation MAE for linear regression model when predicting C1' and H1' chemical shifts, respectively. (C-D) Learning curves of training MAE and validation MAE for random forest model when predicting C1' and H1' chemical shifts, respectively.

score. For example, Figure 3.2 shows the learning curves of random forest regression and linear regression models for H1' and C1' chemical shifts prediction. As is shown in Figure 3.2, when predicting C1' chemical shifts, random forest exhibited lower training error and validation error compared to linear regression (Figure 3.2A and C). Error metric used here is the MAE between measured and predicted chemical shifts. Similarly, for H1' chemical shifts prediction, random forest also exhibited lower training error and validation error compared to linear regression (Figure 3.2B and D). It is worth mentioning that when predicting C1' chemical shifts with random forest, both the training error and validation error showed a decreasing trend. This is probably due to small sample sizes of the H1' and C1' chemical shifts, which were

3152 and 1880, respectively (See Table 3.1). If we could have more training data for C1', we may be able to improve the validation score further.

To compare the performance of different regression models, we next plotted the validation MAE for carbon and proton predictions. As shown in Figure 3.3, random

Figure 3.3: Cross validation error. (A) is the cross validation MAE for proton chemical shifts prediction using different models. (B) is the cross validation MAE for carbon chemical shifts prediction.

forest model (*purple line*) outperformed other six models when predicting proton chemical shifts (Figure 3.3A), exhibiting the lowest or one of the lowest MAE for all proton nuclei. Similarly, for predicting carbon chemical shifts, random forest also exhibited the lowest MAE for most carbon nuclei. The mean validation MAE for proton and carbon prediction were 0.12 ppm and 0.91 ppm, respectively (Table B.9), both are the lowest among six different regression models. Based on the cross validation result, I have selected random forest as our SS2CS model for predicting carbon and proton chemical shifts from secondary structures.

Next, I tested the random forest model on the 20% left-out testing set and cal-

culated the prediction MAE between the predicted chemical shifts and measured chemical shifts. As shown in Table 3.2, random forest model (without any hyperparameter tuning) exhibited an MAE of 0.84 ppm for carbon and 0.11 ppm for proton chemical shifts prediction.

Table 3.2: Testing set MAE when using random forest model

Nucleus type	Test MAE	Nucleus type	Test MAE
C1'	0.87	H1'	0.11
C2'	0.52	H2'	0.12
C3'	1.13	H3'	0.11
C4'	0.72	H4'	0.08
C5'	0.92	H5'	0.14
C2	1.01	H5"	0.11
C5	0.67	H2	0.17
C6	0.86	H5	0.10
C8	0.87	H6	0.09
		H8	0.12
Mean	0.84	Mean	0.11

Currently, in this project, I have not performed any hyperparameter tuning yet; that is, the model uses all default parameters from the Python *scikit-learn*[27] package.

3.3.2 Secondary structure reweighting

3.3.2.1 Method

Now that with SS2CS, we could extract structural features from a given secondary structure model and predict its nonexchangeable chemical shifts. Next, we examined whether SS2CS can be combined with Bayesian/maximum entropy (BME) to reweight a set of structural ensembles using a data set of 16 RNAs.

We first created an ensemble of low energy secondary structure decoys for a given RNA sequence using tool *MC-Fold* from the MC-Sym suite[23] (as it allows the formation of pseudoknotted structures). However, for large RNAs, it may take a long

time to generate decoys using *MC-Fold*, so we used *AllSub* from the RNAstructure modeling suite instead for the largest RNA in our data set, the HIV-1 RNA (PDBID: 2N1Q), which has 155 nts. Using *MC-Fold*, we were able to get 10 different decoys whose folding free energies are within 30% of the lowest energy structure. The exception was the fluoride riboswitch for which we combined decoys generated from *AllSub* and *MC-Fold* together in order to have a diverse structural pool. The decoy structures were then combined with the DSSR-derived native structure for further analysis.

Next, for each valid decoy structure, SS2CS was used to extract structural features and predict chemical shifts for all nonexchangeable nuclei. In order to avoid *overfitting*, we checked and removed (if necessary) chemical shifts associated with the "twins" of the testing RNA from the training set, that are, RNAs whose sequences were similar to the testing RNA.

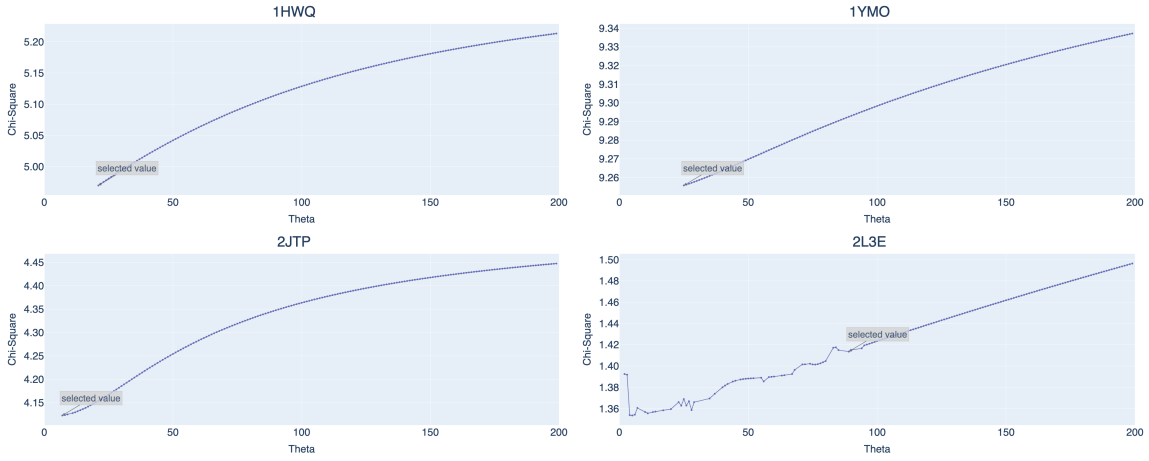

Figure 3.4: The relationship between χ^2 and θ. We scanned the value of θ from 1.0 to 200.0 with a step of 1.0 and calculated corresponding χ^2 using a reweighted ensemble. Then, to select the best θ, we started from 200.0 and chose the smallest θ until the increasing trend did not exist.

After generating chemical shifts predictions using SS2CS, we then applied BME to the experimental and predicted chemical shifts and assigned weights to individual decoys in the structural ensemble. According to Eq. 3.3, θ is a global scaling factor

that controls the relative contribution of the entropy terms in the overall loss function $\mathcal{L}$. It reflects the trade-off between two terms: (1) χ^2, which is the agreement between experimental data and predicted chemical shifts; (2) S_{rel}, which is the deviation of the new distribution from the original uniform distribution. To find the best θ, we scanned different values from 1.0 to 200.0 (with a step of 1.0) and calculated χ^2 (using Eq. 3.4) at different θ.[10] Theoretically, the smaller θ is, the more $\mathcal{L}$ is dependent on χ^2, and the better agreement we should be able to achieve between experimental reweighted ensemble-averaged chemical shifts.

Table 3.3: Optimized θ and corresponding χ^2 for 16 RNAs in our data set

PDBID	θ	χ^2	PDBID	θ	χ^2
1HWQ	21	4.97	1YMO	25	9.26
2JTP	7	4.12	2L3E	89	1.41
2LUB	6	1.69	2N6X	3	1.10
2N7X	11	1.87	2N82	40	5.95
2NBY	16	2.15	2NC0	23	2.90
5KH8	13	4.73	5KMZ	37	7.30
5V16	3	1.71	6GZK	106	15.94
2N1Q	78	2.37	2LU0	29	2.65

But in reality, we found that for some RNAs, the relationship between θ and χ^2 may not be positively correlated. To set a standard for selecting θ, we then started from 200.0 and chose the smallest θ, after which the positive correlation did not exist. For example, in Figure 3.4, we show how χ^2 changed when we scanned different θ for four RNA examples. The plots of 1HWQ, 1YMO, and 2JTP behaved like we expected, exhibiting a positive correlation between χ^2 and θ. For RNAs like these, the optimized θ was the smallest value that the program could converge. Note that the plot may not start from $\theta = 1.0$ because small θ sometime failed to converge. However, for 2L3E, the plot when θ was small was unstable. We then selected the smallest θ (89.0), after which the plot started to increase monotonically. We reported

the optimized θ and corresponding χ^2 values in Table 3.3. The relationship between χ^2 and θ for the remaining RNAs are included in Figure B.3 in `Appendix`.

There are some potential problems for using a small θ. The initial weights may be distorted a lot, with a few members assigned the majority of the weights. When applying BME to combine MD simulation and experimental data, this may cause significant statistical errors[8] since one would want all of the frames from the simulation to contribute to the ensemble. However, since our goal was not necessarily combining input from all decoys, a small θ would help us to identify the "best" structure model in an ensemble.

We then looked at the optimized θ and the corresponding χ^2 values for each RNA ensemble (Table 3.3). As discussed, χ^2 measures the difference between experimental and predicted chemical shifts (after reweighting). Four RNAs (PDBIDs: 1YMO, 2N82, 5KMZ, 6GZK) exhibited a χ^2 value that was larger than 5.0. There are multiple reasons why some RNAs exhibited very large χ^2. For example, the decoy ensemble may not be diverse enough, or most of the decoys are very different from the native structure, so it is impossible to achieve a good agreement between predicted and measured chemical shifts no matter how BME reweight them. This may be the case for 2N82 and 6GZK. Most of the decoys for 2N82 exhibited very low PPV compared to the native structure, and most of the decoys for 6GZK exhibited both very low TPR and PPV. Another possible reason is that either the predicted chemical shifts or the experimental chemical shifts are not accurate enough. The other two RNAs whose χ^2 values were very large, 1YMO and 5KMZ, are both pseudoknots. It is possible that our SS2CS predictor is not very good at predicting chemical shifts at pseudoknotted regions due to insufficient featurization, or the experimental chemical shifts are either not accurate or not enough for the BME to get a good agreement by reweighting.

3.3.2.2 Overall result

We reported the overall results of BME reweighting in Table 3.4. In general, among the 16 RNA secondary structure ensembles we studied, BME was able to recover the DSSR-derived native structure for 7 RNAs, in which cases BME assigned the highest weight to the DSSR-derived structure. For structures that were assigned the highest BME weight, the TPV and PPV values were both larger than 0.80 for 13 structures. This indicates that in the cases BME was not able to recover the DSSR-derived native structure it still selected a structure similar to the DSSR-derived structure. Finally, I further examined the 3D structures of the members that were assigned the highest BME weight and found that for 9 structures, BME was able to identify the "best" structure, meaning that BME either selected the DSSR-derived native structure directly, or it selected a structure that contained noncanonical interactions that were not highlighted by DSSR.

Table 3.4: Summary of the BME reweighting of 16 RNAs

PDBIDs	Size of ensemble	BME identified the DSSR Structure	TPV>0.80 PPV>0.80	BME identified the "best" structure[1]
1HWQ	11	✓	✓	✓
1YMO	11		✓	
2JTP	11		✓	
2L3E	11		✓	
2LUB	11		✓	✓
2N6X	11	✓	✓	✓
2N7X	11			✓
2N82	11	✓	✓	✓
2NBY	11	✓	✓	✓
2NC0	11	✓	✓	✓
5KH8	16	✓	✓	✓
5KMZ	11			
5V16	11		✓	
6GZK	11			
2N1Q	11		✓	
2LU0	11	✓	✓	✓

[1] Here, the "best" structure means the BME selected structure contained noncanonical interactions that were not highlighted in the DSSR-derive native structure.

In the following sections, I will discuss some representative examples.

3.3.2.3 Representative example 1: the group II intron ai5γ RNA

We first studied the group II intron ai5γ RNA (PDBID: 2LU0), with 49 nts. As

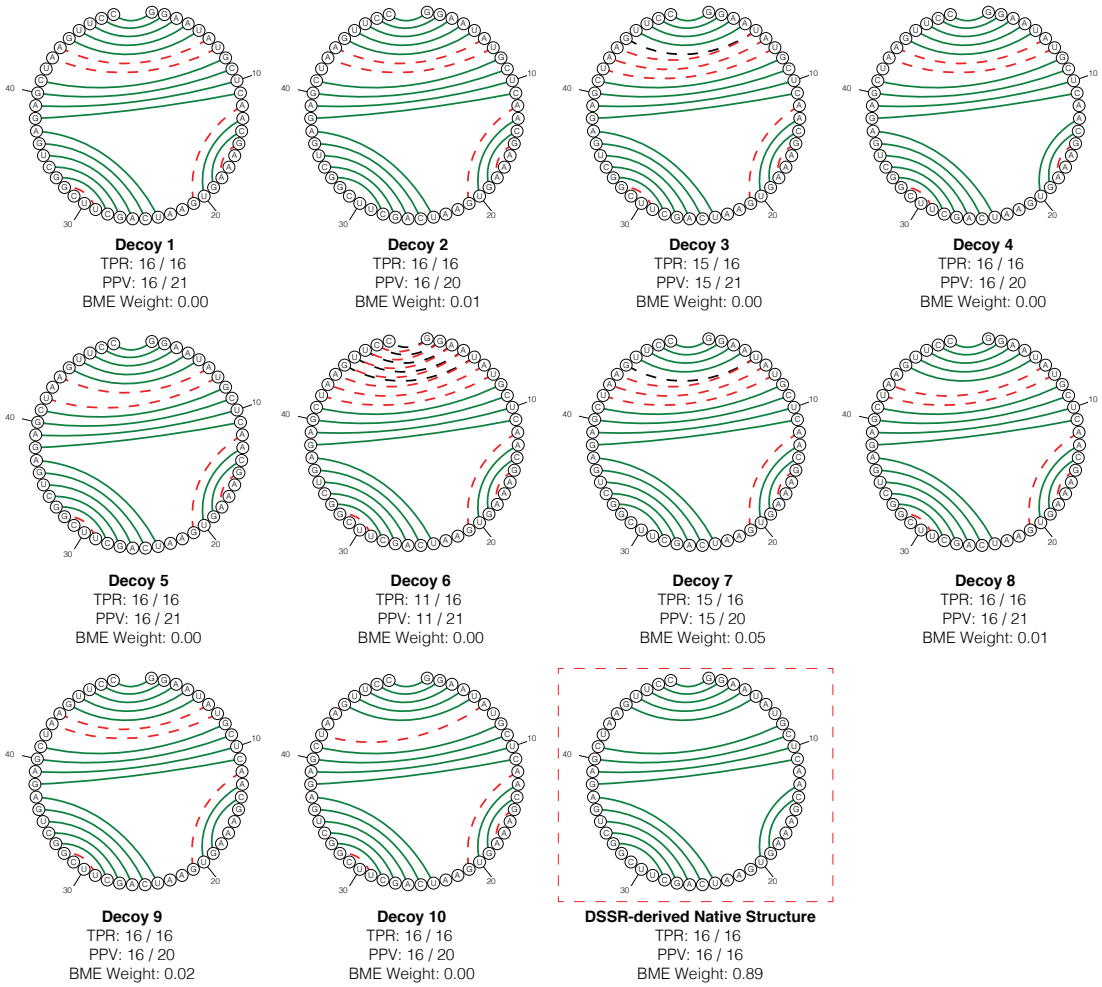

Figure 3.5: Low energy secondary structure models of the group II intron ai5γ RNA (PDBID: 2LU0). The first ten structures were generated by *MC-Fold*. The last structure (in the red box) was the DSSR-derived native structure. Green lines indicate correctly predicted base pairs; red dashed lines represent extraneous base pairs; black dashed lines represent base pairs missing from the predicted structure.

shown in Figure 3.5, the last structure, shown in the red box, was the DSSR generated native structure derived from solution NMR. The first 10 structures (Decoy 1-10) were low energy secondary structure models generated by *MC-Fold*, exhibiting different

perturbations in base pairs from the DSSR-derived structure. We calculated the sensitivity or true positive rate (TPR) and positive predicted value (PPV) between each decoy and the native structure. For example, for Decoy 1, all of the 16 base pairs in the native structure were correctly recovered; and 16 out of 21 predicted base pairs were present in the native structure.

Intuitively, we first examined the error between the predicted chemical shifts and the measured chemical shifts. For each decoy, we calculated the mean absolute error (MAE), the root mean squared error (RMSE), and the Kendall τ coefficient (τ) between the measured and SS2CS predicted chemical shifts.

Table 3.5: Chemical shift error analysis for 2LU0

Structure	TPR	PPV	MAE	RMSE	τ	BME weight
Decoy 1	1.00	0.76	0.53	1.16	0.92	0.00
Decoy 2	1.00	0.80	0.46	1.06	0.93	0.01
Decoy 3	0.94	0.71	0.56	1.25	0.92	0.00
Decoy 4	1.00	0.80	0.52	1.18	0.92	0.00
Decoy 5	1.00	0.76	0.54	1.25	0.92	0.00
Decoy 6	0.69	0.52	0.59	1.30	0.92	0.00
Decoy 7	0.94	0.75	0.50	1.16	0.93	0.05
Decoy 8	1.00	0.76	0.53	1.21	0.92	0.01
Decoy 9	1.00	0.80	0.51	1.08	0.92	0.02
Decoy 10	1.00	0.80	0.51	1.14	0.92	0.00
DSSR Structure	1.00	1.00	0.41	0.96	0.94	0.89

According to Table 3.5, the DSSR-derived native structure exhibited the lowest MAE and RMSE, and the highest Kendall τ coefficient. This result indicates that the difference between measured and predicted chemical shifts have the power to differentiate native-like secondary structure from non-native secondary structures. We then explored whether BME could be applied to solve this problem as well. We applied BME to the ensemble of these 11 structures, using SS2CS predicted chemical shifts as back-calculated properties from simulation and measured chemical shifts as experimental data, to reweight individual structures. According to Eq. 3.3, BME

aims to optimize a set of weights that are assigned to each member in the structural ensemble so that the ensemble-averaged properties can agree with the experimental observations as much as possible, while not deviating too much from initial weights. Here, the initial weights are uniformly distributed ($1/n$ where n is the population of the structures in the ensemble).

The optimized BME weights (Table 3.5) are consistent with the previous analysis, with the native structure assigned a maximal weight of 0.89. This shows that the native-like structure could be identified from a set of low energy structure models.

3.3.2.4 Representative example 2: the fluoride riboswitch

In the next example we studied the structural ensemble of the *apo* state of the fluoride riboswitch (PDBID: 5KH8) generated by both *MC-Fold* and *AllSub*. The native *apo* state adopts a pseudoknotted structure. In the structural ensemble (Figure 3.6), Decoy 1-10 were pseudoknotted structures (generated from *MC-Fold*) and Decoy 11-14 were non-pseudoknotted structures (generated from *AllSub*). The last structure, in the red box, was the DSSR-derived native secondary structure. Finally, Decoy 15, in the blue box, was generated manually by adding two extraneous long range base pairs (red dashed lines) compared to the DSSR structure: A5-U35 and A37-U45 (Figure 3.7). These two base pairs are not annotated in DSSR-derived structure since DSSR only includes canonical base pairs, but these are noncanonical long range interactions. Surprisingly, when not including Decoy 15, the DSSR-derived native structure, was assigned the highest weight among the first 15 decoys (weight = 0.45; Table 3.6). But when including Decoy 15 in the ensemble, Decoy 15 was ranked highest based on the BME weight (0.59).

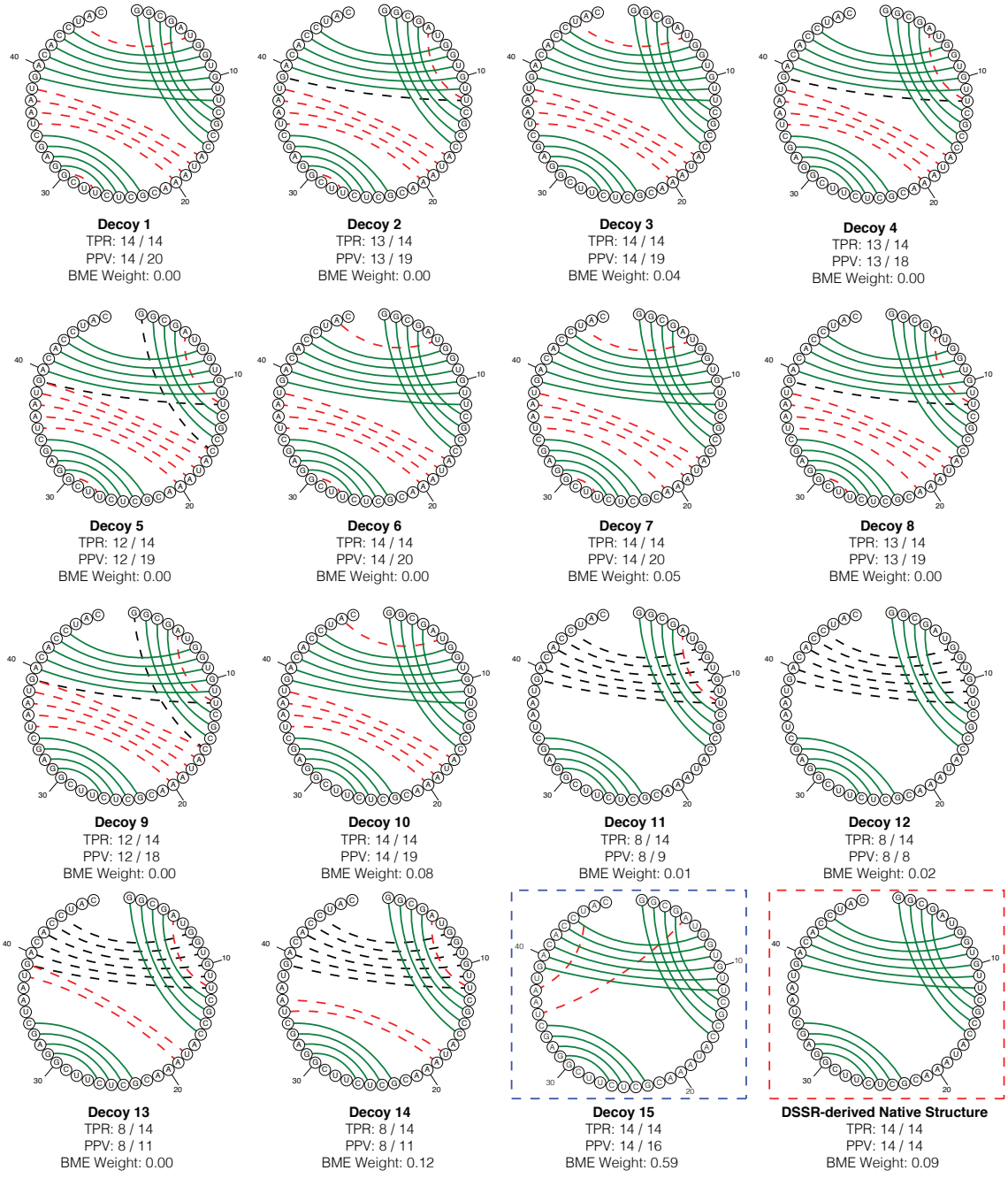

Figure 3.6: Low energy secondary structure models of the fluoride riboswitch RNA (PDBID: 5KH8).

Table 3.6: BME weights when including/not including the two long range base pairs for the fluoride riboswitch (PDBID: 5KH8)

Structure	TPR	PPV	BME weight before adding Decoy 15	BME weight after adding Decoy 15
Decoy 1	1.00	0.70	0.00	0.00
Decoy 2	0.93	0.68	0.00	0.00
Decoy 3	1.00	0.74	0.09	0.04
Decoy 4	0.93	0.72	0.00	0.00
Decoy 5	0.86	0.63	0.00	0.00
Decoy 6	1.00	0.70	0.00	0.00
Decoy 7	1.00	0.70	0.07	0.05
Decoy 8	0.93	0.68	0.00	0.00
Decoy 9	0.86	0.67	0.00	0.00
Decoy 10	1.00	0.74	0.16	0.08
Decoy 11	0.57	0.89	0.03	0.01
Decoy 12	0.57	1.00	0.02	0.02
Decoy 13	0.57	0.73	0.01	0.00
Decoy 14	0.57	0.73	0.16	0.12
Decoy 15	1.00	0.88	N/A	**0.59**
DSSR Structure	1.00	1.00	**0.45**	0.09

This indicates that the SS2CS predicted chemical shifts contained structural information about these two long range interactions which were then affected the BME reweighting.

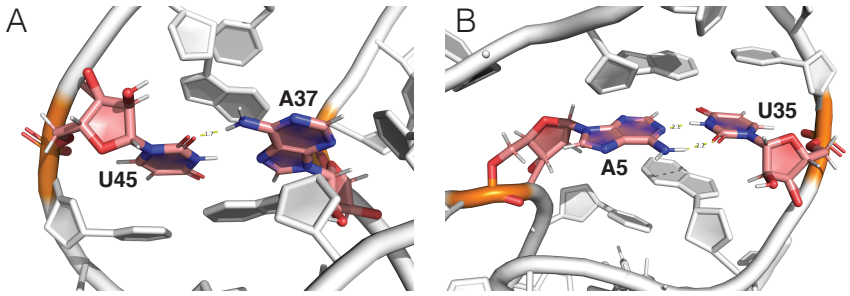

Figure 3.7: Long range base pairing interactions in the fluoride riboswitch.

In `Chapter II`, we modeled two distinct conformational states of the fluoride riboswitch: the Mg^{2+}-free state and the *apo* state (Mg^{2+}-bound state) using CS-Fold with only C1'/H1' and C8/H8 chemical shifts for guanine residues. Here, we explored

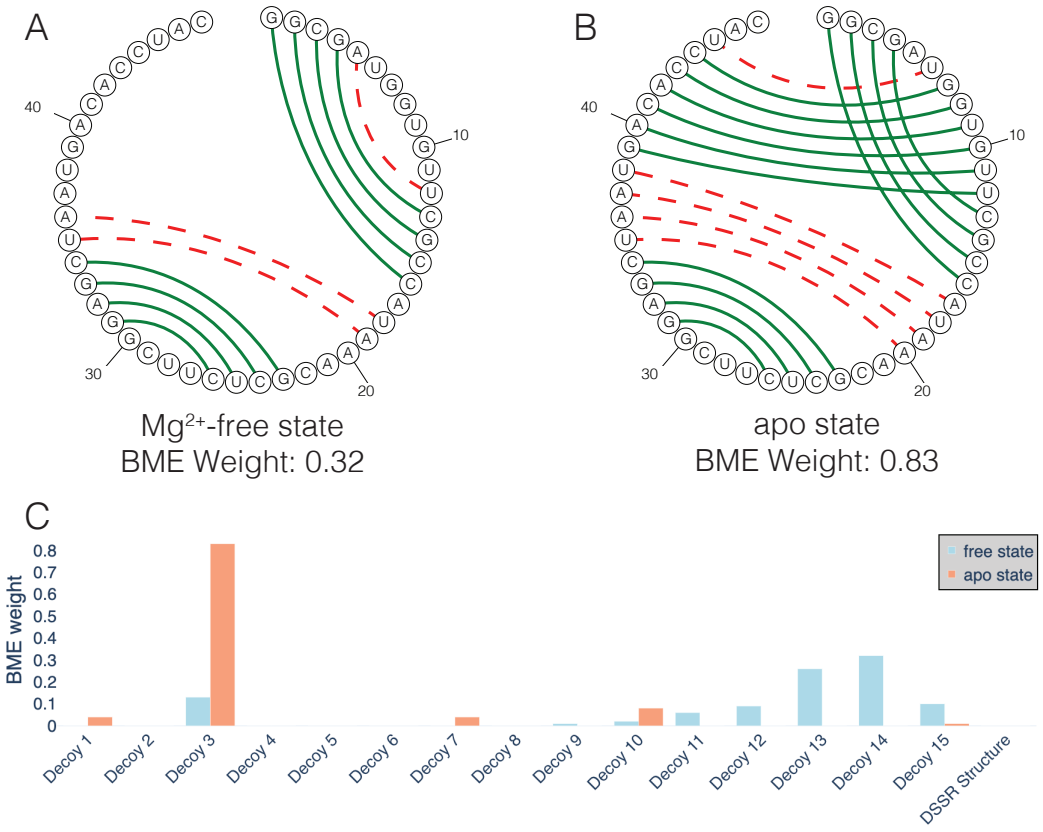

Figure 3.8: Distinct conformational states of the fluoride riboswitch. Using available chemical shifts, which are C1'/H1' and C8/H8 chemical shifts for guanine residues, BME assigned the highest weight to (A) for the Mg^{2+}-free state and (B) for the *apo* state. The detailed BME weights of each structure in the ensemble were reported in (C): *orange* bar represents the *apo* state BME assignment and *light blue* bar represents the Mg^{2+}-free state BME assignment.

whether SS2CS, along with BME, could be used to identify the native structure of the free and the *apo* state using corresponding chemical shifts of that state. The free state, unlike the *apo* state, does not have the pseudoknotted base pairs. Among the 16 structures in the ensemble, the structure that was assigned the highest weight when using the free state chemical shifts was Decoy 14, which is indeed a non-pseudoknotted structure. In Figure 3.8A, we projected the BME selected structure onto the native structure of the free state of the fluoride riboswitch. We noticed that although the selected structure had 3 extraneous base pairs compared to the native structure, the base pairing status, and the pairing partners of all guanine residues were correctly re-

covered. Since the only chemical shifts we have available were C1'/H1' and C8/H8 for guanine residues, the selected structure agreed with the available experimental data. On the other hand, for the *apo* state, when we used complete chemical shifts, Decoy 15 was identified by BME weight. And when we used partial chemical shifts, Decoy 3 (Figure 3.8B) was assigned the highest weight. Similarly, although the structure had extraneous base pairs compared to the native structure of the *apo* state, it correctly recovered the pseudoknotted base pairs and the status of all guanine residues.

Another interesting observation is that the BME weights of the *apo* state is very sparse, with Decoy 3 assigned most of the weight (*orange* bar in Figure 3.8C). However, for the free state (*light blue* bar in Figure 3.8C), the BME weights are not as sparse: 8 structures were assigned a nonzero weight and Decoy 13 was assigned a similar weight (weight=0.26) to Decoy 14 (weight=0.32). One possible explanation is that the *apo* conformational state was coordinated with Mg^{2+}, which helps the RNA fold and stabilizes a single structure. Thus, one conformation dominates the BME weights. On the other hand, the free state does not bind with Mg^{2+} and may interconvert between multiple conformations, making it more difficult for BME to identify one "best" structure.

3.3.2.5 Representative example 3: the microRNA-20b pre-element

Similarly, in `Chapter II`, we also modeled the distinct conformational states of the microRNA-20b pre-element (miR-20b) and showed that CS-Fold was able to predict the *apo* and *holo* structures with high accuracy. In terms of base pairing predictions, we found that for the *apo* state, we predicted residues U6, C17, and U18 to be base paired. By careful examination of the 3D structures, we found there were noncanonical interactions between residues U6 and U18, and between U7 and C17. Here we used the same structural ensemble and studied whether the chemical shifts of the *apo* and *holo* state could be used with BME to identify the corresponding structure models.

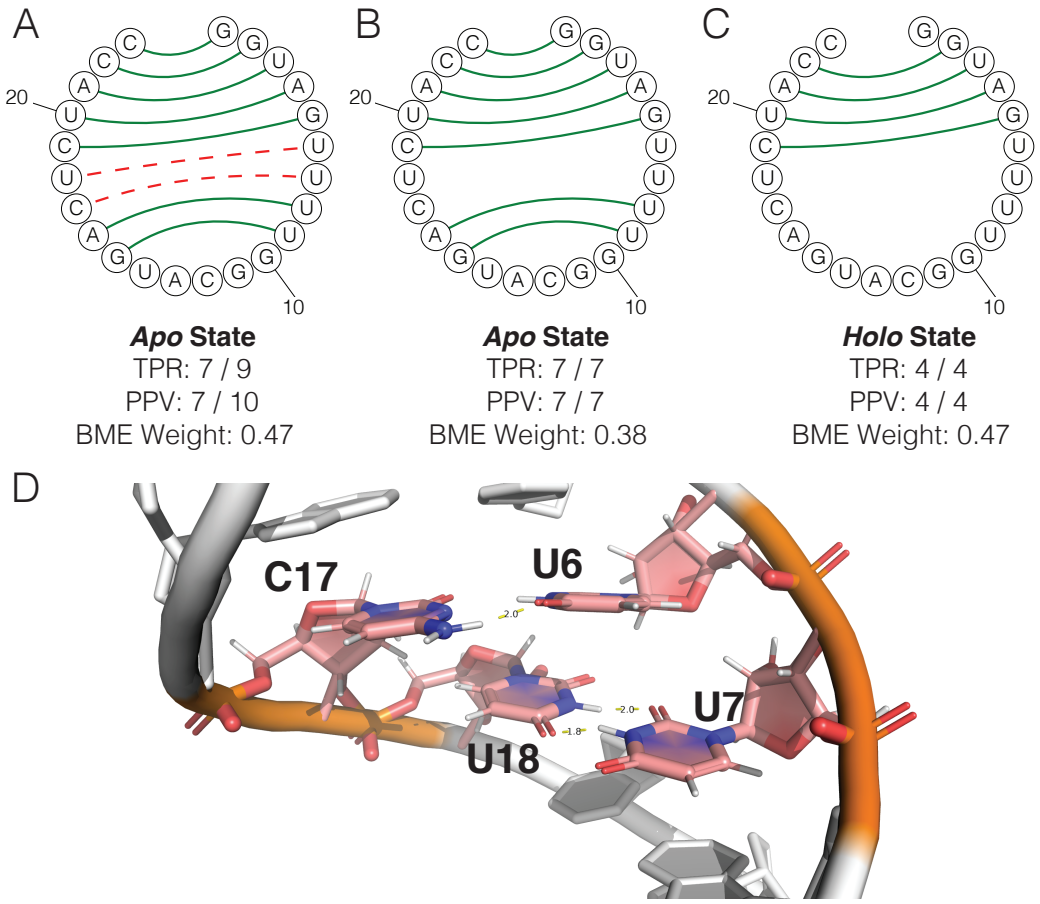

Figure 3.9: The BME selected structures for the two distinct conformational states of miR-20b RNA. (A) is the structure that was assigned the highest BME weight when using the *apo* chemical shifts; (B) is the structure with the second highest BME weight for the *apo* state; (C) is the highest weight structure when using the *holo* chemical shifts. (D) is the noncanonical interaction at residues U6-C17 and U7-U18.

Interestingly, the structure that was assigned the highest weight for the *apo* state was the decoy in Figure 3.9A, which contained two extraneous base pairs (Figure 3.9D; U6-U18 and U7-C17) compared to the DSSR structure. The DSSR structure was assigned a second highest weight (in Figure 3.9B). On the other hand, for the *holo* state, BME was able to identify the DSSR-derive native structure directly (Figure 3.9C).

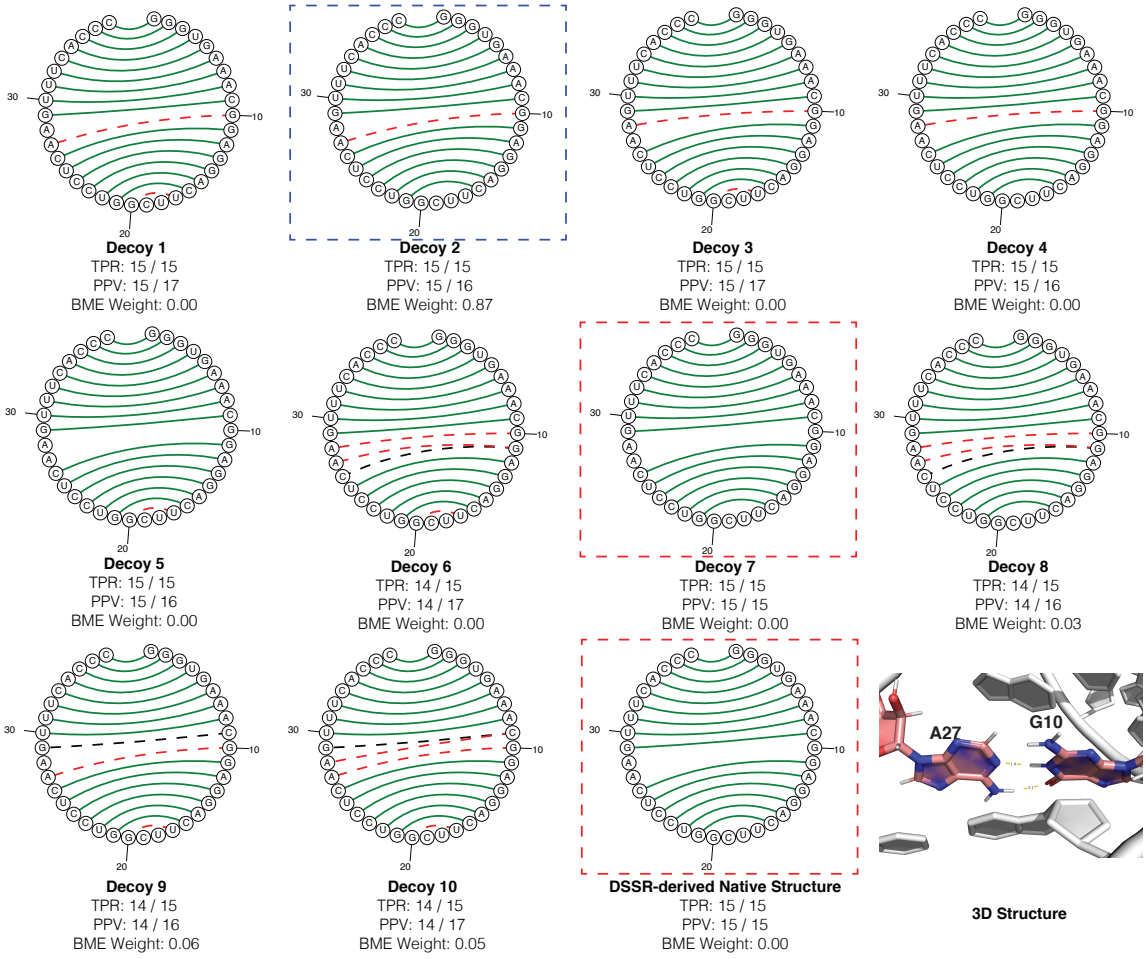

Figure 3.10: Low energy secondary structure models of the human HAR1 RNA (PDBID: 2LUB). The last figure is the 3D structure at residues A27 and G10.

3.3.2.6 Representative example 4: the human HAR1 RNA

We next looked at some examples where BME did not identify the DSSR-derived native structure. For the human HAR1 RNA (PDBID: 2LUB) (Figure 3.10), most decoys were highly similar to the native structure (in the red box), making it more difficult to differentiate the native structure from non-native structures. The structure that was assigned the highest BME weight was Decoy 2, with a weight of 0.87 (in the blue box). We then examined the 3D structure of this RNA. Although not annotated in the DSSR-derived secondary structure, residue G10 and A27 were very close to each other, making it possible to form a base pair (Figure 3.10). This indicates that

maybe the DSSR-derived native structure was missing this interaction, and the SS2CS predictor, along with BME reweighting, was able to recover this interaction between G10 and A27.

3.3.2.7 Representative example 5: the human telomerase RNA

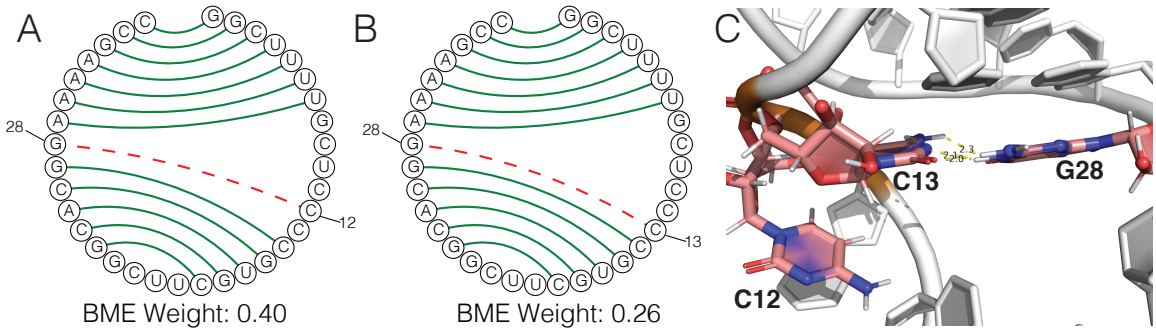

Figure 3.11: Decoys with highest and second highest BME weight of 2L3E. (A) the decoy with the highest BME weight (0.40); (B) the decoy with the second highest weight (0.26); (C) the detailed 3D structure at residues C12, C13, and G28.

Similarly, for the core domain of human telomerase RNA (PDBID: 2L3E), BME assigned the highest weight to a decoy structure (Figure 3.11; weight = 0.40), instead of the DSSR-derived native structure (weight = 0.13). Below, we show the structures of the highest (Figure 3.11A) and second highest (Figure 3.11B, weight = 0.26) decoy, each with only one extraneous base pair compared to the native structure. We then examined the 3D structures at residues C12, C13, and G28, and we discovered that there was noncanonical interaction between residues C13 and G28 (Figure 3.11C). This is probably why the decoy shown in Figure 3.11B was assigned a higher BME weight than the DSSR structure. However, there was no base pairing interaction between residues C12 and G28, and it is unclear why the decoy in Figure 3.11A was assigned the highest weight.

3.3.2.8 Discussion

Chemical shifts contain structural information which may be used to model the secondary or tertiary structures of RNAs. There are two ways to incorporate chemical shifts into RNA structure prediction. The first method uses chemical shifts information directly in the modeling process, for example, in the CS-Fold framework we developed to predict RNA secondary structures, the chemical shifts derived base pairing status predictions were used as restraints when folding secondary structures. Chemical shifts have also been incorporated in Rosetta modeling and were validated to improve tertiary structure prediction accuracy.[28] However, incorporating restraints during simulation will increase computational cost. Thus, in this chapter, we explored the structure resolving power of chemical shifts through probabilistic modeling of RNA secondary structures.

In this chapter, we developed a chemical shifts predictor, SS2CS, which takes secondary structure as input and outputs the predicted chemical shifts for nonexnageable proton and carbon nuclei. Testing result shows that SS2CS was able to achieve similar prediction accuracy (MAE was 0.84 ppm for carbon and 0.11 ppm for proton nuclei) compared to predictors which take tertiary structure as input.[24] With the SS2CS derived predictions, we were able to use BME to identify the native structure or a near-native structure from a low energy structural ensemble. BME was able to identify the DSSR-derived native structures from a set of *MC-Fold* (or *AllSub*) generated decoys for 7 out of the 16 testing RNAs. For some of the cases for which BME did not identify the DSSR-derived native structure, we discovered base pairing interactions between residues which were recovered by BME selected structure, but not by the DSSR-derived native structure. These interactions may be noncanonical interactions that were not included in DSSR annotation.

Table 3.7: Chemical shift error analysis for 1HWQ

Structure	TPR	PPV	MAE	RMSE	τ	BME weight
Decoy 1	1.00	0.64	0.69	1.26	0.88	0.02
Decoy 2	1.00	0.69	0.67	1.22	0.88	0.08
Decoy 3	1.00	0.69	0.71	1.30	0.88	0.01
Decoy 4	1.00	0.69	0.73	1.32	0.88	0.06
Decoy 5	0.89	0.62	0.71	1.27	0.88	0.08
Decoy 6	1.00	0.75	0.69	1.26	0.89	0.06
Decoy 7	1.00	0.75	0.76	1.37	0.88	0.02
Decoy 8	1.00	0.69	0.73	1.32	0.89	0.05
Decoy 9	1.00	0.75	0.64	1.19	0.89	0.15
Decoy 10	1.00	0.69	0.73	1.32	0.88	0.02
DSSR Structure	1.00	1.00	0.73	1.38	0.90	0.45

In general, the structure selected using MAE or RMSE was consistent with the structure selected using BME weight. However, we noticed that there were cases (1HWQ, 2NBY, 2NC0, and 5KH8) where BME identified the native structure, but MAE or RMSE failed to identify the native structure. For example, for the structural ensemble of the VS ribozyme substrate stem-loop RNA (PDBID: 1HWQ), Decoy 9 exhibited the lowest MAE and RMSE between experimental and predicted chemical shifts. However, BME was able to recover the DSSR-derived native structure and assigned the highest weight to it. On the other hand, the variation across different decoys when using Kendall τ coefficient was very small; thus it is impossible to use it to select the native structure. These results indicate that BME is more powerful in identifying the native-like structure compared to MAE, RMSE, or τ.

3.4 References

(1) Bonomi, M.; Heller, G. T.; Camilloni, C.; Vendruscolo, M. *Current opinion in structural biology* **2017**, *42*, 106–116.

(2) Baldwin, A. J.; Kay, L. E. *Nature chemical biology* **2009**, *5*, 808.

(3) Zhao, B.; Guffy, S. L.; Williams, B.; Zhang, Q. *Nat. Chem. Biol.* **2017**, *13*, 968.

(4) Reuter, J. S.; Mathews, D. H. *BMC bioinformatics* **2010**, *11*, 129.

(5) Das, R.; Karanicolas, J.; Baker, D. *Nat. Methods* **2010**, *7*, 291.

(6) Deb, I.; Frank, A. T. *J. Chem. Theory Comput.* **2019**, *15*, 5817–5828.

(7) Choy, W.-Y.; Forman-Kay, J. D. *Journal of molecular biology* **2001**, *308*, 1011–1032.

(8) Bottaro, S.; Bussi, G.; Kennedy, S. D.; Turner, D. H.; Lindorff-Larsen, K. *Science advances* **2018**, *4*, eaar8521.

(9) Cesari, A.; Gil-Ley, A.; Bussi, G. *Journal of chemical theory and computation* **2016**, *12*, 6192–6200.

(10) Bottaro, S.; Bengtsen, T.; Lindorff-Larsen, K. *BioRxiv* **2018**, 457952.

(11) Hummer, G.; Köfinger, J. *The Journal of chemical physics* **2015**, *143*, 12B634_1.

(12) Pitera, J. W.; Chodera, J. D. *Journal of chemical theory and computation* **2012**, *8*, 3445–3451.

(13) Kofinger, J.; Stelzl, L. S.; Reuter, K.; Allande, C.; Reichel, K.; Hummer, G. *Journal of chemical theory and computation* **2019**, *15*, 3390–3401.

(14) Zuker, M.; Mathews, D. H.; Turner, D. H. In *RNA biochemistry and biotechnology*; Springer: 1999, pp 11–43.

(15) Bloomfield, V.; Crothers, D. M., *Nucleic acids: structures, properties and functions*; 574.192 B52, 2000.

(16) Mathews, D. H.; Turner, D. H. *Current opinion in structural biology* **2006**, *16*, 270–278.

(17) Turner, D. H.; Mathews, D. H. *Nucleic acids research* **2010**, *38*, D280–D282.

(18) Xia, T.; SantaLucia Jr, J.; Burkard, M. E.; Kierzek, R.; Schroeder, S. J.; Jiao, X.; Cox, C.; Turner, D. H. *Biochemistry* **1998**, *37*, 14719–14735.

(19) Low, J. T.; Weeks, K. M. *Methods* **2010**, *52*, 150–158.

(20) Hajdin, C. E.; Bellaousov, S.; Huggins, W.; Leonard, C. W.; Mathews, D. H.; Weeks, K. M. *Proceedings of the National Academy of Sciences* **2013**, *110*, 5498–5503.

(21) Wuchty, S.; Fontana, W.; Hofacker, I. L.; Schuster, P. *Biopolymers: Original Research on Biomolecules* **1999**, *49*, 145–165.

(22) Duan, S.; Mathews, D. H.; Turner, D. H. *Biochemistry* **2006**, *45*, 9819–9832.

(23) Parisien, M.; Major, F. *Nature* **2008**, *452*, 51–55.

(24) Frank, A. T.; Law, S. M.; Brooks III, C. L. *J. Phys. Chem. B* **2014**, *118*, 12168–12175.

(25) Lu, X.-J.; Bussemaker, H. J.; Olson, W. K. *Nucleic Acids Res.* **2015**, *43*, e142–e142.

(26) Aeschbacher, T.; Schubert, M.; Allain, F. H.-T. *J. Biomol. NMR* **2012**, *52*, 179–190.

(27) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. *Journal of machine learning research* **2011**, *12*, 2825–2830.

(28) Sripakdeevong, P.; Cevec, M.; Chang, A. T.; Erat, M. C.; Ziegeler, M.; Zhao, Q.; Fox, G. E.; Gao, X.; Kennedy, S. D.; Kierzek, R., et al. *Nature methods* **2014**, *11*, 413.

CHAPTER IV

Chemical Shift-Based Annotation of RNA Structure

4.1 Introduction

Extracting and understanding structural properties is a crucial step in RNA functional studies. In this thesis, we have focused on combining experimental data, especially NMR chemical shifts, with computational tools to improve structure prediction of RNAs.

NMR provides a set of observables from which an atomic model of the RNA and descriptions of the RNA dynamics can be reconstructed. In `Chapter II` and `III`, the importance of NMR chemical shifts on RNA structure prediction was highlighted from two perspectives: (1) base pairing status information can be extracted from chemical shifts via machine learning models and used as folding restraints to guide secondary structure prediction; (2) chemical shifts have structural resolving power that could be used to identify native-like structure from a structure ensemble. The results confirm that chemical shifts are indeed structural "fingerprints" and are extremely sensitive to RNA base pairs and tertiary structures. However, full extraction of structural properties that are contained in chemical shifts is yet to be done.

Here we explored *structural annotation* using only *assigned* chemical shifts. Pre-

vious methods in structural annotation usually require 3D coordinates or secondary structures as input. For example, MC-Annotate and 3DNA[1,2] can perform annotation using tertiary structures to extract a set of structural properties, including base pairing interactions, sugar puckering modes, stacking interactions, and other conformational parameters. These parameters could help understand and reconstruct conformational transitions. Other programs, such as bpRNA and BPViewer,[3,4] focus more on secondary structure properties including, classification of canonical and noncanonical base pairs as well as analysis of secondary structure motifs.

The programs and tools mentioned above require an accurate structure model as input so that the structural parameters can be derived from the coordinates. However, tools are also needed when no such structure model is available. The annotation, or the prediction, of structural properties, could help guide or improve secondary and tertiary structure modeling. For example, in `Chapter II`, we highlighted the improvement in secondary structure folding when base pairing status predictions were incorporated.

Thus, in this chapter, we developed a fast and straightforward approach that uses only assigned chemical shifts as input and outputs the structural properties through a series of classifications. The structural properties to be annotated include solvent accessible surface area (SASA), *syn* and *anti* conformation, base pairing status, stacking interaction, and sugar puckering mode. We converted the annotation of structural properties into a set of classification problems that can be solved with machine learning techniques. For example, our approach can predict whether a residue/nucleotide adopts *syn* or *anti* conformation using the chemical shifts associated with that residue (and the neighboring residues).

Different from `Chapter III`, where independent chemical shifts prediction models were trained for different nucleus types, here we utilized the idea of *multi-task learning* in which the predictions of all properties are generated from a single machine learning

model. In this chapter, I will introduce multi-task learning with a progressive neural network to perform structure annotation based on chemical shifts. The model's performance was compared with independent multilayer perceptron (MLP) models and chained MLP models through cross validation. MLP is a class of "vanilla" neural network models and consists of an input layer, hidden layer(s), and output layer. The details of the neural network were discussed in `Introduction`.

4.2 Methods

4.2.1 Data sets

For 108 RNAs, atomic structures and NMR chemical shifts were retrieved from the Protein Data Bank (PDB: `http://www.pdb.org/`) and the Biological Magnetic Resonance Data Bank (BMRB: `http://www.bmrb.wisc.edu/`) respectively. Four RNAs were used as testing set whose PDBIDs are: 2JTP, 2LU0, 5KH8, and 2N1Q. The first three RNAs were used as representative examples in `Chapter II` to demonstrate the accuracy of CS2BPS and CS-Fold. The last RNA, 2N1Q, is the longest RNA in our dataset with 155 nts.

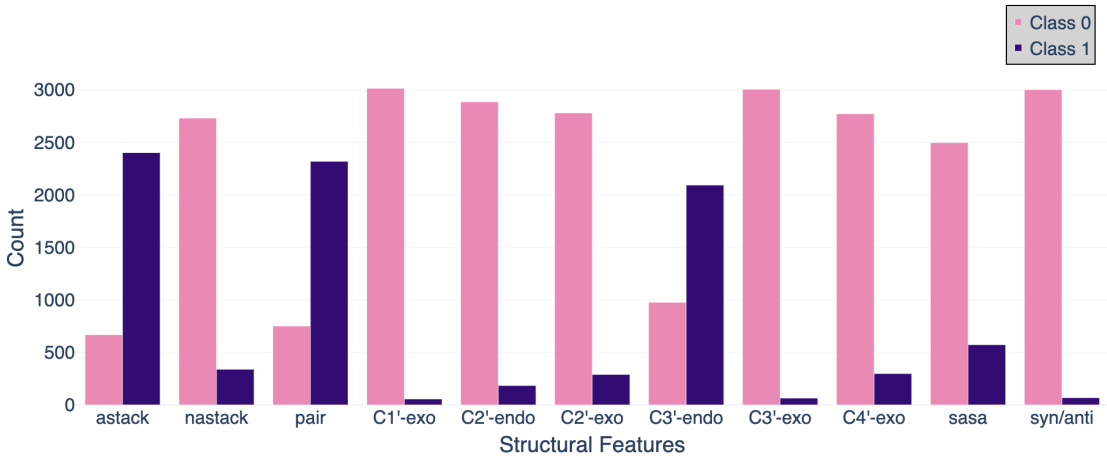

Figure 4.1: Distribution of annotation targets (structural properties) in the training set. Class 1 is the positive class and Class 0 is the negative class. "astack" represents stacking interaction between adjacent bases while "nastack" represents non-adjacent stacking interaction.

Shown in Figure 4.1 is the distribution of 11 annotation targets (structural properties) in our training set. As mentioned in `Introduction`, I have converted the task of structural annotation into a series of machine learning classification problems. As shown in Figure 4.1, Class 1 is the "positive" class (meaning the residue is involved in such interaction), and Class 0 is the "negative" class (meaning there is no such interaction). The label of each property indicates the interaction that this nucleotide is involved in:

- stacking interaction: the first two properties are stacking interactions between adjacent bases and non-adjacent bases; more nucleotides were involved in adjacent stacking.

- base pairing interaction: more nucleotides are paired than unpaired in the training set; here, we used MC-Annotate derived base pairs instead of DSSR (in `Chapter II`) derived base pairs, so noncanonical base pairs are included.

- sugar puckering mode: instead of predicting directly which sugar puckering mode a residue adopts, we predicted the probability of adopting each sugar puckering mode, as shown in the next six properties; most RNA nucleotides adopt C3'-endo puckering mode.

- solvent accessible surface area (SASA): a threshold was calculated using the mean and standard deviation (SD) of SASA values across the training set. Nucleotide with SASA value above the threshold was defined as Class 1, and below the threshold was defined as Class 0.

- *syn* or *anti* conformation: different from previous properties, there is no positive or negative class for *syn* or *anti* conformations; we simply defined *syn* as Class 0 and *anti* as Class 1.

The structural annotation properties were retrieved from 3D coordinates of RNAs

using MC-Annotate.[1] MC-Annotate takes the PDB files of DNA or RNA and quickly extracts the structural information to simplify further analysis. The solvent accessibility data (SASA) was also calculated using MC-Annotate.

We combined these annotation properties with chemical shift data associated with individual residues and of neighboring residues to form a CS-structural properties dataset. The training set had 3068 samples, each corresponding to a residue, and the testing set had 284 samples. We included chemical shifts for neighboring residues because we believe the local physio-chemical environment will have an impact on the chemical shifts of these neighboring residues. The detailed comparison of how many neighbors should be included will be discussed in `Results`.

4.2.2 Multi-task classifiers

We have the dataset with both chemical shift data and the corresponding annotation properties for each residue, and we have converted the problem of structural annotation to a set of classification problems. The next question is model selection. Similar to what has been done in `Chapter III`, where we developed independent chemical shifts predictors for individual nucleus types, the most naive approach would be to develop separate classifiers for different properties/tasks. However, this approach may be inefficient and may not fully take advantage of the correlation between different annotation properties. I calculated the Pearson correlation coefficients between pairs of annotation properties (Figure 4.2) and found that there were correlations between some of the properties. For example, there was a negative correlation among different sugar puckering modes, since one nucleotide cannot adopt multiple modes. There was also a negative correlation between SASA and adjacent stacking; the more stacked a base may be, the smaller surface area it has that is accessible to solvent.

In addition to the naive approach, we also tried a chained model where the pre-

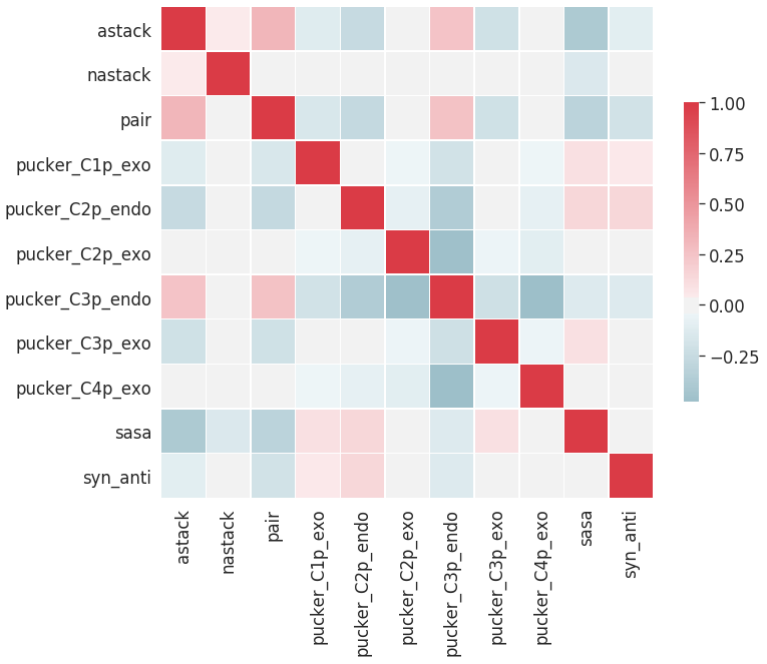

Figure 4.2: Correlation between structural properties.

dictions of the previous properties were used as input for the following predictions. Combining results from chains with a different order of properties, we developed an ensemble model based on individual chained models and tested it against the naive classifiers.

The final model I explored is a *progressive neural network* model.[5] Figure 4.3 is a simple description of a progressive neural network model trained on two tasks (output_1 and output_2), each with two hidden layers (h_1^j and h_2^j, where j is the jth task). When training the first task, the model is a regular neural network model with two hidden layers $h_1^{(1)}$ and $h_2^{(1)}$. When training the second task, the parameters learned for the first task are fixed. The hidden layer for the second task $h_2^{(2)}$ takes input from the its previous layer $h_1^{(2)}$ as well as from $h_1^{(1)}$ via a lateral adapter layer a (the blue box in Figure 4.3). Similarly, the output of the second task takes input from $h_2^{(2)}$ and $h_2^{(1)}$ (via lateral connection). The lateral connections between different neural networks can transfer knowledge between tasks and improve convergence speed.[5] Moreover, using a new neural network for a new task could avoid catastrophic inference,[6] which is the

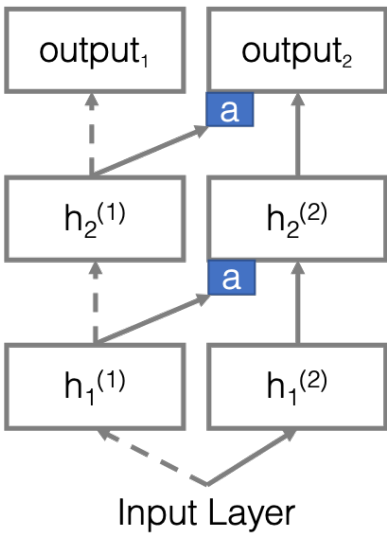

Figure 4.3: Progressive neural network model adapted from reference.[5] $h_i^{(j)}$ represents the ith hidden layer for the jth task/property. When training the second task (the second column), the parameters associated with the first task are "frozen" and served as input via a lateral connection (arrow pointing to the blue box).

tendency of neural networks to forget previously learned knowledge. The progressive neural network model in this chapter was implemented through *DeepChem* library in Python.[7]

4.3 Results

4.3.1 Model selection

We explored three different methods to perform the multi-task classification using the training set. The training set combined residues from 104 RNAs, including the non-exchangeable chemical shifts associated with each residue and the neighboring residues as well as the 11 annotation properties that were retrieved from the 3D structure using MC-Annotate. Here, the definition of "neighboring residues" refers to the *three* residues before and after the current residue. We started by including three neighbors, but the impact of the number of neighbors was also explored at the end of this section. The training set had 3068 samples (or residues) and 167 features

including chemical shifts and nucleotide types.

The first method was to develop individual multilayer perceptron (MLP) models (or vanilla neural network models) for different structural annotation tasks. The base model used was the *MLPClassifier* from scikit-learn[8] with one hidden layer and 100 hidden neurons. The activation function was ReLU. The model was optimized using Adam optimizer based on the log-loss function. This approach treated each annotation property as an independent task and did not use predictions or information from other tasks.

To utilize the correlation between different properties, I then explored a chained classifier. In the chained classifier, I still employed the same base learner which was the MLP classifier with the same architecture as in the first method. However, the input features for training the model were not just the chemical shifts associated with each residue and the neighboring residues. Instead, output predictions from previous tasks, such as the predictions of whether a residue has adjacent stacking, were used as input features in the training of the subsequent tasks. That is, the last task that was trained took as input all the predictions from the previous 10 annotation properties. Since the order of annotation properties did not have any chemical or structural relevance, and it was impossible for us to know the optimal order in advance, I then ran 10 random orders and calculated the average predictions based on the 10 random chains. The performance of the independent MLP classifier and the chained classifier was compared via a 5-fold cross validation by calculating the balanced accuracy of the classification for all annotation properties. The balanced accuracy was defined as the average of "sensitivity" and "specificity". I used the balanced accuracy instead of the overall accuracy to better assess the imbalanced dataset.

In Figure 4.4, I show the cross validation results of the first three annotation properties (adjacent stacking, non-adjacent stacking, and base pairing) when using independent and chained classifiers. Since it was impossible to know the optimal order

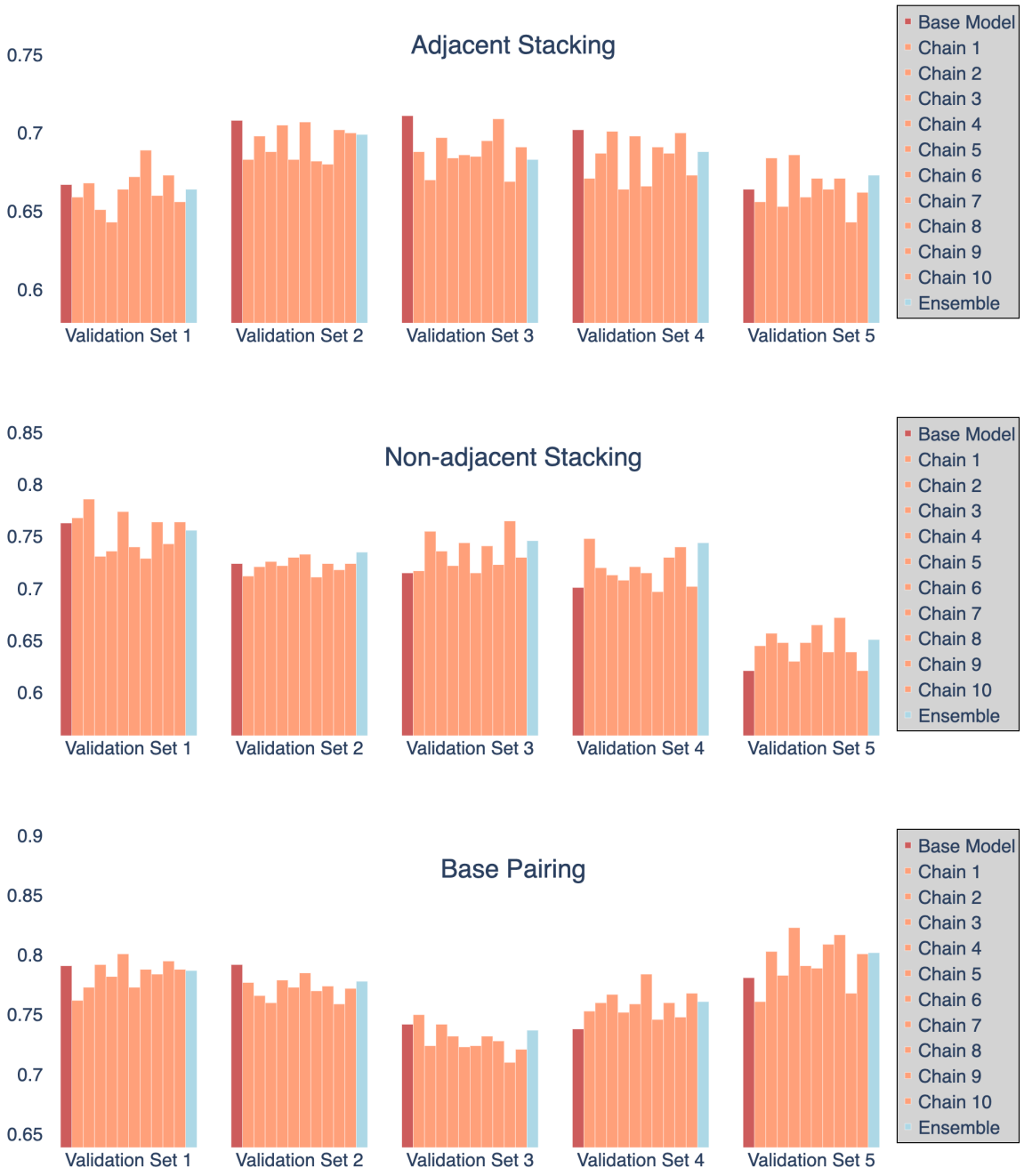

Figure 4.4: Performance comparison between independent MLP model and chained model. Shown in the figures are the balanced accuracies of three annotation tasks (adjacent stacking, non-adjacent stacking, and base pairing interaction) via the 5-fold cross validation. The first bar in each group (*dark red*) represents the independent MLP model; the next 10 bars (*salmon*) represent chained models with different orders of annotation properties; the last bar (*light blue*) represents the ensemble model that is calculated by averaging the predictions of 10 chained models.

of properties in advance, I tested 10 random orders (*salmon* bar). For these three tasks, some of the chained models performed better than the independent classifiers (*dark red* bar). And when using the ensemble classifier (*light blue* bar), there was improvement in balanced accuracy for some but not all of the structural properties (which was also shown in Table 4.1). The results for the other 8 annotation properties are shown in Supporting Information (Figure B.8, B.9, and B.10).

Table 4.1: Balanced accuracy of validation results using three different models

Property	Base model[1]	Ensemble classifier[2]	Progressive neural network
Adjacent Stacking	0.690	0.681	0.699
Non-adjacent Stacking	0.705	0.727	0.808
Base Pairing	0.769	0.773	0.812
C1'-exo	0.521	0.541	0.642
C2'-endo	0.634	0.627	0.787
C2'-exo	0.538	0.532	0.568
C3'-endo	0.623	0.624	0.661
C3'-exo	0.544	0.564	0.780
C4'-exo	0.536	0.542	0.555
SASA	0.695	0.701	0.719
syn/*anti*	0.684	0.678	0.762

[1] Base model refers to the independent MLP classifiers;
[2] Ensemble classifier uses the averaged predictions from 10 chained classifiers.

Shown in Table 4.1 are the averaged balanced accuracy scores of the 5 validation sets for each annotation property. The performance of the ensemble model and the base model (independent MLP classifier) was very similar to each other, with slight improvement when using the ensemble classifier.

I then explored the progressive neural network classifier using the same model architecture as the MLP classifier with the same number of hidden layers (1), number of hidden neurons (100), learning rate (0.001), optimizer (Adam), and activation function (ReLU). I compared and showed the balanced accuracy in Figure 4.5. The balanced accuracy for each method was calculated from averaging 5 validation sets. As

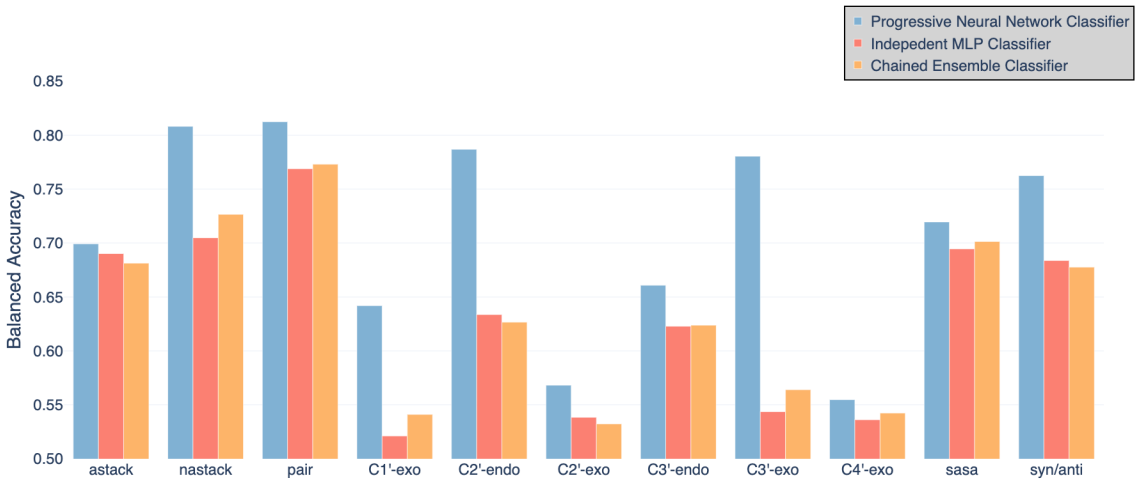

Figure 4.5: Balanced accuracy of the base model (independent MLP classifier; *salmon*), chained ensemble classifier (*orange*) and progressive neural network classifier (*blue*) averaged from 5 validation sets.

mentioned above, the ensemble classifier did not show significant improvement when compared with the independent classifier. However, the improvement of balanced accuracy when using the progressive neural network model is much more significant compared with the ensemble classifier, as can be seen from Figure 4.5. The model was worse in predicting sugar puckering modes than in predicting other structural properties. The impact of the change of sugar puckering on the physio-chemical environment surrounding an atom may be very subtle and cannot be reflected in chemical shifts. Based on the cross validation results, the progressive neural network model was used for the following analysis.

4.3.2 Error analysis

I then trained the progressive neural network model on the entire training set and tested on the testing set. The testing set consists of residues from 4 testing RNAs whose PDBIDs are 2JTP, 2LU0, 5KH8, and 2N1Q. The testing set had 284 residues and 167 features.

In Table 4.2, I reported the balanced accuracy, sensitivity, and specificity of the

testing set predictions for each property. The balanced accuracy of cross validation was also reported (in "()") to for comparison. In general, the performance of the testing set was consistent with the cross validation results: sugar puckering modes were predicted with relatively low accuracy, except for C2'-endo. This might be due to the limited sample size of the training data; it is possible that for these difficult-to-predict structural properties, a larger training size is needed for the generalization of the model. Also, as mentioned before, the consistent low prediction accuracy may be attributed to the weak correlation between chemical shifts and sugar puckering mode. The change of sugar puckering mode may cause very subtle changes in the surrounding chemical environment of an atom, thus an insignificant change of chemical shift data.

Table 4.2: The balanced accuracy, sensitivity, and specificity of the testing set predictions

Property	Balanced accuracy[1]	Sensitivity	Specificity
Adjacent Stacking	0.721 (0.699)	0.626	0.816
Non-adjacent Stacking	0.845 (0.808)	0.830	0.861
Base Pairing	0.735 (0.812)	0.681	0.788
C1'-exo	0.668 (0.642)	0.500	0.837
C2'-endo	0.852 (0.787)	0.909	0.795
C2'-exo	0.651 (0.568)	0.615	0.686
C3'-endo	0.597 (0.661)	0.643	0.551
C3'-exo	0.580 (0.780)	0.250	0.911
C4'-exo	0.635 (0.555)	0.667	0.604
SASA	0.735 (0.719)	0.708	0.762
syn/anti	0.662 (0.762)	0.400	0.923

[1] : the numbers in the "()" are the mean balanced accuracy values of the cross validation sets.

Shown in Figure 4.6 are the "maps" between the predicted class and the actual class for four testing RNAs. In the figure, *black* rectangles indicate that the corresponding structural property was correctly classified; *biege* rectangles represent false positive predictions, meaning the predicted structural property did not exist; *teal* rectangles represent false negative predictions, meaning that there was such structural property, but the model did not predict it. The values labeled at the top of

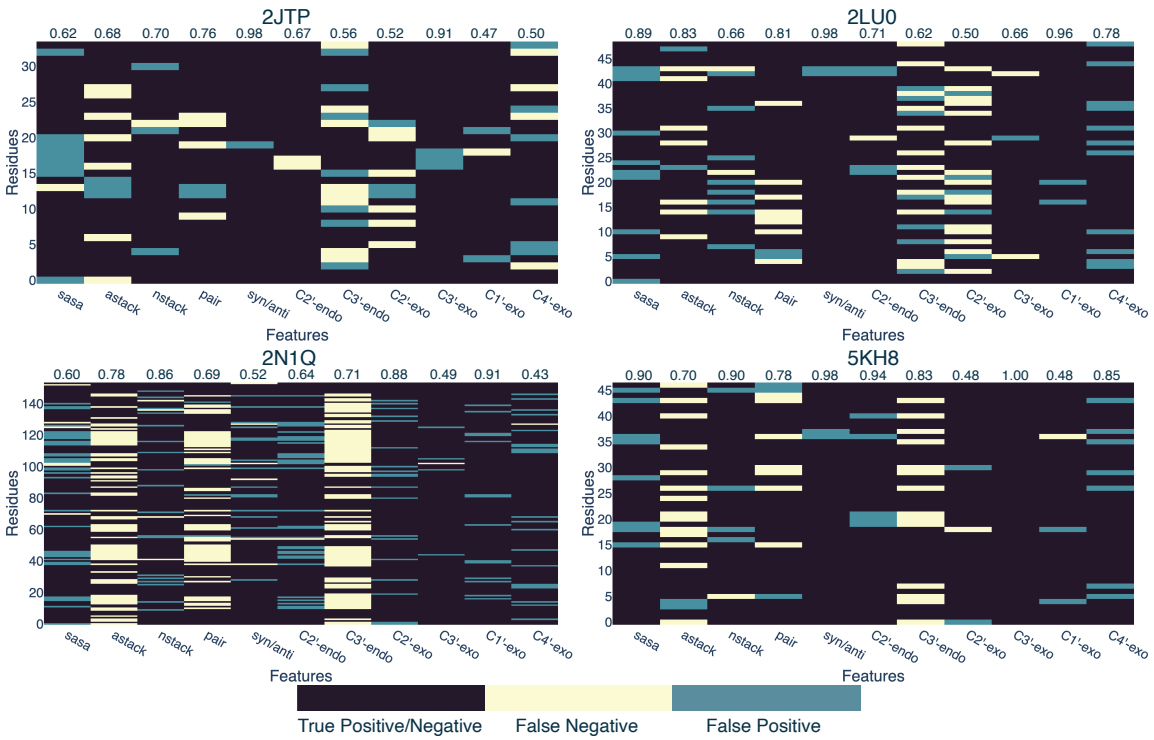

Figure 4.6: Prediction maps of structural properties for testing RNAs. *Black* rectangles indicate that for the current residue and structural property, the prediction was correct; *biege* rectangles indicate that the predicted structural property did not exist for the current residue; *teal* rectangles indicate that the residue had such property, but the model failed to predict it. The values at the top of each map are the balanced accuracy values.

each map are the balanced accuracy values. When predicting base pairing status, all four testing RNAs had more false negatives than false positives. This is probably because MC-Annotate included noncanonical base pairs, which could be more difficult to predict using chemical shifts than canonical base pairs. When predicting *syn*/*anti* conformation, only 2N1Q had false negatives, meaning the model misclassified some residues with *syn* conformation as *anti* conformation. 2N1Q also exhibited the lowest prediction accuracy among the four testing RNAs when predicting other structural properties. One possible source of errors was the imputation of missing chemical shifts, as for 2N1Q, we only had proton chemical shifts and had to impute all carbon chemical shifts. Moreover, in terms of predicting stacking interactions, all four testing RNAs had more false negatives for adjacent stacking and more false positives for

non-adjacent stacking. This is probably due to the imbalanced dataset (Figure 4.1). Residues that exhibited different prediction patterns, that are, residues for which adjacent stacking predictions were false positives and for which non-adjacent stacking predictions were false negatives, were usually found in regions with complex structures. For example, internal loop (for 2JTP), three way junction (for 2LU0), and long range base pairs (for 5KH8).

4.3.3 Impact of neighboring residues

I started by including chemical shifts and nucleotide types associated with 3 neighboring residues as features: for residue i, the features are the chemical shifts and nucleotide types for residues $i-3, i-2, i-1, i, i+1, i+2$, and $i+3$. Chemical shifts that are not available were imputed using MICE as in `Chapter II`. Chemical shifts of neighboring residues that do not exist were encoded as 0. We included neighboring residues because we believe that the residues in close distance will affect the physiochemical environment and this local structural information may be contained in the chemical shifts of the neighboring residues.

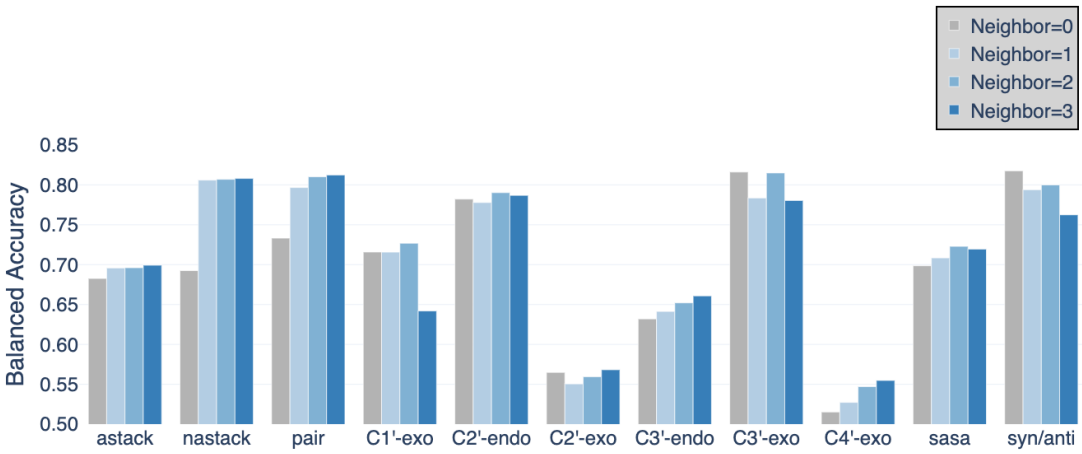

Figure 4.7: Balanced accuracy of validation sets when including different number of neighboring residues.

I then explored how many neighbors should be included to achieve the best accu-

racy. Figure 4.7 confirms that our hypothesis was correct, that including neighboring residues did improve prediction accuracy for most of the structural annotation tasks, except for C3'-exo sugar puckering and *syn*/*anti* conformation. Since *syn*/*anti* conformation is a global property, it is reasonable that it was not affected by including neighboring chemical shifts. On the other hand, in terms of stacking interaction and base pairing status, including features from more neighboring residues improved the prediction accuracy. This is because these were interactions between different residues that are spatially close and chemical shifts from neighboring residues may contain structural information about these interactions. It is also interesting to see that the number of neighbors included does not affect the accuracy as we hypothesized, that is, including more neighboring residues will increase the prediction accuracy. In fact, for some properties, including 2 neighboring residues $(i-2, i-1, i, i+1, i+2)$ exhibited better balanced accuracy score compared with using 3 neighboring residues.

4.4 Discussion

In this chapter, I developed a method for annotating structural properties, including solvent accessibility, base pairing interaction, stacking interaction, conformation, and sugar puckering mode using only non-exchangeable chemical shifts. The structural annotation tasks were converted to a set of classification problems, allowing us to apply machine learning methods. With a careful model selection, we found that the progressive neural network model was able to utilize the inherent correlations between different annotation properties and outperformed independent MLP classifier and chained MLP classifier. The progressive neural network model applied in this project was built upon *DeepChem* library in Python. We also explored whether including chemical shifts from neighboring residues could enhance model performance. Although including more features improved prediction accuracy for most of the properties (Figure 4.7), it is not necessary to include 3 neighboring residues.

We then applied the trained model to the testing set, which contained residues from 4 RNAs (2JTP, 2LU0, 5KH8, and 2N1Q). The model was able to predict some structural properties with reasonable accuracy, exhibiting similar low accuracy when predicting sugar puckering mode as in validation sets. We hypothesize that sugar puckering change may be too subtle to be reflected in chemical shifts. The quality of structural annotation also depends on the proportion of missing chemical shifts data and the method we used to impute missing data. When there are too many chemical shifts missing, the annotation accuracy for that RNA may be affected. We currently used the same technique, MICE, to impute missing chemical shifts, as in `Chapter II`, but more tests should be done to improve the accuracy of chemical shifts imputation.

Many questions remain to be answered about the relationships between chemical shifts and structural properties. For example, by studying the correlation between missing data and annotation accuracy of each structural property, we could understand the impact of chemical shifts of different nucleus types on each structural property. It is also interesting to explore the importance of chemical shifts of spatially adjacent residues, instead of sequentially adjacent neighbors. Moreover, CS-Annotate could also be used for evaluating the quality of a structural model by comparing the structural properties annotated from chemical shifts and properties derived from a structure.

4.5 References

(1) Gendron, P.; Lemieux, S.; Major, F. *Journal of molecular biology* **2001**, *308*, 919–936.

(2) Lu, X.-J.; Olson, W. K. *Nucleic acids research* **2003**, *31*, 5108–5121.

(3) Danaee, P.; Rouches, M.; Wiley, M.; Deng, D.; Huang, L.; Hendrix, D. *Nucleic acids research* **2018**, *46*, 5381–5394.

(4) Yang, H.; Jossinet, F.; Leontis, N.; Chen, L.; Westbrook, J.; Berman, H.; Westhof, E. *Nucleic acids research* **2003**, *31*, 3450–3460.

(5) Rusu, A. A.; Rabinowitz, N. C.; Desjardins, G.; Soyer, H.; Kirkpatrick, J.; Kavukcuoglu, K.; Pascanu, R.; Hadsell, R. *arXiv preprint arXiv:1606.04671* **2016**.

(6) McCloskey, M.; Cohen, N. J. In *Psychology of learning and motivation*; Elsevier: 1989; Vol. 24, pp 109–165.

(7) Ramsundar, B.; Leswing, K. In *ABSTRACTS OF PAPERS OF THE AMERICAN CHEMICAL SOCIETY*, 2019; Vol. 257.

(8) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. *Journal of machine learning research* **2011**, *12*, 2825–2830.

CHAPTER V

Conclusions and Perspectives

Recent studies have revealed the existence of thousands of functional non-coding RNAs (ncRNAs) with complex cellular functions. To better understand the functional mechanism of these RNAs, it is important to determine their structures first. However, characterizing the structure of an RNA can be challenging both experimentally and computationally. RNA molecules can be very dynamic and thus hard-to-crystallize, making it challenging to use X-ray crystallography to solve their structures. To study these hard-to-crystallize RNAs, NMR spectroscopy is widely used. NMR provides experimental observables, such as chemical shifts and NOE distances, that can be used to derive the structure of an RNA. In particular, chemical shifts are considered as structural "fingerprints" and contain information about the surrounding physio-chemical environment. In my thesis, I have developed different computational methods that were based on machine learning and probabilistic modeling to extract information from chemical shifts and use that information to improve the structure modeling of RNAs.

Predicting the secondary structure of RNAs is typically the first step in exploring relationships between their sequence, structure, and function. Given the sequence of an RNA, most algorithms attempt to identify a single structure that is compatible with that sequence. However, to achieve their function, RNAs typically transition

between distinct conformational states. As such, to rationalize relationships between the sequence, the structure(s), and the function of RNA, methods are needed to map their entire conformational landscape. In principle, if the structural "fingerprints" of individual conformational states of an RNA can be obtained experimentally, these could be used together with well-established computational algorithms to "conditionally" predict the structure of each of these states – thus mapping many distinct conformational states to a single RNA.

In `Chapter II`, I developed a chemical shift-based folding framework, referred to as the **CS-Fold** framework,[1] for "conditionally" predicting the secondary structure of RNAs using assigned NMR chemical shift data. Extensive testing of the CS-Fold framework proves that from assigned NMR chemical shifts, we could (1) accurately predict the base-pairing status of individual residues in an RNA (via the **CS2BPS** classifiers) and (2) accurately predict the secondary structure of RNAs using folding restraints derived from our classifiers.

Using experimental data or the information derived from experimental data as restraints during structure modeling can be computationally expensive. The alternative is to generate a set of low energy structure models using free energy minimization and identify the structure model that is most consistent with experimental data. Thus, in `Chapter III`, I have explored using chemical shifts as "filters" to identify the "best" structure model from a set of low energy structure models. To do this, tools are required that can back-calculate or predict NMR chemical shifts from a given structure model. I developed **SS2CS**, a chemical shift predictor that was based on random forest technique. When given a secondary structure of an RNA, SS2CS can predict the nonexchangeable chemical shifts of carbon and proton nuclei with high accuracy: the mean absolute errors (MAEs) between predicted and measured chemical shifts were 0.84 ppm for carbon nuclei and 0.11 ppm for proton nuclei.

In `Chapter III`, I have explored the probabilistic modeling of RNA secondary

structures using Bayesian/maximum entropy (BME).[2] In this application, I investigated whether the difference between measured and predicted chemical shifts could be used to identify the "best" secondary structure model. For an ensemble of low energy secondary structure models, I applied BME to reweight members in the ensemble in order to achieve the best agreement with experimental chemical shifts. The result indicates that BME weight has the resolving power to recover the native or near-native structures from a set of low energy secondary structures. BME was able to identify the DSSR-derived[3] native structure for 7 out of the 16 RNA ensembles. For the structures that were assigned the highest BME weights, 13 of them exhibited high true positive rates (TPRs) and high positive predicted values (PPVs) (both $>$ 0.80) compared to the native structure. It is also found that BME was able to recover some noncanonical interactions that were not highlighted in DSSR-derived structures as DSSR only included canonical base pairs.

These experiments confirmed that NMR chemical shifts provide valuable structural information of RNAs, which can be used to improve structure modeling or determine the quality of a structure model. In `Chapter IV`, I further explored if there were other structural information that could be extracted from chemical shifts. In other words, I investigated whether chemical shifts could be used to annotate a set of structural properties, including sugar puckering, stacking interactions, *syn*/*anti* conformation, base pairing status, and solvent accessibility. The individual structural annotation tasks were converted to a series of machine learning classification problems. For each residue, our **CS-Annotate** model can annotate the probability of each structural property for individual residues in an RNA. It is found that the CS-Annotate model was able to predict most structural properties with decent accuracies. However, the model exhibited lower prediction accuracies for sugar puckering, probably because the change of chemical shifts caused by sugar puckering was too subtle.

Overall, in my thesis, I have developed different computational tools to extract structural information from NMR chemical shifts and use the extracted information on the study of RNA structure modeling. There remain many questions that can be answered. For example, using the CS-Fold framework, we could study some functional important RNA transient states.[4] Due to their low population and short lifetime, transient states are "invisible" for conventional experimental techniques. It is also challenging to model their structures by free energy minimization since they are not the global minimum in the free energy landscape. However, for some transient states, chemical shifts are accessible via the chemical exchange saturation transfer[5] (CEST) NMR. It would be interesting to explore whether CS-Fold could be used to model the secondary structures of these transient states. On the other hand, the chemical shifts predictor we developed in `Chapter II`, SS2CS, used a simple random forest model built upon a set of secondary structure features. Although the features of adjacent residues were considered, the current model did not include the impact of spatially adjacent residues. Residues that are far away in sequence may still have interactions in 3D structure. I believe that *graph neural network*[6] *(GNN)* may be a better alternative since it can take advantage of the whole secondary structure. One major limitation of the tools developed in this thesis is they require *assigned* chemical shifts. And to assign chemical shifts, a secondary structure model is typically assumed. One feasible plan is to explore the use of *unassigned* chemical shifts in secondary structure modeling and structure annotation. Using the chemical shifts predictor, SS2CS, one could predict chemical shifts for a set of secondary structure models. And based on each secondary structure model, experimental chemical shifts can be assigned. We believe that the native-like structure model should exhibit the lowest errors between optimally assigned chemical shifts and experimental (or predicted) chemical shifts. Based on this assumption, one could optimize the assignment of experimental chemical shifts and then use the computationally assigned chemical

shifts to model RNA structures or perform structural annotations.

References

(1) Zhang, K.; Frank, A. T. *The Journal of Physical Chemistry B* **2019**, *124*, 470–478.

(2) Bottaro, S.; Bengtsen, T.; Lindorff-Larsen, K. *BioRxiv* **2018**, 457952.

(3) Lu, X.-J.; Bussemaker, H. J.; Olson, W. K. *Nucleic Acids Res.* **2015**, *43*, e142–e142.

(4) Dethoff, E. A.; Petzold, K.; Chugh, J.; Casiano-Negroni, A.; Al-Hashimi, H. M. *Nature* **2012**, *491*, 724.

(5) Zhao, B.; Zhang, Q. *Curr. Opin. Struct. Biol.* **2015**, *30*, 134–146.

(6) Scarselli, F.; Gori, M.; Tsoi, A. C.; Hagenbuchner, M.; Monfardini, G. *IEEE Transactions on Neural Networks* **2008**, *20*, 61–80.

APPENDICES

APPENDIX A

PyShifts

The contents of this chapter were published in the following reference:

Jingru Xie, Kexin Zhang, and Aaron T. Frank. "PyShifts: A PyMOL Plugin for Chemical Shift-Based Analysis of Biomolecular Ensembles." *Journal of Chemical Information and Modeling 60.3* (**2020**): 1073-1078.

Here we present PyShifts — a PyMOL plugin for chemical shift-based analysis of biomolecular ensembles. With PyShifts, users can compare and visualize differences between experimentally measured and computationally predicted chemical shifts. When analyzing multiple conformations of a biomolecule with PyShifts, users can also sort a set of conformations based on chemical shift differences and identify the conformers that exhibit the best agreement between measured and predicted chemical shifts. Though we have integrated PyShifts with the chemical shift predictors LARMOR$^{\mathrm{D}}$ and LARMOR$^{\mathrm{C}\alpha}$, PyShifts can read in chemical shifts from any source, and so, users can employ PyShifts to analyze biomolecular structures using chemical shifts computed by any chemical shift predictor. We envision, therefore, that PyShifts (`https://github.com/atfrank/PyShifts`) will find utility as a general-purpose tool

for exploring chemical shift-structure relationships in biomolecular ensembles.

Introduction

Determining the structure of biomolecules is an important step in understanding how they execute specific cellular functions. NMR spectroscopy provides a number of observables that can be used to probe both the structural and dynamical properties of biomolecules.[1,2] In particular, NMR-derived chemical shifts provide valuable information about the conformational state(s) that are accessible to a given biomolecule. Accordingly, chemical shifts are now routinely used to model the secondary[3] and tertiary structure of proteins.[4–6] Similar approaches are now being applied to ribonucleic acids (RNAs)[7–9] and small molecules.[10] A critical component of many chemical shift-based modeling frameworks is the comparison between experimentally measured chemical shifts and chemical shifts computed from 3-dimensional (3D) coordinates of biomolecules.

Here we introduce PyShifts — a PyMOL[11] research tool to visualize and analyze chemical shift differences along a 3D biomolecular ensemble. PyShifts can compute chemical shifts directly from coordinates that are loaded into PyMOL or load chemical shifts from an external file. Once the chemical shift data is loaded, PyShifts can be used to compute the difference between those chemical shifts and a reference set of chemical shifts. Implemented in PyShifts are several features that will facilitate chemical shift-based modeling of biomolecules, with specific use cases that include: assessing consistency between measured chemical shifts and a given structural model; identifying the subset of structures in a larger ensemble that exhibit the best agreement between measured and computed chemical shifts; assigning conformational weights to individual conformers; and finally, clustering structures in an ensemble based on the (dis)similarity of their computed chemical shifts.

Related Software

Some of the features in PyShifts resemble those implemented in the PyMOL plugin, Cheshift.[12] Like Cheshift, PyShifts can be used to compute chemical shifts for $^{13}C\alpha$ and $^{13}C\beta$ in protein and facilitate comparison between the computed chemical shifts and a set of experimentally measured chemical shifts. However, PyShifts can also be used to compute chemical shifts for ^{1}H, ^{15}N, and ^{13}C backbone nuclei from 3D coordinates of proteins, as well as for ^{1}H, ^{15}N and ^{13}C from 3D coordinates of RNA. Though PyShifts does not directly compute chemical shifts, it facilitates chemical shift computation from structure by making calls to external structure-based chemical shift predictors (see below). PyShifts also contains powerful visualization capabilities that facilitate detailed comparisons between measured and computed chemical shifts. In addition, PyShifts is interfaced with tools that enable conformational weights to be optimally assigned to individual structures in a conformational ensemble and tools for clustering conformers based on their chemical shift (dis)similarity.

Methods

Basic Usage.

PyShifts takes the name of a loaded PyMOL object as input that stores the coordinates for one or multiple state(s) of a biomolecule (Figure A.1A). PyShifts then computes the chemical shifts from the coordinates of the biomolecule using either LARMOR^{D} (for RNA) or LARMOR$^{C\alpha}$ (for proteins) (Figure A.1B). For RNA, PyShifts will compute the chemical shifts for ^{1}H, ^{13}C, and ^{15}N nuclei (specifically, H1, H3, H1', H2', H3', H4', H5', H5", H2, H5, H6, H8, C1', C2', C3', C4', C5', C2, C5, C6, C8, N1, and N3 nuclei). For proteins, PyShifts will compute chemical shifts for the backbone nuclei (specifically, HA, HN, CA, CB, C, and N nuclei). In addition to computing the chemical shifts, PyShifts can also read in chemical shifts for each

state in the object from an external file (Figure A.1B and A.1C).

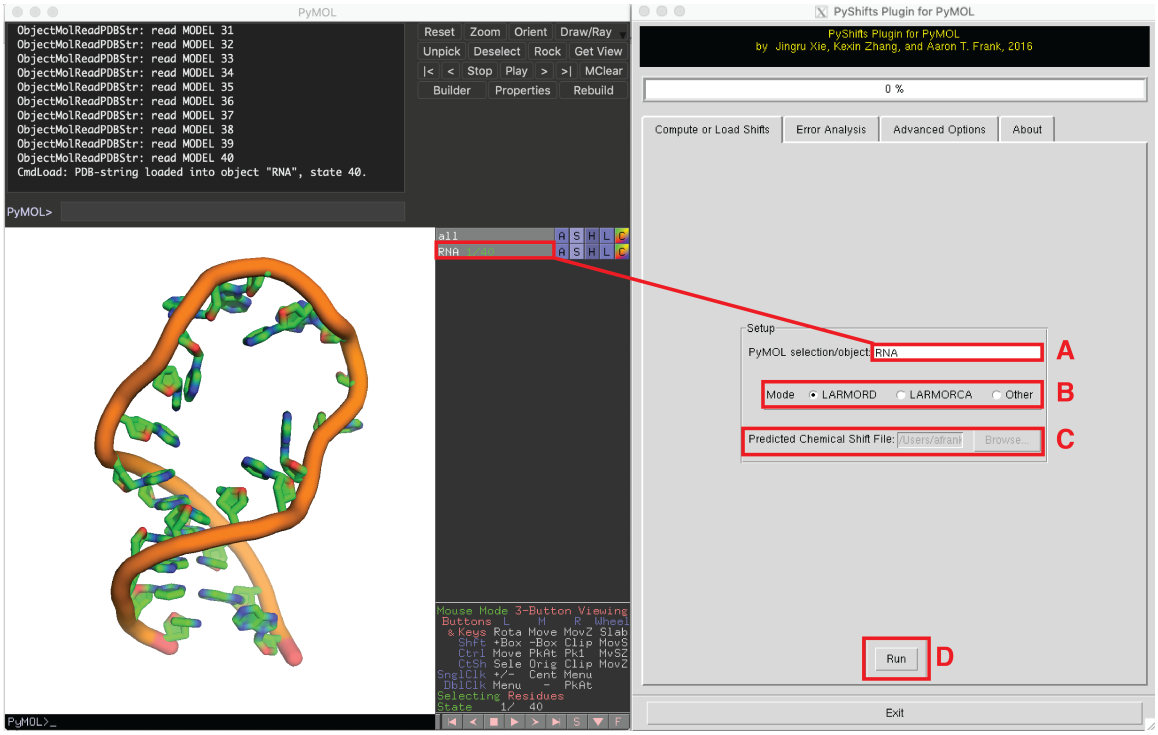

Figure A.1: To initialize PyShifts, users specify the name of loaded PyMOL object (A) from which chemical shifts will be computed using LARMOR$^{\mathrm{D}}$ or LARMOR$^{\mathrm{C}\alpha}$ by setting the mode to `LARMORD` or `LARMORCA`, respectively (B). Alternatively, by setting the mode to `Other` (B), chemical shifts for the states in the specified PyMOL object (A) can be read in from a user specified file (C). The computation or loading of chemical shifts can be initiated by clicking `Run` button (D).

Error (or Difference) Analysis.

With PyShifts, users can carry out various analyses based on computed or loaded chemical shifts. In the `Error Analysis` tab (Figure A.2), when the `Compare Shifts` button is clicked (Figure A.2A), PyShifts computes the weighted mean absolute error (MAE), the weighted root mean squared error (RMSE), and the Pearson correlation coefficient (R) between computed or loaded chemical shifts and reference chemical shifts. Typically, users will supply a measured chemical shift file that contains the chemical shifts relative to which the error analysis will be carried out (Figure A.2B). If no reference file is specified, analysis will be carried out relative to the first state.

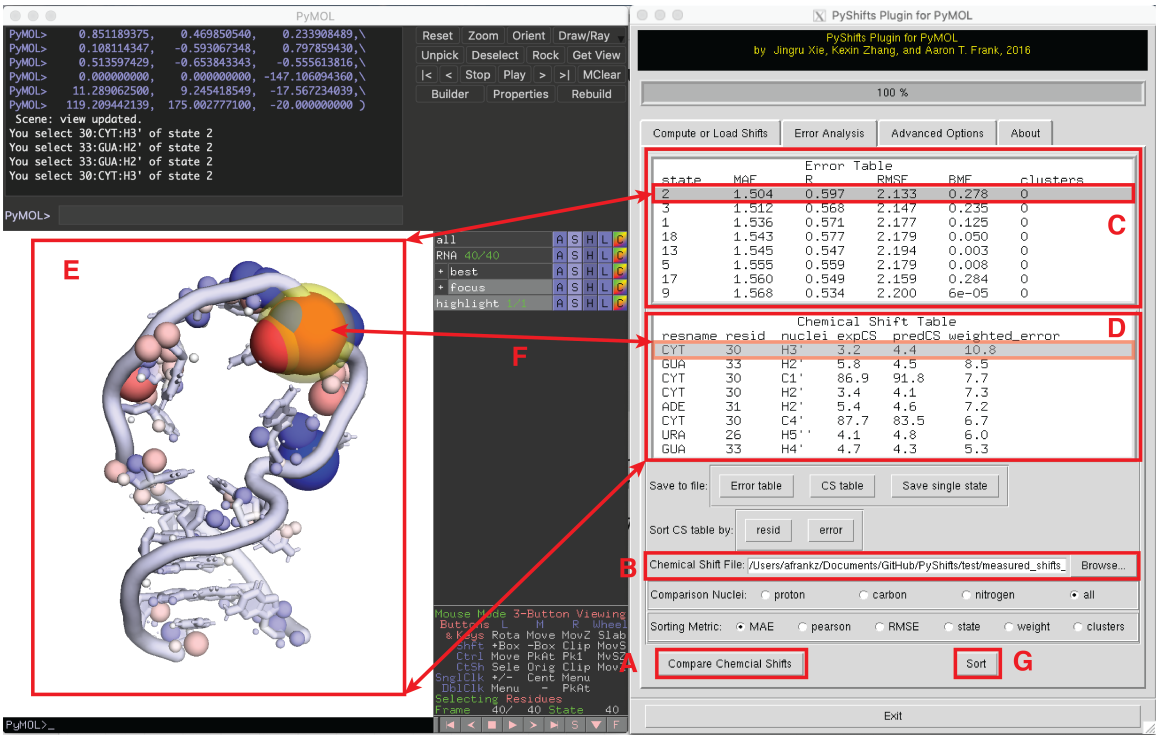

Figure A.2: PyShifts `Error Analysis` interface.

Users can specify if all chemical shifts should be used to compute these statistics, or if only chemical shifts for proton, carbon, and nitrogen nuclei should be used (Figure A.2). Error statistics (namely, MAE, RMSE, and R) for each state in the reference object are reported in the `Error Table` (Figure A.2C).

The weighted MAE and RMSE are calculated using:

$$\text{MAE} = \frac{1}{N} \sum_i \left| \frac{\delta_i^{\text{exp}} - \delta_i^{\text{pred}}}{\sigma_i} \right| \tag{A.1}$$

and

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_i \left(\frac{\delta_i^{\text{exp}} - \delta_i^{\text{pred}}}{\sigma_i} \right)^2} \tag{A.2}$$

where δ_i^{exp} is the measured chemical shift for nucleus i, δ_i^{pred} is the the predicted chemical shift, σ_i is the expected accuracy of predicted chemical shifts of type of nucleus i, and N is the total number of nuclei used in error calculation.

To facilitate more detailed analyses of differences between reference and computed chemical shifts, PyShifts allows users to select a single row in the `Error Table` (Figure A.2C), which populates a data frame that contains residue name, atom name, measured chemical shifts, computed chemical shifts, and weighted absolute difference for each nucleus of interest for the state associated with the selected row (Figure A.2D). Upon selecting a row in the `Error Table`, the corresponding structure is rendered in the molecular viewer window (Figure A.2E), with the differences between measured and comparison chemical shifts encoded in the spheres at each nucleus of interest. The size of the sphere is proportional to magnitude of the difference and color encodes the sign: red (negative differences) and blue (positive differences). When a given nucleus in the `Chemical Shift Table` is selected, the corresponding nucleus is highlighted in the molecular viewer (Figure A.2F). Highlighting the selected nucleus allows users to effortlessly identify regions in sites that exhibit large discrepancies between measured and computed chemical shifts and which may be outliers. Pyshifts also allows users to control over some aspects of error calculation and rendering in its `Advanced Options` tab, which is discussed in the Supporting Information (Figure B.4).

Multi-model Analysis.

Implemented in PyShifts are several features that facilitate chemical shift-based analysis of multi-state objects loaded in PyMOL.

A.0.0.1 Assigning Conformational Weights.

PyShifts automatically assigns conformational weights to the set of structures using Bayesian maximum entropy (BME) method.[13] BME is used to weight each structure in a multi-state PyMOL object, conditioned on user supplied measured chemical shifts, computed chemical shifts, and expected errors between reference and

computed chemical shifts. BME derived weights are included in the `Error Table` (Figure A.2C).

A.0.0.2 Chemical Shift-Based Clustering.

PyShifts automatically clusters the collection of structures in the reference PyMOL object using their computed chemical shifts as clustering features.[10] One advantage of clustering structures based on their computed chemical shifts rather than their RMSD dissimilarity as is typically done, is that chemical shifts depend only on interatomic distances and are invariant to translation or rotation. As such, no structure alignment is required prior to clustering. Currently, PyShifts uses a K-means clustering algorithm to cluster the set of conformations into a user specified number of clusters before returning the cluster ID for each state in the object and reporting it in the `Error Table` (Figure A.2C). K-means clustering is an unsupervised machine learning method that divides data points into k clusters based on (dis)similarity of their features, here, the computed chemical shifts. PyShifts takes as input the complete structure of each conformation and outputs the K-means determined cluster ID.

A.0.0.3 Sorting Structures.

Each time the `Sort` button is clicked, the `Error Table` is sorted and structures in the molecular viewer are updated to reflect the new order. PyShifts allows users to sort the collection of structures by MAE, RMSE, or R as well as by BME assigned weights or cluster ID (Figure A.2F).

Application Examples

Below we briefly describe three use cases that serve to highlight the utility of PyShifts as a research tool. Below we show applications of PyShifts to NMR ensem-

bles of RNA. Similar examples of applying PyShifts to proteins are presented in the Supporting Information (Figures B.5, B.6, B.7).

Sample Case 1: Detecting Referencing Errors.

The presence of referencing errors in chemical shift data can confound chemical shift-based structural analysis. Accordingly, tools to detect and possibly correct such errors are urgently needed. Here we demonstrate how PyShifts can be used to visually detect the presence of systematic errors in measured chemical shifts when a structural model of the corresponding RNA is available. To accomplish this, we analyzed the first model in the NMR bundle for the 32-nt U6 intramolecular stem-loop (U6 ISL) RNA (PDBID: 1XHP), which is found in the U2/U6 complex of the spliceosome.[14] The chemical shift dataset deposited in the BMRB (BMRB ID: 6320) has been experimentally validated to contain ^{13}C referencing errors.[15]

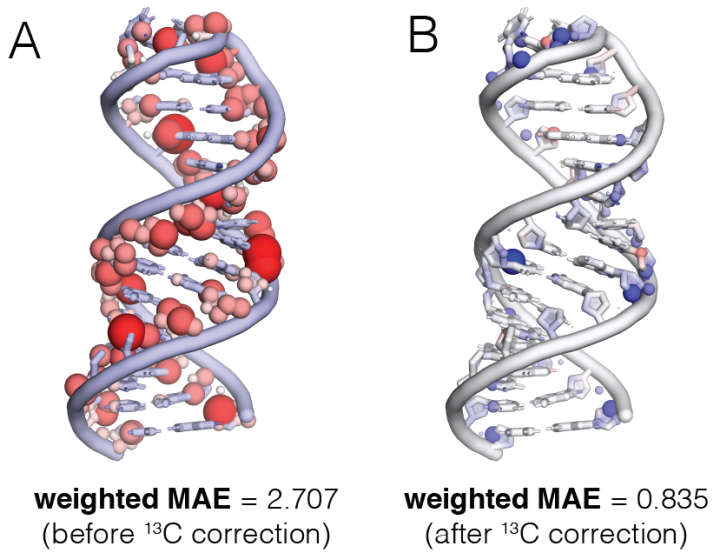

Figure A.3: Visual detection of systematic referencing errors. (A-B) Shown is the projection of the error between measured and computed chemical shifts for the RNA, U6 ISL onto the first model in the corresponding NMR bundle (PDB ID: 1XHP). At each nucleus for which computed and measured chemical shifts are available, PyShifts renders spheres whose radius is proportional to the difference between measured and computed chemical shifts and whose color indicates whether the difference is negative (*red*) or positive (*blue*).

In Figure A.3A, the differences between measured and computed ^{13}C chemical shifts are projected onto the first model in the NMR bundle of U6 ISL RNA, using PyShifts. The consistently large and negative differences between measured and computed chemical shifts (Figure A.3A) are indicative of a systematic offset. Adding a 2.700 ppm offset prior to computing the differences reduces the weighted MAE from 2.707 to 0.835 (Figure A.3B).

Sample Case 2: Identifying Structure that Exhibit the Best Agreement Between Computed and Measured Chemical Shifts.

To illustrate how PyShifts could be used to carry out chemical shift-based analysis of biomolecular ensembles, we used it to analyze an ensemble of the 23-nt RNA, microRNA-20b (miR-20b).[16] Specifically, we used PyShifts to examine a 40-membered ensemble that was composed of the structures from the 20-membered NMR bundle of the free state of miR-20b (PDBID: 2N7X) and the structures from the 20-membered NMR bundle of bound state of miR-20b (PDBID: 2N82). Collectively, the NMR ensemble of the free (Figure A.4A) and bound (Figure A.4B) states of miR-20b indicated that in the presence of protein Rbfox RRM, miR-20b undergoes a 4.33 Å structural change that involves the disruption of several base-pairs in the apical loop region of miR-20b pre-element.[16]

With access to chemical shifts for both free (Figure A.4(A) and bound (Figure A.4(B) states, we used PyShifts to identify structure(s) in the combined ensemble that exhibited the best agreement between measured and computed chemical shifts. Shown in Figure A.4(C,D) is PyShifts' rendering of structures in the combined ensemble that exhibited the lowest weighted MAE and RMSE and the highest Pearson correlation (R), respectively, between computed chemical shifts and measured chemical shifts of the free (Figure A.4C) and bound state (Figure A.4D). For free state, structures that exhibited the best agreement between computed and measured chemical shifts

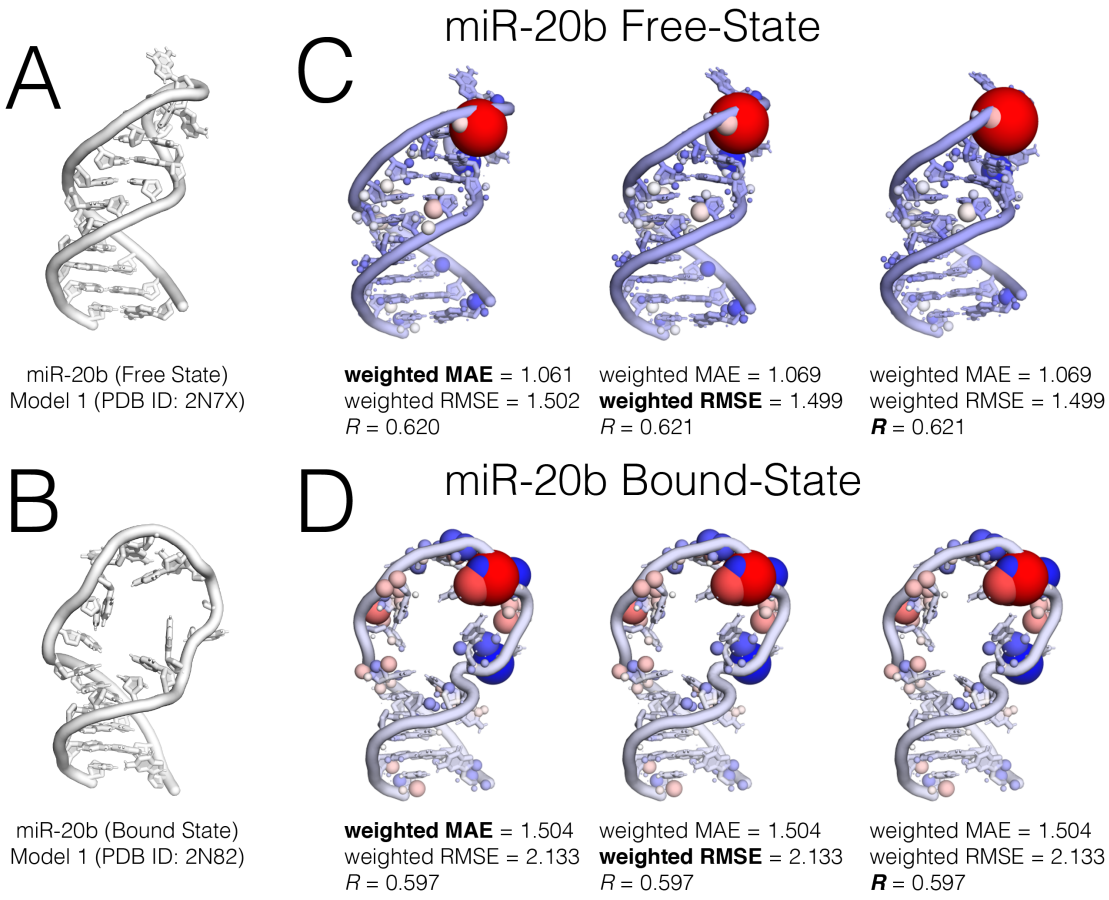

Figure A.4: (A-D) Structures in the combined ensemble miR-20b ensemble (PDB ID: 2N7X (free) and PDB ID: 2N82 (bound)), that exhibited the best between computed chemical shifts and the measured chemical shifts of the *free* (C) and *bound* (D) states, respectively.

as quantified using MAE, RMSE and R were structures from the free state NMR bundle (Figure A.4A, C). Conversely, for bound state, the best structures were from the bound state NMR bundle (Figure A.4B, D).

Sample Case 3: Clustering Structures Based on Their Chemical Shift (Dis)Similarity.

To demonstrate the ability of PyShifts to carry out a structural analysis of multi-state objects in the *absence* of measured chemical shift data, we used it to cluster the 40 structures in the combined ensemble of miR-20b. Within this unsupervised

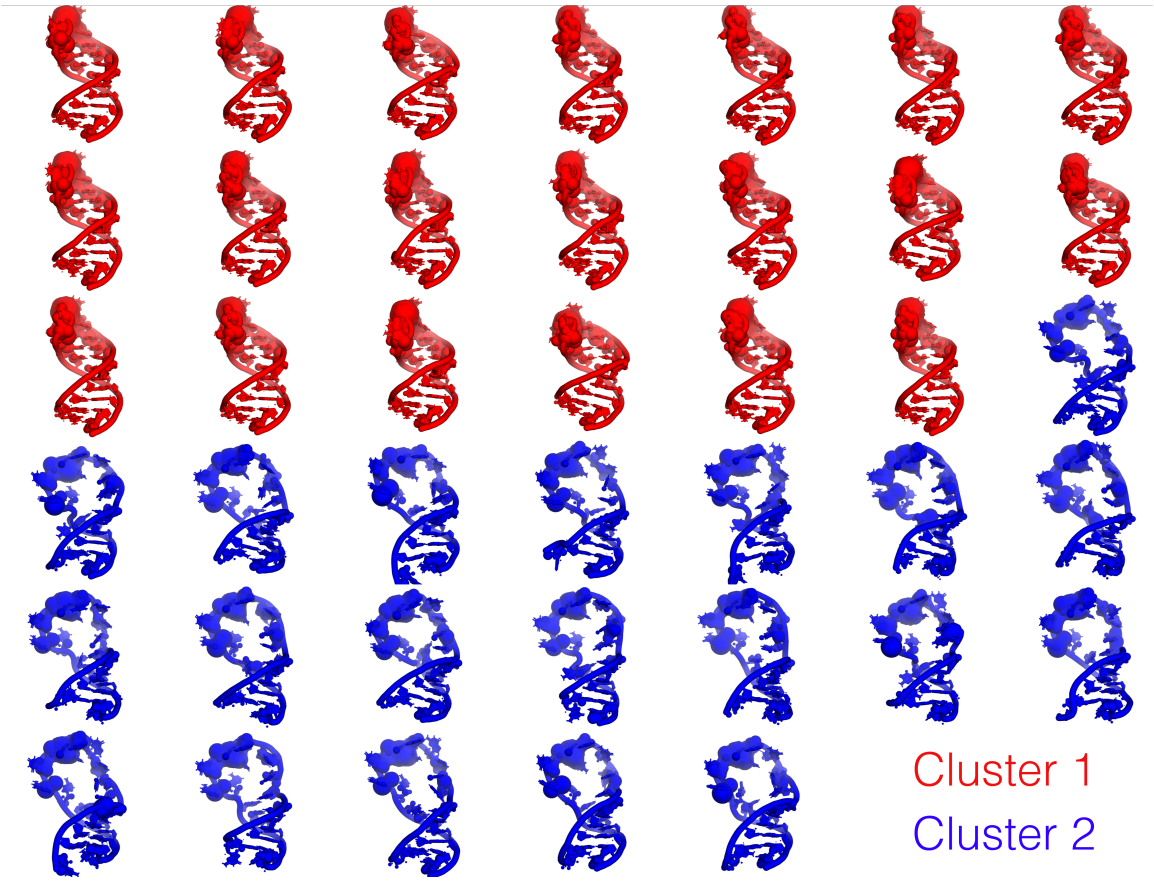

Figure A.5: Results obtained by clustering the structures of the free-state and bound-state structures of miR-20b using their computed chemical shifts as features. After clustering, the structures were sorted in PyShifts based on their cluster ID. As can be seen, clustering the structures based on their *computed* chemical shifts and then sorting them enabled the correct separation of the combined ensemble into two clusters containing the free-state (*red*) and bound-state (*blue*) structures.

machine learning task, we used the computed chemical shifts as clustering features. Shown in Figure A.5 are the chemical shift-based clustering results we obtained using PyShifts. Using the computed chemical shifts of free and bound state of miR20-b as clustering features and the K-means clustering algorithm that we interfaced with PyShifts, we were able to correctly separate free and bound state structures into their two distinct clusters.

References

(1) Clore, G. M.; Schwieters, C. D. *Curr. Opin. Struct. Biol.* **2002**, *12*, 146–153.

(2) Al-Hashimi, H. M. *J. Magn. Reson.* **2013**, *237*, 191–204.

(3) Shen, Y.; Delaglio, F.; Cornilescu, G.; Bax, A. *J. Biomol. NMR* **2009**, *44*, 213–223.

(4) Shen, Y.; Lange, O.; Delaglio, F.; Rossi, P.; Aramini, J. M.; Liu, G.; Eletsky, A.; Wu, Y.; Singarapu, K. K.; Lemak, A., et al. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 4685–4690.

(5) Martin, O. A.; Arnautova, Y. A.; Icazatti, A. A.; Scheraga, H. A.; Vila, J. A. *Proc. Natl. Acad. Sci. U. S. A.* **2013**, *110*, 16826–16831.

(6) Hafsa, N. E.; Berjanskii, M. V.; Arndt, D.; Wishart, D. S. *J. Biomol. NMR* **2018**, *70*, 33–51.

(7) Chen, J. L.; Bellaousov, S.; Tubbs, J. D.; Kennedy, S. D.; Lopez, M. J.; Mathews, D. H.; Turner, D. H. *Biochemistry* **2015**, *54*, 6769.

(8) Sripakdeevong, P.; Cevec, M.; Chang, A. T.; Erat, M. C.; Ziegeler, M.; Zhao, Q.; Fox, G. E.; Gao, X.; Kennedy, S. D.; Kierzek, R., et al. *Nature methods* **2014**, *11*, 413.

(9) Icazatti, A. A.; Loyola, J. M.; Szleifer, I.; Vila, J. A.; Martin, O. A. *PeerJ* **2019**, *7*, e7904.

(10) Engel, E. A.; Anelli, A.; Hofstetter, A.; Paruzzo, F.; Emsley, L.; Ceriotti, M. *Phys. Chem. Chem. Phys.* **2019**, *21*, 23385–23400.

(11) Schrodinger, L. *Version* **2010**, *1*, 0.

(12) Martin, O. A.; Vila, J. A.; Scheraga, H. A. *Bioinformatics* **2012**, *28*, 1538–1539.

(13) Bottaro, S.; Bussi, G.; Kennedy, S. D.; Turner, D. H.; Lindorff-Larsen, K. *Science advances* **2018**, *4*, eaar8521.

(14) Sashital, D. G.; Cornilescu, G.; Butcher, S. E. *Nat. Struct. Mol. Biol.* **2004**, *11*, 1237.

(15) Aeschbacher, T.; Schubert, M.; Allain, F. H.-T. *J. Biomol. NMR* **2012**, *52*, 179–190.

(16) Chen, Y.; Zubovic, L.; Yang, F.; Godin, K.; Pavelitz, T.; Castellanos, J.; Macchi, P.; Varani, G. *Nucleic Acids Res.* **2016**, *44*, 4381–4395.

APPENDIX B

Supporting Information

Supporting Tables

Table B.**1:** Train set RNA information

PDBIDs	Length	Number of unpaired residues	Number of paired residues
1A60	44	16	28
1HWQ	30	12	18
1JO7	31	11	20
1KKA	17	5	12
1L1W	29	9	20
1LC6	24	8	16
1LDZ	30	10	20
1MFY	31	13	18
1NA2	30	14	16
1NC0	24	8	16
1OW9	23	9	14
1PJY	22	4	18
1Q75	15	5	10
1R7W	34	10	24
1R7Z	34	10	24
1SCL	29	15	14
1SY4	24	8	16
1SYZ	24	8	16
1UUU	19	7	12

PDBIDs	Length	Number of unpaired residues	Number of paired residues
1XHP	32	10	22
1YMO	47	17	30
1YSV	27	5	22
1Z2J	45	7	38
1Z30	18	4	14
1ZC5	41	7	34
28SP	28	12	16
28SR	28	12	16
2F87	12	4	8
2FDT	36	8	28
2GVO	18	6	12
2JR4	17	7	10
2JTP	34	8	26
2JWV	29	11	18
2JYM	22	4	18
2K66	22	4	18
2KEZ	24	8	16
2KF0	24	8	16
2KOC	14	4	10
2KXM	27	11	16
2KZL	55	27	28
2L3E	35	11	24
2L5Z	26	8	18
2L6I	16	4	12
2L8H	29	9	20
2LAC	17	7	10
2LBJ	17	3	14
2LBK	17	5	12
2LBL	17	7	10
2LDL	27	9	18
2LDT	31	9	22
2LHP	37	7	30
2LI4	32	4	28
2LJJ	27	9	18
2LK3	24	6	18
2LP9	16	4	12
2LPA	15	3	12
2LPS	34	6	28
2LQZ	27	9	18
2LU0	49	17	32
2LUB	37	7	30
2LUN	28	10	18
2LV0	24	8	16

PDBIDs	Length	Number of unpaired residues	Number of paired residues
2M12	23	9	14
2M21	21	7	14
2M22	23	7	16
2M4W	17	7	10
2M5U	22	4	18
2M8K	48	16	32
2MEQ	19	7	12
2MFD	19	5	14
2MHI	53	13	40
2MIS	26	8	18
2MNC	16	6	10
2MTJ	47	17	30
2MXL	39	13	26
2N1Q	155	49	106
2N2O	23	11	12
2N2P	23	11	12
2N3Q	62	20	42
2N3R	62	20	42
2N4L	53	9	44
2N6S	36	6	30
2N6T	42	16	26
2N6W	68	20	48
2N6X	43	13	30
2NBY	39	9	30
2NBZ	40	10	30
2NC0	28	8	20
2NC1	67	17	50
2NCI	28	12	16
2O33	20	8	12
2QH2	24	6	18
2QH3	23	7	16
2QH4	18	6	12
2RVO	34	8	26
2Y95	14	4	10
4A4S	22	4	18
4A4T	22	4	18
4A4U	22	4	18
5A17	32	10	22
5A18	32	10	22
5IEM	57	13	44
5KH8	47	19	28
5KQE	36	10	26
5UF3	23	7	16

PDBIDs	Length	Number of unpaired residues	Number of paired residues
5UZT	31	11	20
5V16	41	11	30
5WQ1	23	5	18

Table B.**2:** Optimized hyperparameters for CS2BPS classifiers (one of the six runs)

PDBIDs	Batch size	Dropout rate	Epochs	Loss	Learning rate	Optimization
1A60	256	0	25	BCE	0.001	RMSprop
1HWQ	128	0	25	BCE	0.001	RMSprop
1JO7	256	0.1	50	logcosh	0.01	RMSprop
1KKA	128	0.1	25	logcosh	0.001	Adam
1L1W	128	0.1	25	BCE	0.001	Adam
1LC6	256	0	50	BCE	0.001	Adam
1LDZ	256	0.1	25	BCE	0.001	Adam
1MFY	128	0.1	25	BCE	0.001	Adam
1NA2	128	0	25	BCE	0.001	Adam
1NC0	128	0.1	25	BCE	0.001	Adam
1OW9	256	0	25	BCE	0.001	Adam
1PJY	256	0	25	BCE	0.001	Adam
1Q75	256	0.1	50	BCE	0.001	RMSprop
1R7W	128	0.1	50	BCE	0.001	Adam
1R7Z	256	0	50	BCE	0.001	RMSprop
1SCL	256	0.1	50	BCE	0.001	Adam
1SY4	128	0	25	BCE	0.001	Adam
1SYZ	128	0.1	25	BCE	0.001	Adam
1UUU	128	0	25	BCE	0.001	RMSprop
1XHP	128	0.1	25	BCE	0.001	Adam
1YMO	128	0	25	BCE	0.001	RMSprop
1YSV	128	0.1	25	BCE	0.001	Adam
1Z2J	128	0	25	BCE	0.001	Adam
1Z30	128	0	25	logcosh	0.001	Adam
1ZC5	256	0.1	50	BCE	0.001	Adam
28SP	256	0.1	50	BCE	0.001	RMSprop
28SR	128	0.1	25	BCE	0.001	Adam
2F87	128	0	25	BCE	0.001	Adam
2FDT	256	0.1	50	logcosh	0.001	Adam
2GVO	128	0	50	BCE	0.001	Adam
2JR4	128	0	25	BCE	0.001	Adam
2JTP	128	0	25	logcosh	0.001	RMSprop
2JWV	256	0	25	BCE	0.001	Adam
2JYM	128	0.1	25	logcosh	0.001	RMSprop

PDBIDs	Batch size	Dropout rate	Epochs	Loss	Learning rate	Optimization
2K66	128	0	25	BCE	0.001	RMSprop
2KEZ	256	0	50	BCE	0.001	Adam
2KF0	128	0.1	25	BCE	0.001	Adam
2KOC	128	0.1	25	BCE	0.001	RMSprop
2KXM	256	0.1	50	BCE	0.001	RMSprop
2KZL	128	0.1	50	logcosh	0.001	Adam
2L3E	128	0.1	25	logcosh	0.001	RMSprop
2L5Z	128	0.1	50	logcosh	0.01	RMSprop
2L6I	128	0.1	25	BCE	0.001	RMSprop
2L8H	256	0.1	25	BCE	0.001	RMSprop
2LAC	256	0.1	50	BCE	0.01	Adam
2LBJ	128	0.1	25	BCE	0.001	Adam
2LBK	128	0.1	25	BCE	0.001	RMSprop
2LBL	128	0.1	25	BCE	0.001	RMSprop
2LDL	128	0	25	BCE	0.001	RMSprop
2LDT	128	0.1	25	BCE	0.001	Adam
2LHP	128	0	50	logcosh	0.001	Adam
2LI4	256	0	25	BCE	0.001	Adam
2LJJ	256	0.1	50	BCE	0.001	RMSprop
2LK3	128	0	25	BCE	0.001	Adam
2LP9	128	0.1	25	BCE	0.001	RMSprop
2LPA	128	0	50	BCE	0.01	Adam
2LPS	128	0.1	25	BCE	0.001	Adam
2LQZ	256	0	50	BCE	0.001	Adam
2LU0	256	0	50	BCE	0.001	Adam
2LUB	256	0.1	50	BCE	0.001	RMSprop
2LUN	128	0.1	50	BCE	0.001	RMSprop
2LV0	128	0.1	25	BCE	0.001	Adam
2M12	128	0	50	BCE	0.01	Adam
2M21	256	0.1	25	BCE	0.001	Adam
2M22	128	0.1	25	BCE	0.001	Adam
2M4W	128	0.1	25	BCE	0.001	Adam
2M5U	128	0	25	logcosh	0.001	Adam
2M8K	256	0.1	50	BCE	0.001	RMSprop
2MEQ	128	0.1	25	BCE	0.001	RMSprop
2MFD	256	0	50	BCE	0.001	Adam
2MHI	256	0.1	50	BCE	0.001	Adam
2MIS	128	0	25	BCE	0.001	RMSprop
2MNC	128	0.1	25	BCE	0.001	RMSprop
2MTJ	128	0.1	25	logcosh	0.001	Adam
2MXL	128	0	25	BCE	0.001	RMSprop
2N1Q	128	0.1	25	BCE	0.001	RMSprop
2N2O	128	0	25	BCE	0.001	RMSprop

PDBIDs	Batch size	Dropout rate	Epochs	Loss	Learning rate	Optimization
2N2P	128	0.1	25	BCE	0.001	Adam
2N3Q	128	0	25	BCE	0.001	RMSprop
2N3R	128	0.1	25	BCE	0.001	Adam
2N4L	128	0	25	BCE	0.001	Adam
2N6S	128	0.1	50	logcosh	0.001	Adam
2N6T	256	0.1	50	BCE	0.001	RMSprop
2N6W	128	0	25	logcosh	0.001	RMSprop
2N6X	256	0	25	BCE	0.001	Adam
2NBY	128	0	25	BCE	0.001	RMSprop
2NBZ	256	0.1	25	logcosh	0.001	RMSprop
2NC0	128	0.1	25	BCE	0.001	Adam
2NC1	256	0	25	BCE	0.001	Adam
2NCI	128	0.1	25	BCE	0.001	RMSprop
2O33	128	0.1	25	BCE	0.001	Adam
2QH2	256	0.1	25	BCE	0.001	Adam
2QH3	256	0	50	BCE	0.001	RMSprop
2QH4	128	0.1	25	BCE	0.001	Adam
2RVO	128	0.1	25	BCE	0.001	RMSprop
2Y95	128	0.1	25	BCE	0.001	RMSprop
4A4S	256	0	25	BCE	0.01	Adam
4A4T	128	0.1	25	BCE	0.001	Adam
4A4U	128	0.1	50	BCE	0.001	Adam
5A17	128	0	25	BCE	0.001	Adam
5A18	128	0	25	BCE	0.001	RMSprop
5IEM	128	0.1	25	logcosh	0.001	RMSprop
5KH8	256	0	50	BCE	0.001	Adam
5KQE	256	0	50	BCE	0.001	Adam
5UF3	128	0.1	25	BCE	0.001	RMSprop
5UZT	128	0	50	BCE	0.01	RMSprop
5V16	256	0.1	50	BCE	0.001	Adam
5WQ1	128	0	25	BCE	0.001	RMSprop

[1] BCE is binary cross-entropy loss;

Table B.**3:** CS2BPS predictions for the fluoride riboswitch (PDBID: 5KH8)

Residue	Run						Variance[1]
	1	2	3	4	5	6	
1	0.0001	0.0000	0.0005	0.0010	0.0003	0.0000	0.06
2	0.0001	0.0000	0.0002	0.0017	0.0021	0.0000	0.07
3	0.0000	0.0000	0.0001	0.0001	0.0001	0.0000	0.05
4	0.0003	0.0000	0.0001	0.0002	0.0009	0.0000	0.06
5	**0.0011**	**0.0000**	**0.0009**	**0.0040**	**0.0173**	**0.2968**	**0.24**

Residue	Run						Variance[1]
	1	2	3	4	5	6	
6	0.0000	0.0000	0.0000	0.0004	0.0009	0.0000	0.06
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.04
8	0.0000	0.0000	0.0000	0.0008	0.0006	0.0000	0.06
9	0.0000	0.0000	0.0000	0.0000	0.0001	0.0000	0.05
10	0.0178	0.0004	0.0028	0.0098	0.0394	0.0023	0.12
11	0.0250	0.0155	0.0039	0.1050	0.0429	0.0028	0.15
12	**0.0194**	**0.0025**	**0.2173**	**0.0715**	**0.1375**	**0.0899**	**0.20**
13	0.0004	0.0006	0.0028	0.0095	0.0019	0.0000	0.09
14	0.0010	0.0003	0.0023	0.0029	0.0191	0.0028	0.10
15	0.0000	0.0000	0.0000	0.0004	0.0028	0.0000	0.07
16	0.0000	0.0001	0.0005	0.0029	0.0008	0.0001	0.07
17	0.9959	0.9710	0.9932	0.9843	0.9784	1.0000	0.11
18	0.9997	1.0000	0.9984	0.9969	0.9982	1.0000	0.07
19	0.9998	1.0000	0.9989	0.9999	0.9974	1.0000	0.07
20	**0.3816**	**0.7329**	**0.9966**	**0.9417**	**0.8826**	**0.0013**	**0.53**
21	0.9999	0.9995	0.9898	0.9969	0.9983	1.0000	0.09
22	0.9780	0.9982	0.9921	0.9933	0.9140	0.9998	0.15
23	0.0052	0.0014	0.0026	0.0184	0.0128	0.0088	0.10
24	0.0339	0.0004	0.0017	0.0174	0.0130	0.0007	0.12
25	0.0070	0.0452	0.0244	0.0587	0.0218	0.0001	0.13
26	0.0017	0.0000	0.0142	0.0031	0.0196	0.0000	0.10
27	0.9993	0.9999	0.9931	0.9992	0.9962	1.0000	0.08
28	0.9975	1.0000	0.9990	0.9998	0.9978	0.9993	0.07
29	1.0000	1.0000	0.9991	0.9994	0.9998	1.0000	0.06
30	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	0.04
31	0.0001	0.0000	0.0002	0.0002	0.0815	0.0000	0.15
32	0.0000	0.0000	0.0000	0.0007	0.0018	0.0000	0.07
33	0.0004	0.0001	0.0004	0.0040	0.0042	0.0001	0.08
34	0.1586	0.0050	0.0558	0.0726	0.0613	0.0080	0.17
35	0.0001	0.0000	0.0001	0.0007	0.0035	0.0000	0.08
36	**0.2304**	**0.0078**	**0.0620**	**0.5117**	**0.5823**	**0.2636**	**0.34**
37	**0.0068**	**0.0000**	**0.0005**	**0.0014**	**0.2861**	**0.0080**	**0.23**
38	0.0010	0.0000	0.0000	0.0159	0.0057	0.0423	0.12
39	**0.0002**	**0.0028**	**0.0001**	**0.4659**	**0.0036**	**0.0493**	**0.30**
40	**0.7184**	**0.0193**	**0.0048**	**0.5181**	**0.2921**	**0.1885**	**0.40**
41	**0.0420**	**0.0444**	**0.1527**	**0.2700**	**0.1270**	**0.7488**	**0.38**
42	**0.9394**	**0.2242**	**0.6976**	**0.8235**	**0.3449**	**0.8605**	**0.41**
43	0.0031	0.0007	0.0077	0.0602	0.0542	0.0001	0.14
44	**0.7708**	**0.9962**	**0.9629**	**0.8985**	**0.8875**	**0.8598**	**0.20**
45	0.0002	0.0013	0.0031	0.0681	0.0101	0.0001	0.14
46	0.0013	0.0001	0.0008	0.0070	0.0116	0.0000	0.09

Residue	Run						Variance[1]
	1	2	3	4	5	6	
47	0.0001	0.0000	0.0000	0.0004	0.0012	0.0000	0.07

[1] Here, variance was calculated using $-1/\log$(prediction variance). The bold lines represent "outlier" residues with high CS2BPS prediction variance (whose prediction variance was $> 1.5 \times$ IQR (the interquartile range).

Table B.**4:** CS2BPS predictions for the simian immunodeficiency virus (SIV) RNA (PDBID: 2JTP)

Residue	Run						Variance[1]
	1	2	3	4	5	6	
1	0.0015	0.0010	0.0056	0.0096	0.0055	0.0001	0.09
2	0.0553	0.0076	0.0114	0.0242	0.0092	0.0070	0.13
3	0.0721	0.0449	0.0891	0.0976	0.0678	0.1071	0.13
4	0.0018	0.0006	0.0106	0.0180	0.0092	0.0028	0.10
5	0.0021	0.0006	0.0036	0.0100	0.0071	0.0013	0.09
6	0.0008	0.0000	0.0007	0.0013	0.0006	0.0005	0.06
7	0.0016	0.0000	0.0032	0.0116	0.0015	0.0026	0.09
8	0.0217	0.0023	0.0178	0.0346	0.0088	0.0050	0.11
9	0.0048	0.0002	0.0272	0.0112	0.0048	0.0025	0.11
10	0.0043	0.0016	0.0220	0.0598	0.0071	0.0037	0.13
11	0.9373	0.9870	0.8510	0.8956	0.9090	0.9304	0.16
12	**0.9630**	**0.9909**	**0.7291**	**0.8032**	**0.8493**	**0.4831**	**0.30**
13	**0.9608**	**0.8440**	**0.4732**	**0.5568**	**0.7560**	**0.4809**	**0.32**
14	0.8537	0.9873	0.6945	0.7748	0.8013	0.8840	0.22
15	**0.7419**	**0.5396**	**0.7312**	**0.4204**	**0.5434**	**0.8502**	**0.27**
16	**0.4963**	**0.9805**	**0.5220**	**0.4225**	**0.3977**	**0.9040**	**0.37**
17	0.9962	0.9997	0.9962	0.9975	0.9982	0.9957	0.08
18	0.9861	0.9996	0.9931	0.9936	0.9942	0.9971	0.09
19	0.9718	0.9979	0.9923	0.9990	0.9969	0.9774	0.11
20	0.0157	0.0792	0.0220	0.0087	0.0118	0.0213	0.14
21	0.0105	0.0033	0.0227	0.3151	0.1347	0.0113	0.24
22	0.9912	0.9998	0.9893	0.9854	0.9913	0.8049	0.19
23	0.7439	0.8314	0.8475	0.7549	0.6228	0.7294	0.20
24	0.8946	0.9819	0.8687	0.9284	0.9243	0.9144	0.15
25	**0.2970**	**0.4055**	**0.2289**	**0.1883**	**0.1647**	**0.5186**	**0.25**
26	0.0269	0.0059	0.1526	0.0655	0.0597	0.0393	0.17
27	0.0144	0.0036	0.0229	0.0173	0.0142	0.0674	0.13
28	0.0106	0.0011	0.0372	0.0148	0.0146	0.0271	0.11
29	0.0181	0.0042	0.0678	0.0181	0.0197	0.0164	0.13
30	0.1026	0.0207	0.1014	0.0956	0.0718	0.0223	0.15
31	0.1394	0.0071	0.1098	0.0323	0.2332	0.0284	0.20

Residue	Run						Variance[1]
	1	2	3	4	5	6	
32	0.0065	0.0004	0.0208	0.0181	0.0115	0.0061	0.10
33	0.0014	0.0002	0.0091	0.0093	0.0023	0.0027	0.09
34	0.0263	0.0009	0.0323	0.0225	0.0334	0.0124	0.11

[1] Here, variance was calculated using $-1/\log$(prediction variance). The bold lines represent "outlier" residues with high CS2BPS prediction variance (whose prediction variance was $> 1.5 \times$ IQR (the interquartile range).

Table B.5: CS2BPS predictions for the group II intron Sc.ai5γ RNA (PDBID: 2LU0)

Residue	Run						Variance[1]
	1	2	3	4	5	6	
1	0.0019	0.0071	0.0021	0.0006	0.0000	0.0021	0.08
2	0.0000	0.0000	0.0003	0.0003	0.0000	0.0003	0.06
3	0.0000	0.0000	0.0006	0.0024	0.0000	0.0004	0.07
4	0.1217	0.0074	0.1288	0.2312	0.0259	0.2124	0.21
5	**0.3120**	**0.7393**	**0.4139**	**0.4779**	**0.1860**	**0.4989**	**0.30**
6	**0.9402**	**0.7611**	**0.5340**	**0.8485**	**0.9698**	**0.9211**	**0.28**
7	**0.2835**	**0.4900**	**0.5154**	**0.9249**	**0.6068**	**0.6851**	**0.33**
8	**0.1511**	**0.0571**	**0.1356**	**0.4864**	**0.0001**	**0.1577**	**0.28**
9	0.0003	0.0001	0.0142	0.0031	0.0000	0.0047	0.10
10	**0.1476**	**0.9891**	**0.4186**	**0.2001**	**0.0000**	**0.2334**	**0.48**
11	0.0307	0.0015	0.1826	0.1690	0.0050	0.1927	0.21
12	0.9992	0.9995	0.9797	0.9459	1.0000	0.9756	0.13
13	0.0330	0.0262	0.1051	0.0246	0.1489	0.2349	0.20
14	**0.4399**	**0.4969**	**0.1999**	**0.0375**	**0.0001**	**0.2801**	**0.31**
15	1.0000	1.0000	0.9998	0.9973	1.0000	0.9955	0.08
16	1.0000	1.0000	1.0000	0.9986	1.0000	0.9972	0.07
17	0.9999	0.9999	0.9915	0.9877	1.0000	0.9862	0.10
18	1.0000	1.0000	1.0000	0.9994	1.0000	0.9997	0.06
19	0.0002	0.0001	0.0085	0.0004	0.0000	0.0036	0.09
20	0.0761	0.0510	0.3046	0.1181	0.0000	0.1899	0.23
21	0.9438	0.9967	0.8031	0.9051	1.0000	0.7763	0.21
22	0.9759	0.9935	0.9817	0.9686	0.9999	0.9475	0.13
23	0.9926	0.9513	0.8415	0.9305	0.9792	0.8993	0.17
24	0.0019	0.1350	0.0813	0.0312	0.1470	0.3329	0.23
25	0.0033	0.0006	0.0180	0.0104	0.0000	0.0274	0.11
26	0.0014	0.0005	0.0020	0.0022	0.0000	0.0033	0.07
27	0.0001	0.0002	0.0028	0.0004	0.0000	0.0015	0.07
28	0.0001	0.0000	0.0052	0.0027	0.0000	0.0055	0.08
29	0.9064	0.9965	0.9598	0.9475	1.0000	0.8572	0.17
30	0.9988	1.0000	0.9974	0.9871	1.0000	0.9882	0.10

Residue	Run						Variance[1]
	1	2	3	4	5	6	
31	1.0000	1.0000	0.9999	0.9947	1.0000	0.9991	0.08
32	1.0000	1.0000	1.0000	0.9882	1.0000	0.9985	0.09
33	0.0000	0.0000	0.0002	0.0002	0.0000	0.0000	0.05
34	0.0000	0.0000	0.0005	0.0003	0.0000	0.0025	0.07
35	0.0000	0.0001	0.0011	0.0026	0.0000	0.0020	0.07
36	0.0000	0.0000	0.0009	0.0043	0.0000	0.0007	0.08
37	**0.9800**	**0.9093**	**0.8339**	**0.8825**	**0.5311**	**0.7985**	**0.27**
38	0.0085	0.0020	0.0357	0.0183	0.0000	0.0517	0.13
39	0.0002	0.0007	0.0055	0.0170	0.0000	0.0023	0.10
40	0.0038	0.0014	0.0111	0.0012	0.0000	0.0067	0.09
41	0.0007	0.0005	0.0153	0.0007	0.0000	0.0089	0.10
42	0.9659	0.9917	0.8497	0.7930	0.9334	0.8147	0.20
43	0.9999	0.9999	0.9976	0.9959	1.0000	0.9915	0.09
44	0.9999	0.9999	0.9991	0.9942	1.0000	0.9911	0.09
45	0.0056	0.0015	0.0373	0.0030	0.0000	0.0099	0.12
46	0.0048	0.0053	0.0345	0.0857	0.0000	0.0532	0.15
47	0.0337	0.0293	0.0442	0.0055	0.0000	0.0290	0.12
48	0.0041	0.0200	0.0295	0.0673	0.0000	0.0145	0.13
49	0.0030	0.0120	0.0054	0.0124	0.0000	0.0080	0.09

[1] Here, variance was calculated using $-1/\log$(prediction variance). The bold lines represent "outlier" residues with high CS2BPS prediction variance (whose prediction variance was $> 1.5 \times$ IQR (the interquartile range).

Table B.**6:** Secondary structure prediction accuracy with and without CS2BPS predictions

Algorithms	With CS2BPS predictions		Without CS2BPS predictions	
	TPR	PPV	TPR	PPV
Fold	0.96	0.95	0.94	0.93
ProbKnot	0.96	0.93	0.95	0.92
MaxExpect	0.96	0.96	0.94	0.93

[1] TPR is defined as the fraction of base-pairs in the predicted structure that also appeared in the NMR-derived structure. PPV is defined as the fraction of base-pairs in the NMR-derived structure that also appeared in the predicted structure.

[2] TPR and PPV were calculated using the program scorer in the *RNAstructure* suite; For Fold, ProbKnot and MaxExpect, values obtained with and without using CS2BPS predictions as folding restraints are shown. Using Fold, ProbKnot and MaxExpect with and without CS2BPS predictions, we obtained 6 secondary structure models. Then the base-pairing status consistency scores between these 6 secondary structure models and CS2BPS predictions were calculated and the model with the highest consistency score was selected as the CS-Fold secondary structure model.

Table B.7: Imputation accuracy for RNAs with both ^{1}H and ^{13}C chemical shifts and only ^{1}H chemical shifts

Both ^{1}H and ^{13}C chemical shifts				Only ^{1}H chemical shifts	
Nucleus type	Error (ppm)	Nucleus type	Error (ppm)	Nucleus type	Error (ppm)
C1'	1.12	H1'	0.20	H1'	0.18
C2'	0.98	H2'	0.18	H2'	0.19
C3'	1.31	H3'	0.17	H3'	0.20
C4'	0.84	H4'	0.11	H4'	0.17
C5'	1.14	H5'	0.22	H5'	0.18
C2	1.68	H2	0.38	H2	0.35
C5	1.65	H5	0.20	H5	0.19
C6	0.88	H6	0.13	H6	0.20
C8	1.63	H8	0.25	H8	0.33
		H5"	0.18	H5"	0.12
Mean	1.25		0.20		0.21

[1] To estimate the magnitude of errors that are introduced into the chemical shift data via MICE imputation, we randomly removed 10% of measured chemical shifts from training set for each nucleus type and then imputed the remaining data set using MICE. We then calculated the mean absolute error (MAE) between the measured and imputed chemical shifts.

Table B.8: Chemical shift error analysis for 5KH8 without long range base pairs

Structure	MAE	RMSE	R	BME weight
Decoy 1	0.72	1.27	0.91	0.00
Decoy 2	0.72	1.24	0.90	0.00
Decoy 3	0.70	1.26	0.91	0.09
Decoy 4	0.71	1.24	0.91	0.00
Decoy 5	0.75	1.32	0.90	0.00
Decoy 6	0.69	1.21	0.91	0.00
Decoy 7	0.71	1.26	0.91	0.07
Decoy 8	0.72	1.23	0.90	0.00
Decoy 9	0.70	1.23	0.91	0.00
Decoy 10	0.68	1.20	0.91	0.16
Decoy 11	0.77	1.37	0.91	0.03
Decoy 12	0.75	1.31	0.91	0.02
Decoy 13	0.76	1.33	0.90	0.01
Decoy 14	0.73	1.31	0.91	0.16
DSSR Structure	0.70	1.22	0.92	0.45

Table B**.9:** SS2CS testing error of different models.

Nucleus	Linear	Ridge	SVR	RF	ET	GB
C1'	0.84	0.84	0.88	0.79	0.88	0.79
C2'	0.68	0.67	0.66	0.67	0.74	0.65
C3'	1.38	1.38	1.43	1.27	1.42	1.29
C4'	0.80	0.80	0.86	0.78	0.86	0.75
C5'	0.98	0.98	1.06	0.98	1.08	0.94
C2	1.28	1.25	1.35	1.16	1.37	1.28
C5	1.07	1.06	3.03	0.97	1.08	1.01
C6	0.73	0.74	1.08	0.67	0.78	0.66
C8	0.94	0.94	1.43	0.90	1.02	0.87
Mean	0.97	0.96	1.31	0.91	1.03	0.92
H1'	0.15	0.15	0.15	0.11	0.13	0.13
H2'	0.13	0.13	0.14	0.12	0.13	0.12
H3'	0.13	0.13	0.13	0.11	0.12	0.12
H4'	0.10	0.10	0.10	0.09	0.10	0.09
H5'	0.17	0.16	0.16	0.13	0.15	0.14
H5"	0.15	0.15	0.14	0.12	0.13	0.12
H2	0.17	0.17	0.26	0.16	0.17	0.16
H5	0.12	0.12	0.14	0.11	0.12	0.12
H6	0.10	0.10	0.11	0.10	0.10	0.10
H8	0.15	0.15	0.20	0.13	0.15	0.14
Mean	0.14	0.14	0.15	0.12	0.13	0.12

Supporting Figures

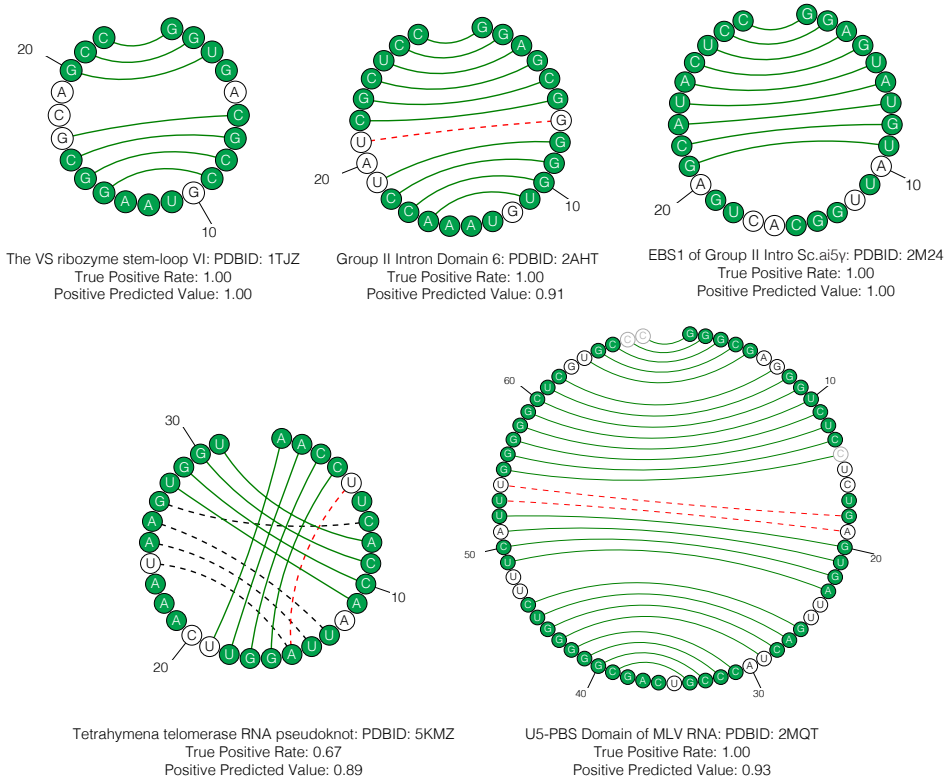

Figure B.**1:** RNAs that have been removed from CS2BPS model training set (Part 1).

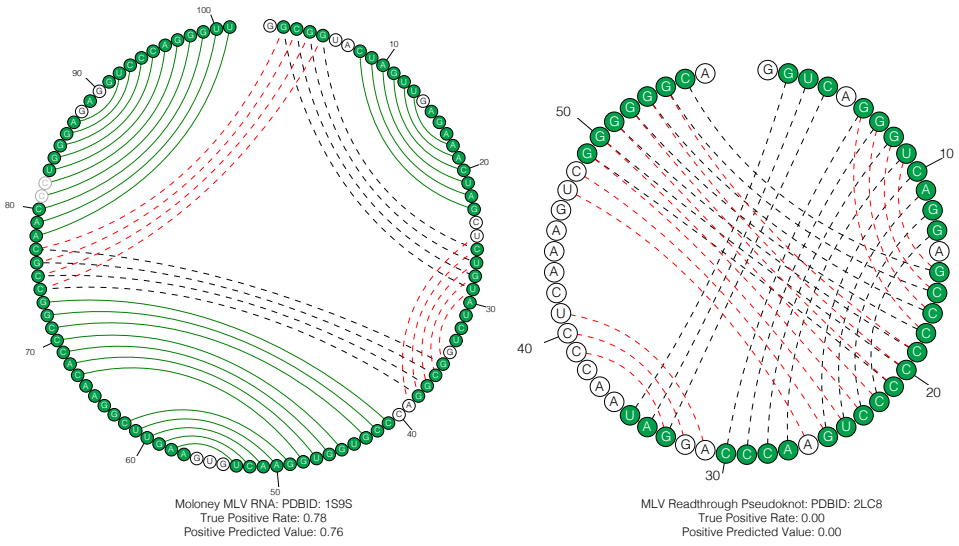

Figure B.**2:** RNAs that have been removed from CS2BPS model training set (Part 2).

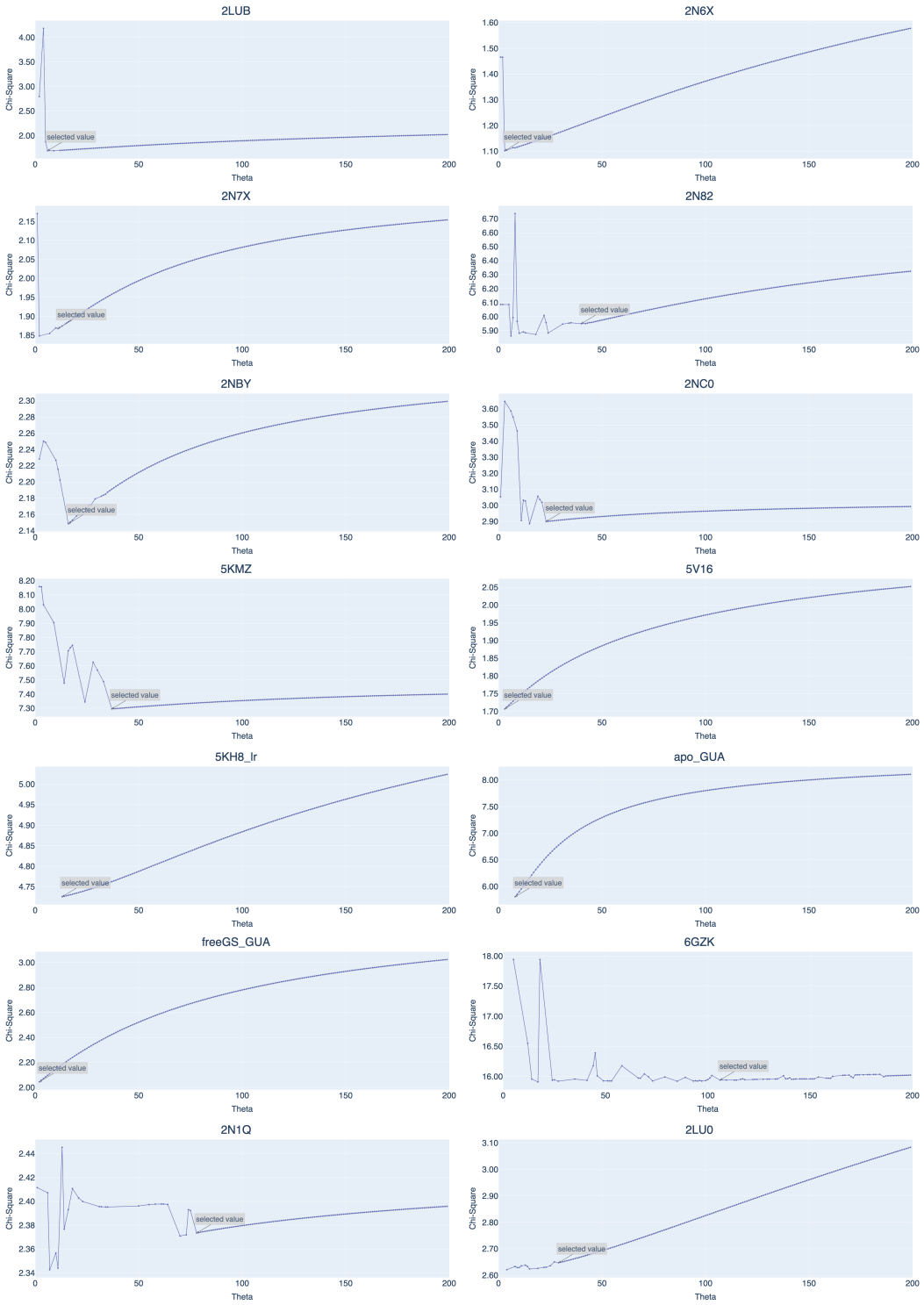

Figure B.**3:** Optimized θ for the remaining RNAs.

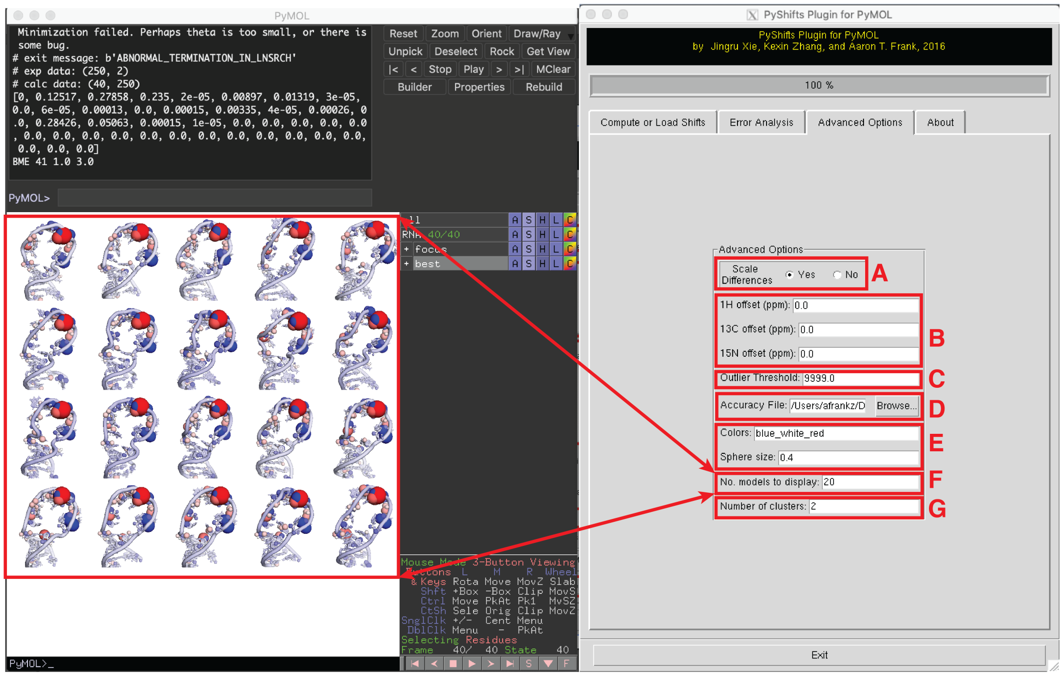

Figure B.4: PyShifts' `Advanced Options` interface. In the `Advanced Options` tab, users can: (A) toggle whether weighted differences should be computed; (B) add offsets to measured chemical shifts; (C) set the outlier threshold value; (D) specify path to the file containing expected chemical shift errors (i.e., σ values in Eq. 1 and 2); (E) change the color palette and size of the spheres used to visualize computed chemical shift differences; (F) set the number of structures in the `Error Table` to display; (G) set number of clusters PyShifts should use for K-means clustering.

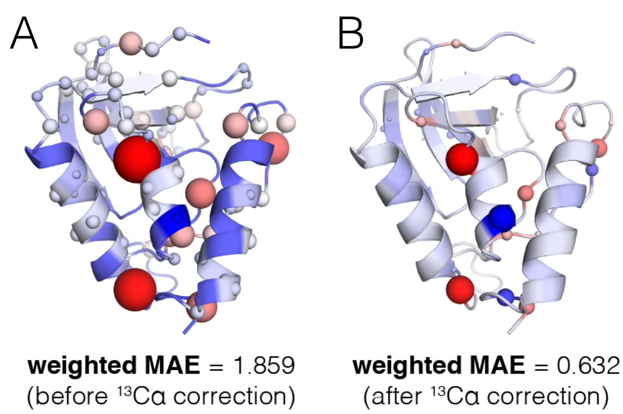

Figure B.5: Visual detection of systematic referencing errors in protein. (A-B) Shown is the projection of the error between measured and computed chemical shifts for the T120S mutant of the *Staphylococcal* nuclease onto the X-ray structure (PDB ID: 2EYO). At each nucleus for which computed and measured chemical shifts are available, PyShifts renders spheres whose radius is proportional to the difference between measured and computed chemical shifts and whose color indicates whether the difference is negative (*red*) or positive (*blue*).

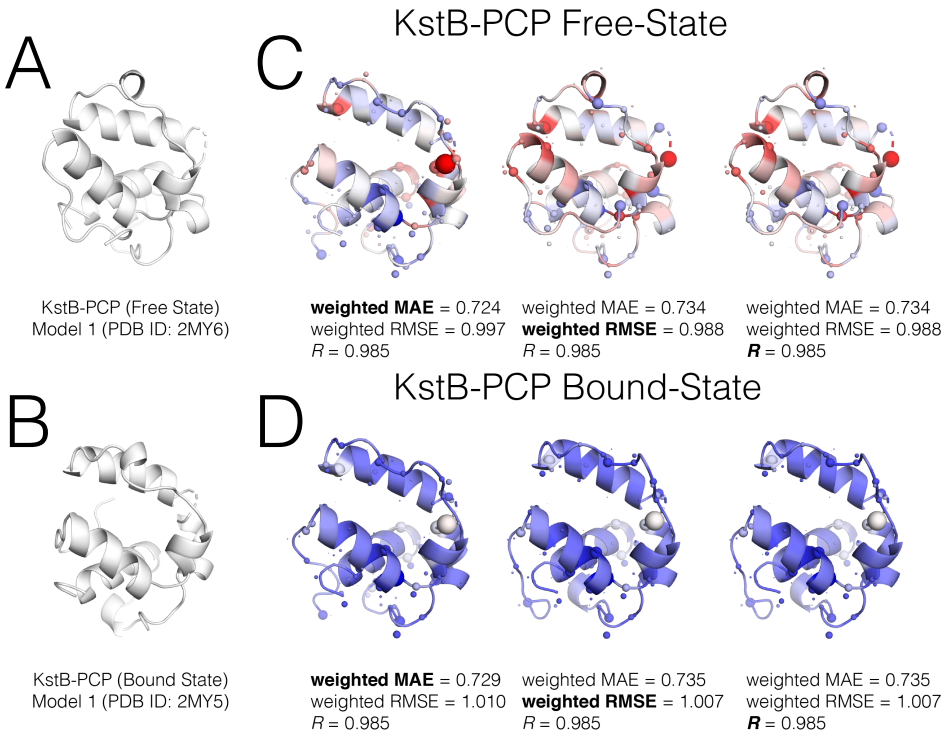

Figure B.**6:** (A-D) Structures in the combined ensemble of the protein KstB-PCP (PDB ID: 2MY6 (free) and PDB ID: 2MY5 (bound)), that exhibited the best between computed chemical shifts and the measured chemical shifts of the *free* (C) and *bound* (D) states, respectively.

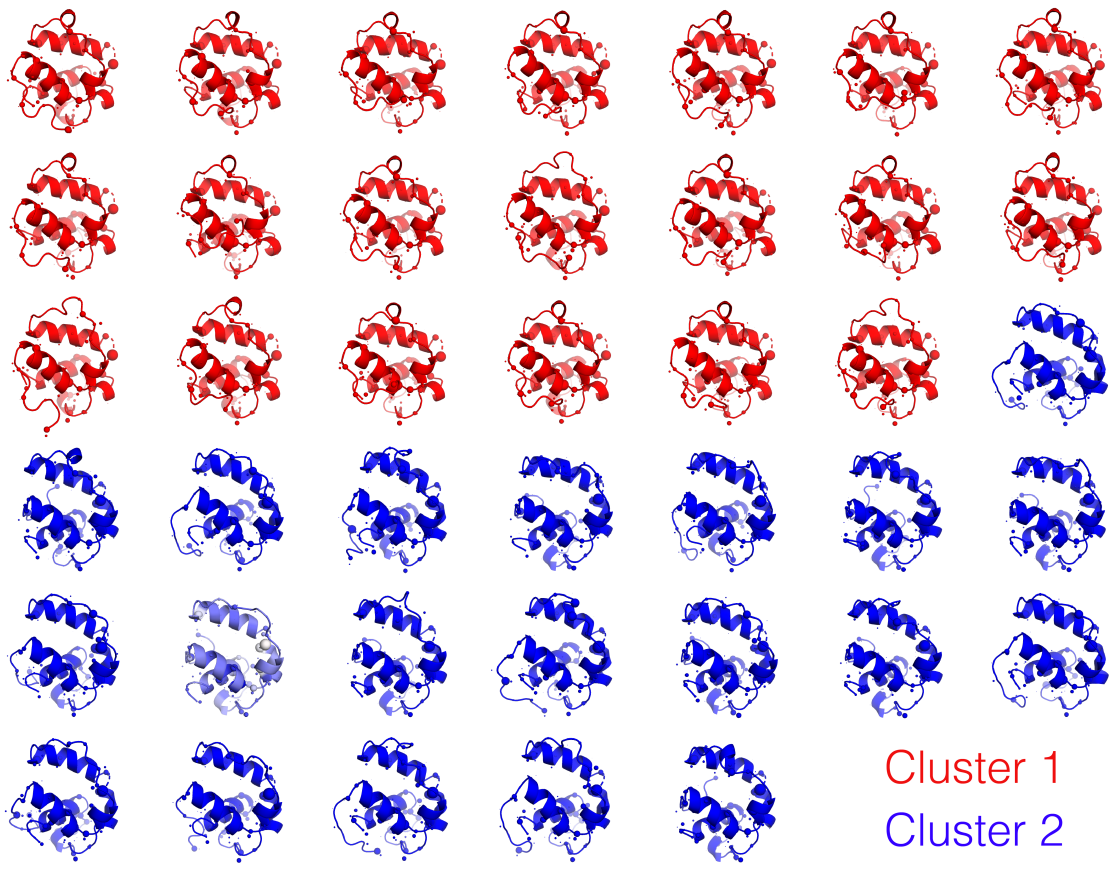

Figure B.**7:** Results obtained by clustering the structures of the free-state (*red*; PDB ID: 2MY6) and bound-state (*blue*; PDB ID: 2MY5) structures of the protein, KstB-PCP, using their *computed* chemical shifts as features.

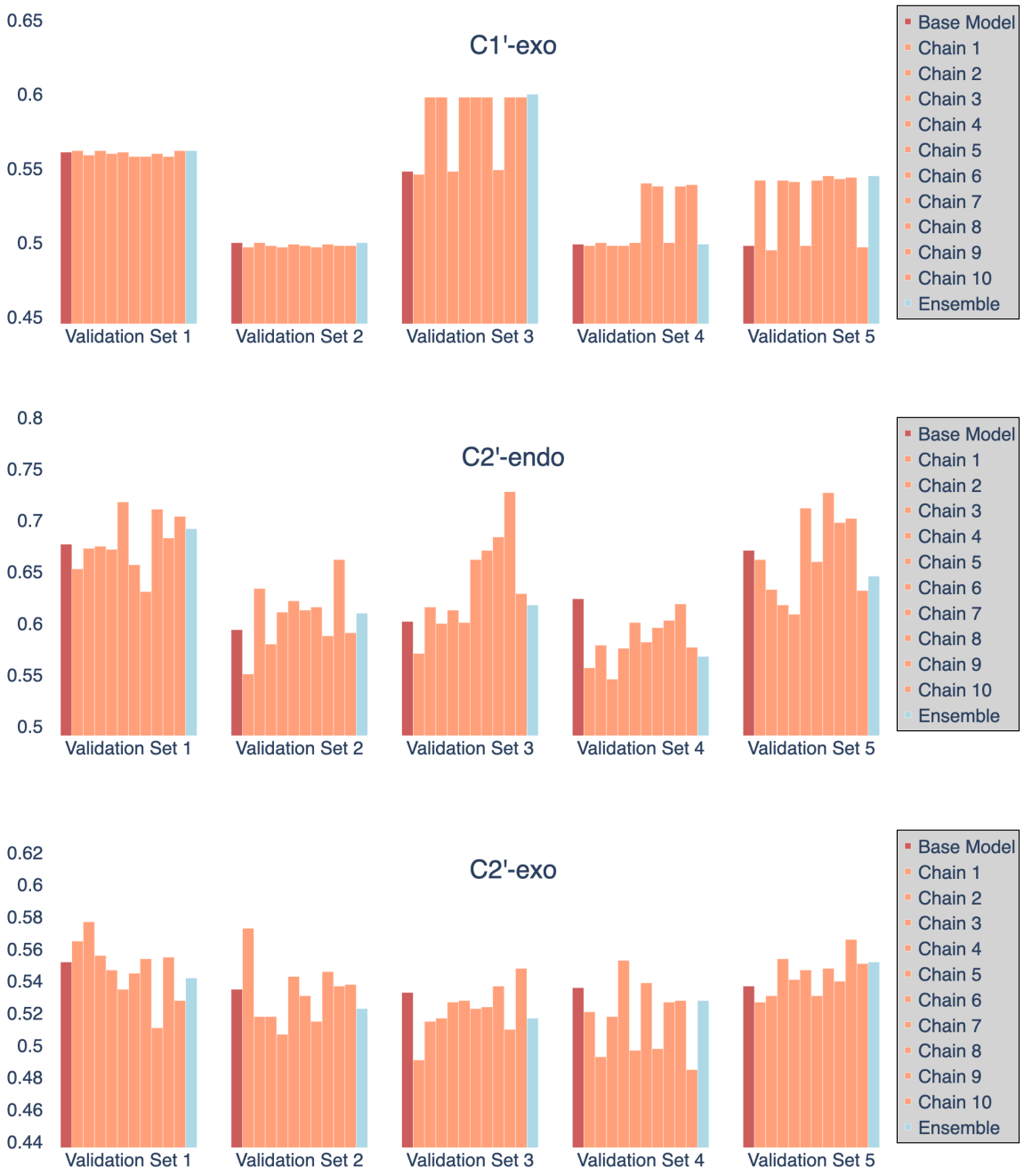

Figure B.**8:** Performance comparison between vanilla MLP model and chained model (Part 1). Shown in the figures are the balanced accuracy scores via a 5-fold cross validation assessment. The first bar in each block (*dark red*) represents the independent MLP model; the next 10 bars (*salmon*) represent chained models with random orderings; the last bar (*light blue*) represents the ensemble model that is averaged from 10 chained models.

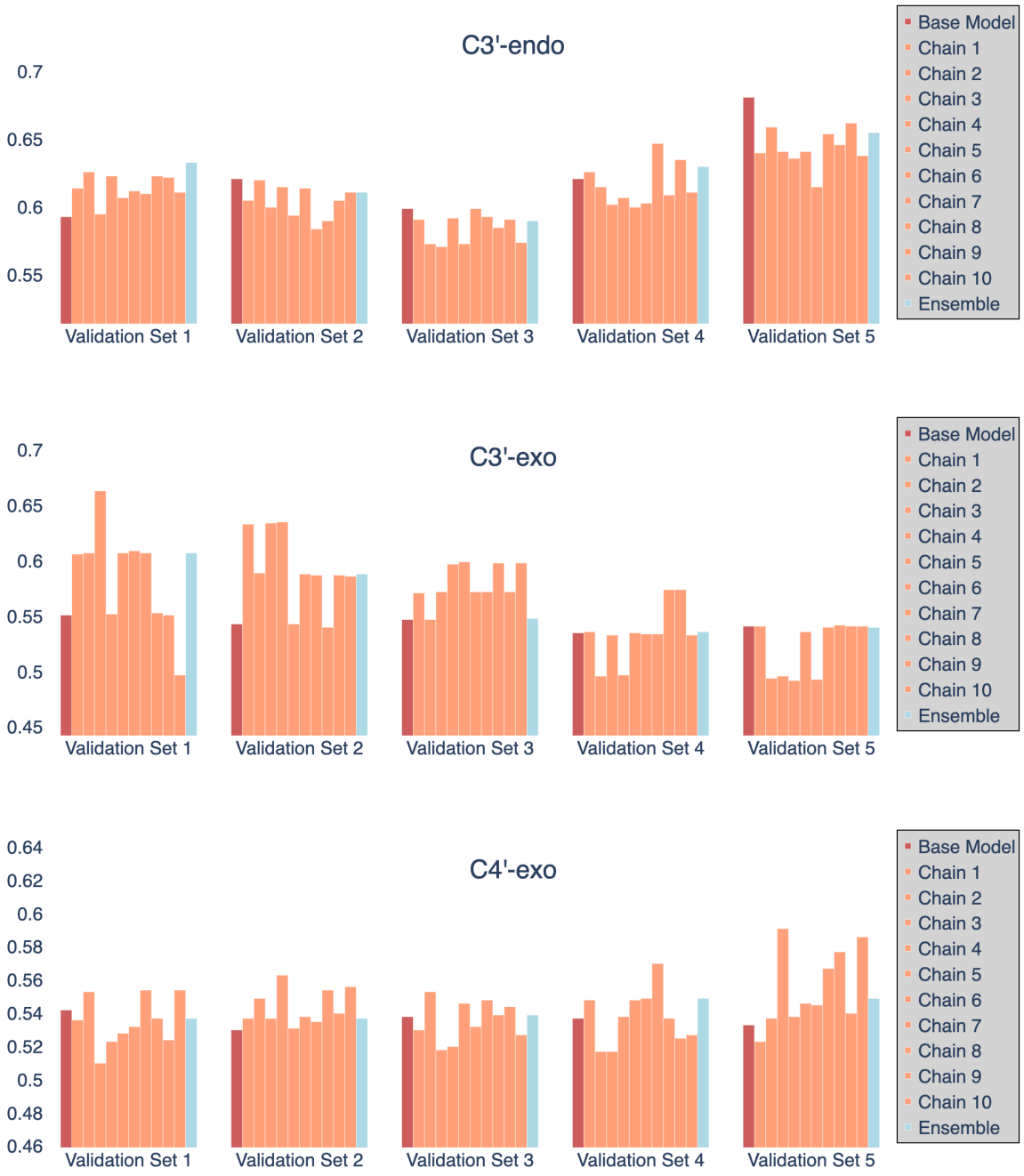

Figure B.**9:** Performance comparison between vanilla MLP model and chained model (Part 2). Shown in the figures are the balanced accuracy scores via a 5-fold cross validation assessment. The first bar in each block (*dark red*) represents the independent MLP model; the next 10 bars (*salmon*) represent chained models with random orderings; the last bar (*light blue*) represents the ensemble model that is averaged from 10 chained models.

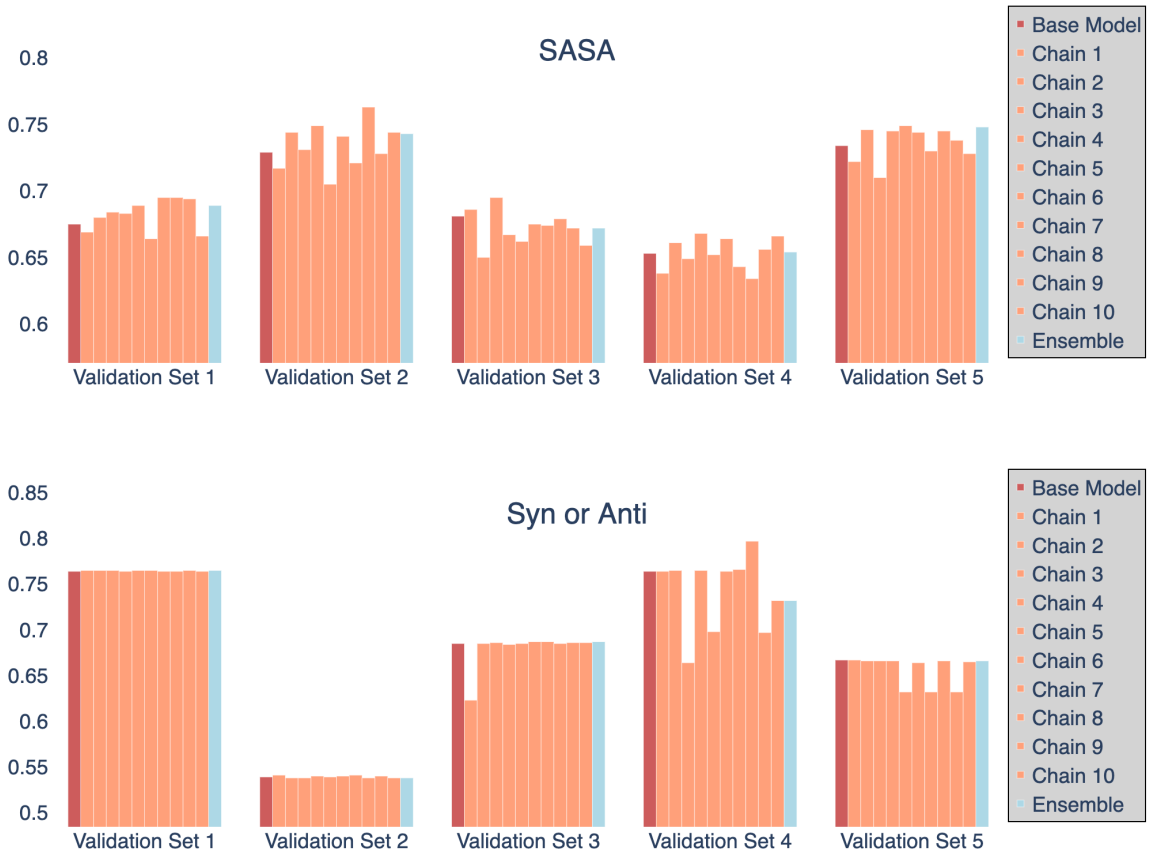

Figure B.**10:** Performance comparison between vanilla MLP model and chained model (Part 3). Shown in the figures are the balanced accuracy scores via a 5-fold cross validation assessment. The first bar in each block (*dark red*) represents the independent MLP model; the next 10 bars (*salmon*) represent chained models with random orderings; the last bar (*light blue*) represents the ensemble model that is averaged from 10 chained models.