INTERACTING WITH RNA SECONDARY STRUCTURE THROUGH SONIFICATION

Andreas Papagiannis, Danyi Liu, Alan Zammit, Alexander P. Gulyaev and Fons J. Verbeek
Leiden Institute of Advanced Computer Science, The Netherlands

ABSTRACT

In this study, sonification strategies are explored by enabling the creation of audio files that represent the RNA secondary structure. A specific rule-set is employed which ensures that audio files generated for different RNA secondary structures will also yield distinct melodies, based on the underlying sound structural rules. The final goal is to create audio files that are constructed in such a way that a user familiar with RNA secondary structure will be able to easily detect audio nuances corresponding to the different RNA structures, as well as understanding which melodies belong to clusters of related RNA secondary structure.

KEYWORDS

RNA Secondary Structure, MIDI, RNA Pseudoknots, Sonification

1. INTRODUCTION

In this work, a novel approach based on standard programming tools and sonification techniques has been used to create non-speech audio for conveying information concerning RNA secondary structure. In addition to listening the outputted audio in a preconfigured way, a user can also use control a purpose-built device to interact with the audio in order to explore and perceptualize the sonified RNA structure.

1.1 Sonification

Sonification is the representation of data in an auditory way. In other words, one can perceive sonification as the equivalent of visualization stimulating our ears instead of our eyes. Sonification has already been used as a tool to help understand obscure datasets, e.g. electrocardiography signals (Kather et al. 2017), or help visually impaired individuals to interpret data (Pereira et al. 2013). In the past, sonification has already been used as a tool to comprehend biological phenomena in a number of projects (Temple, 2017; Shi et al., 2007). It is worth mentioning that sonification can work well in conjunction with traditional visualization techniques, or even a virtual reality environment (Rau et al. 2015), in order to further enhance comprehension by the user (Du et al. 2018). A simplistic example of audio mappings for sonification purposes as described in (Temple 2017) (named Mononucleotides algorithm) is mapping each of the four DNA bases (A, G, C, T) to a different notes (pitch frequencies). So as one can imagine, sonification is about mapping features obtained from data to sound characteristics in a meaningful way.

1.2 RNA Secondary Structure

Two-dimensional (RNA) structure conveys information about the physical conformation that may be attained by biological molecules. This largely depends on whether or not the nucleotide bases present in an RNA biopolymer, and which are located sequentially along the molecule's backbone, are able to interact (pair) with others along the same backbone, creating contiguous double- or single-stranded regions. Classical (*Watson-Crick*) base-pairing, and the set of complex, biochemical rules that drive cellular organisms, act in concert to govern the most stable conformation an RNA molecule can achieve, all derivable from an arbitrarily large set of shapes. In practice, however, natural biological processes tend to favor specific

conformations, and -even more so the case- specific modules (*motifs*) that not only denote highly-common 2D structure but also biological function (Hendrix, Brenner and Holbrook 2005). Thus it is common for RNA stems or helices of (paired nucleic acid bases), hairpin loops (a stem with an intervening loop), interior loops, pseudoknots as well as a number of other well-documented structural features. It should be noted that some of these features, such as pseudoknots, fall under tertiary structure, and are typically resolved using 3D probing techniques. *This is particularly relevant to this study, as a complete description of the complex 2D/3D arrangement of RNA molecules motivates for accurate, but more intuitive representations of such structures.* Three of the most common RNA structural models are dumbbell, stem, and cloverleaf structures. In order to represent these structures in an easy way for input to a computer representation, the consensus secondary structure descriptor (CSSD¹) format is used (Hendrix, Brenner and Holbrook 2005). CSSD is a textual notation for describing RNA secondary structure. This is accomplished by representing base pairs with '< >' and '()'. The latter are used for helices closing a multi-branched structure, whereas angled brackets represent base pairs of a terminal stem structure. Unpaired nucleotides in multi-branch loops are represented by the comma symbol ','. Underscores '_' represent hairpin loop residues, dashes '-' for bulge or interior loops and lastly square brackets '[']' represent pseudoknot positions in an RNA sequence. Finally, semicolons ';' are used to denote unstructured nucleotides.

Figure 1. RNA structure example with respective CSSD format

1.3 MIDI Protocol

Musical Instrument Digital Interface (MIDI) is an information protocol used by the majority of synthesizers, audio engines, audio samplers and digital audio workstations. In short, the MIDI protocol can dictate audio generators what notes to play. However, there is a plethora of parameters that can be set by MIDI communication. For our case study, we will use MIDI files generated in Python to drive any sound engine that we will use.

1.4 Aim

The aim of this project is to create audio stimuli that enable listeners to recognize and identify distinct RNA structures. Audio stimuli can serve both as an enhancement for existing visualization of RNA secondary structures as well as for recognition or identification of structure characteristics only with sounds. The sound sources themselves are not as important as in this study, we focus more on the melodic structure of the respective audio stimuli as well as its sound direction.

1 CSSD is an extension of the WUSS notation that supports classic 2D structure in addition to biologically meaningful, tertiary interactions such as pseudoknots and base triples

2. EXPERIMENTS & METHOD

2.1 Experimental Pipeline

In order to achieve the aforementioned aim, the following sequential steps are taken. First, an RNA structure is recruited into the framework. This is done by either creating a random structure or by inputting a known structure in CSSD format. After having the secondary structure in the desired format entered, features of the structure are extracted. Subsequently, depending on the features and the audio mappings in use, a MIDI file specifying audio parameters dictating how the final melody will sound like is created. Lastly, MIDI information must be transferred to an audio generator. This can be done in two different ways. Either the MIDI files created are manually imported to any audio engine in use, or all MIDI information saved in the file are streamed in real time using a virtual MIDI port.

2.2 Audio Mappings in use

After extrapolating and selecting a set of RNA secondary structure features to describe, they are mapped to various MIDI messages that will be saved MIDI files. These MIDI messages occur according to the features extracted previously, such that the final sonified audio files will have a meaning in terms of biology and will not sound totally random. For every MIDI file we are able to change a number of parameters such as pitch or MIDI control messages (MIDI CC) that can be mapped as an audio mapping to any parameter of the target sound engine. In this section, we explain how all audio mappings were made. These can be divided in two categories: mappings altering the frequency of the sound spectrum, i.e. low to high frequencies, and mapping altering the spatial characteristics of any audio.

Pitch: Pitch may be one of the most distinguishable of all audio features when someone listens to a melody. In other words, humans are able to classify low from high pitches. Thereafter, the audio mapping chosen to it must also represent a truly important feature of each RNA secondary structure. In our case pitch frequency is mapped to the kind of connection every nucleotide has in the secondary structure of the RNA sequence. Also, the pitch is depending on the previous pitch of the sequence. More specifically, every time a new base pair appears in the sequence (shown as '(' or '<' in CSSD format), the pitch frequency will ascend a step. That means the if for example the previous note played was C4 (note C of the 4th octave), if the next nucleotide is the first of a base pair of the structure then the note that will represent the current nucleotide will be C#4. If the current nucleotide represents a closing base pair (shown as ')' or '>' in CSSD format), then the pitch will descend a step. In case the nucleotide is an internal loop (shown as '-' in CSSD format), then the pitch will stay the same as the pitch previously played. In case of a hairpin loop ('(' in CSSD format) there will be no sound playing. The pitch will supposedly stay same, but since there will be no 'note-on' MIDI message included, it is conceived as a pause. Note that this is the only case where a 'note-off' MIDI message instead of a 'note on' is passed to both the MIDI file as well as the MIDI port. If the corresponding nucleotide represents an unpaired nucleotide in a multi-branch loop or an unstructured one (shown as ',' and ':' respectively in CSSD format), then as it is important identifying these parts of the sequence, the pitch changes to a random note from the selected scale two octaves above the previously played note. This sudden octave ascension results to a change the listener always notices.

Pseudoknots: So far, each nucleotide is assigned to a specific pitch but this is not the case whenever a pseudoknot appears in the sequence. In this case, an arpeggio in the selected scale starts playing. The length, in notes, of the arpeggio is equal to the number of nucleotides tied into the pseudoknot. The arpeggio is split in two halves in the same fashion the nucleotides forming a pseudoknot as split in two groups. The first half of the arpeggio plays the notes in an ascending way, meaning that the first note to be played is of lower pitch than the second pitch, and starts exactly where the first set of nucleotides form the pseudoknot. In the exact same way, when encountering the second set of nucleotides closing the pseudoknot, the same arpeggio plays as before but this time the notes are played in a descending order to indicate that the pseudoknot is now closing.

Table 1. Sonification Pitch strategy

CSSD symbol	Nucleotide action	Pitch
(	helix opening	up
)	helix closing	down
<	base pair opening	up
>	base pair closing	down
[	pseudoknot opening	arpeggio up
]	pseudoknot closing	arpeggio down
—	hairpin loop	pause
:	unstructured	pause
-	interior loop	chord & repeat
,	unpaired	dissonance/random

Reverb: Reverb is an audio effect that add depth to a sound. More specifically it is used to simulate spaces. Imagine shouting inside a huge cathedral, the sound waves of your voice will be reflected to all walls of the cathedral and we hear a mixture of direct sound (sound that travels straight to your ears) and reflected sound (sound that radiates, bounces off of walls and things, and travels back to your ears). In our case, the distance each nucleotide has from the center of the RNA's secondary structure will affect the level of reverb applied to the sound. Nucleotides adjacent to the structure's center will have no reverb on their sound whereas nucleotides far away from the center will have more reverb applied to their sound receptively.

Sound direction: The index of the sequence is always mapped to the direction of the sound no matter how many speakers are included in the setup. In case of two speakers (stereo setup) the audio starts from the left side speaker and following a sigmoid function moves over to the right speaker. When using a four speaker setup (or any setup having placing multiple speakers in a 3D plane), the sound will rotate from the first to the last speaker. These two last mappings are used to help users conceive better the selected RNA structure in a spatial way.

2.3 Framework Tools

2.3.1 Python

Random RNA structures following the Consensus Secondary Structure Descriptor (CSSD) notation as a format are created in Python (<https://www.python.org/>). This is done in order to test as many RNA secondary structures possible and look in detail every case of RNA structure. Of course, any RNA structure following the same format can be inputted instead of a random one as long as they follow the same notation.

Created random RNA structure had one, two or three stem like branches. In case of having two or three branches, a pseudoknot (holding from four to eight nucleotides) may appear between the first to hairpin loops with probability 0.5. The length of the random structure created was between 100 and 200 (chosen at random) nucleotides. Single-stranded regions of 10 nucleotides were also added at the beginning as well as the end of the randomly created structure. The number of nucleotides forming a closing helix is also randomized to values close to 1/10 of the total length of the structure. More unpaired nucleotides (than the absolutely essential ones) are added to the sequence with probability of 0.14 and number close to 1/22 of the total length of the structure. Lastly the way the random structure is created takes into account that base pairings as well as interior loops tend to group with each other in a meaning that if a nucleotide is forming a base pair, it is highly probable that it's neighboring nucleotides will also form base pairings.

Python 3.7 again is used in order to parse, analyze and extract features out of RNA each sequence concerning its secondary structure. These features consist of nucleic acid pairings or unpairings, hairpin loops or even 'meta-features' such as pseudoknots.

The Mido (short for MIDI Objects) Python library is used in order to create midi files in Python (<https://mido.readthedocs.io/en/latest/>). There is no restriction concerning the number of MIDI files generated. That means that a single RNA sequence can correspond to multiple time synchronized MIDI files. Every MIDI message contains information about the note frequencies as well as general MIDI information corresponding to a selection of the features extracted. All MIDI files are then transferred to the selected sound engine. This can be done in two ways. Either the associated MIDI files will be inputted manually to any MIDI player, or all information will flow through any of the selected available MIDI ports in the computer running the Python.

2.3.2 Pure Data

After having creating all MIDI messages needed in Python, a Pure Data patch (<https://puredata.info/>) is made in order to work as an audio engine and MIDI receiver. In this patch, all MIDI CC created before are mapped to audio parameters of the audio engine in use. The Pure Data patch is set up in way that it will store and record all MIDI messages sent to the selected MIDI port. After storing all MIDI information, the whole note sequence can be played once, looped so that it continuously keeps playing and exported as a .wav audio file.

2.3.3 Controller Device

A controller device is built in order to help the user explore the sound file created after its respective MIDI messages has been imported to Pure Data. The user is able to control a number of parameters concerning the playback of the sound.

Firstly, three potentiometers help the user control the overall volume and speed of the sequence to be played as well as the specific index of the sequence. Here speed affects the time period between notes of the sequence. If for example the entered sequence is large, playing it in the predetermined speed may take a lot of time which is not convenient for any user as it is counter intuitive listening to a sound for more than a minute. So, in this case the user can change the speed so the in between notes time will be shorter resulting in a shorter overall sound.

Figure 2. Sketch design of the controller device connectable to computer via USB. There are three potentiometers controlling speed, index and volume of sequence. The four buttons below control sequence playback

Figure 2 shows a layout of how the controller device could look like. At the bottom of the device (as seen in 5) four buttons are located. The 'Play' button will start the playback of the sound from the starting point of the sound. As expected, 'Stop' will simply pause any sound currently playing but also set the starting point of the sound at its beginning. The 'Previous' and 'Next' buttons will play the next and previous notes respectively if a sound is already playing. In this way a user can set the speed at its lowest position and explore the whole sound back and forth by using these two buttons. When no sound is playing, these buttons will set the starting point of a sound a note before or a note after from the previous starting point.

The device is powered and programmed in conjunction with an Arduino micro (<https://www.arduino.cc/>) and is connected to any computer via micro USB. More specifically, it acts as a USB MIDI controller meaning that it can transfer MIDI messages to any MIDI receiver via USB. In our case the MIDI receiver unit is any computer running Pure Data. The MIDI messages transferred are MIDI control changes that are then mapped to any specific values one wishes to control (such as playback speed or volume) inside Pure Data.

Figure 3. Depiction of generated audio of RNA 2D structure. Y-axis, represents pitch from low to high (keys C1 to C5) while X-axis represents the time domain from start to end of the MIDI file generated. Below the CSSD format of the respective RNA secondary structure is shown

3. CONCLUSION & FUTURE WORK

Sonification as a tool for conceptualizing biological data should be considered as a deep dive exploration in alternative ways of human information perception. In conjunction with visual stimulation, sonification of biological data can possibly help users understand difficult to perceive phenomena in depth. The research project is realized having in mind user experience so that the user is able to interact through the controller device and explore each RNA 2D structure in a comprehensive way. More specifically, in this paper we propose a model based sonification. In other words, we construct a fixed model to sonify structural data. The pitch and sound direction strategies used were built and chosen considering what works best for us in order to classify RNA structural differences by listening to their respective audio generated by MIDI files. However, in order to evaluate this method, a testing procedure should be followed where participants will spend time interacting with the controller built while listening to the audio sequence of a respective RNA structure. After (or even while) auditioning multiple MIDI files, each participant should fill a questionnaire targeted to their understanding of the structural properties of the respective RNAs. Analyzing the answers of such a questionnaire, we will then be able to evaluate if this whole method is suitable for all.

Future work may include embedding this whole experimental pipeline in a micro-controller device such as Bela (<https://bela.io/>) as well as experimenting with different controller devices such as motion detectors (<https://www.leapmotion.com/>) in order to further enhance user experience. Moreover, this whole procedure could be replicated sonifying 3D instead of 2D RNA structures and mapping nucleotide position in 3D space to sound direction using eight speakers positioned in a cubic way. Last, relevant to the RNAs structure, visual stimuli can be displayed in order to help the user perceptualize even better the respective structure.

REFERENCES

- Temple M. D., 2017, An auditory display tool for DNA sequence analysis. *BMC Bioinformatics*, 18. 10.1186/s12859-017-1632-x.
- Shi X. J., Cai Y. Y. And Chan C. W., 2007, Electronic Music for Bio-Molecules Using Short Music Phrases. *Leonardo*, Vol. 40, No. 2 (2007), pp. 137-141.
- Pereira F. et al, 2013, Sonified Infrared Spectra and Their Interpretation by Blind and Visually Impaired Students. *J. Chem. Educ.*, 2013, Vol. 90, No. 8, pp 1028–1031.
- Rau B. et al, 2015, Enhancing visualization of molecular simulations using sonification. *2015 IEEE 1st International Workshop on Virtual and Augmented Reality for Molecular Science*. Arles, France.
- Du M. et al, 2018, Exploring the Role of Sound in Augmenting Visualization to Enhance User Engagement. *2018 IEEE Pacific Visualization Symposium*. Kobe, Japan.
- Kather J. N. et al., 2017, Polyphonic sonification of electrocardiography signals for diagnosis of cardiac pathologies. *Scientific Reports*, Vol. 7.
- Lawrence C. and Ding Y, 2004, A Statistical Sampling Algorithm for RNA Secondary Structure Prediction. *Nucleic Acids Research*, Vol. 31, No. 24, pp. 7280 – 7310.
- Hendrix DK, Brenner SE, Holbrook SR, 2005, RNA structural motifs: building blocks of a modular biomolecule. *Quarterly reviews of biophysics*, Vol. 38, No. 3, pp. 221-243.