Title	Prediction of RNA Joint Secondary Structures Based on Integer Programming
Author(s)	Kato, Yuki; Sato, Kengo
Citation	Methods in Molecular Biology. 2026, 2949, p. 99-107
Version Type	AO
URL	https://hdl.handle.net/11094/103361
rights	
Note	

Prediction of RNA joint secondary structures based on integer programming

Yuki Kato and Kengo Sato

Abstract Predicting a joint secondary structure formed by two RNAs plays a key role in understanding mechanisms and functions of complex RNA–RNA interactions. Here we describe a practical proceedure of a fast and accurate computational method for predicting joint structures based on integer programming.

Key words RNA–RNA interaction, RNA joint secondary structure, Regulatory RNA, Integer programming, Maximum expected accuracy

1 Introduction

A non-coding RNA can interact with its target RNA by base pairing that mainly consists of Watson–Crick $\{$A, U$\}$ and $\{$G, C$\}$ pairs along with wobble $\{$G, U$\}$ pairs. This string-based rule has motivated computational scientists to develop a variety of algorithms for predicting RNA–RNA interactions. They are classified into two approaches: one that neglects intramolecular base pairs and the other that considers internal structures of two RNAs [1]. Most of the latter approach aim to predict a joint secondary structure, which tends to reflect more realistic interactions than the former approach that mainly computes thermodynamic scores of duplexes between two RNAs. In general, predicting a joint structure with minimum free energy is NP-hard if there is no constraint on the topology [2]. Thus, adding valid constraints on the topology of predictable joint structures is reasonable, and existing methods impose

Yuki Kato:
Department of RNA Biology and Neuroscience, Graduate School of Medicine, and Integrated Frontier Research for Medical Science Division (iFremed), Institute for Open and Transdisciplinary Research Initiatives (OTRI), Osaka University, 2-2 Yamadaoka, Suita, Osaka 565-0871, Japan, e-mail: ykato@rna.med.osaka-u.ac.jp;
Kengo Sato:
School of System Design and Technology, Tokyo Denki University, 5 Senju Asahi-cho, Adachi-ku, Tokyo 120-8551, Japan.

such constraints. For example, intramolecular crossing base pairs (pseudoknot) and external crossing interactions are disallowed. An example of complicated but restricted class of RNA–RNA interactions is a kissing hairpin, where loops of hairpin substructures from two RNAs are interacted with each other (Fig. 1). Some computational methods can predict kissing hairpins [2, 3, 4, 5, 6], but their computation time is $O(n^6)$, where n is the sequence length.

To overcome such high computational complexity, RactIP (RNA–RNA interaction prediction using integer programming) was proposed on the basis of an integer programming (IP) formulation [7]. In an IP problem, we would like to minimize/maximize a linear objective function subject to linear constraints described by linear equalities and/or inequalities with respect to variables of non-negative integers. Notably, modeling with IP is flexible enough to solve a wide range of combinatorial problems, and recent optimization solvers can solve IP problems faster than previously thought. In the RactIP formulation, an approximate posterior probability of a joint secondary structure is incorporated into the objective function of its IP formulation. Maximizing this objective function corresponds to maximizing expected accuracy of a predicted joint secondary structure. Such approximation achieves considerable speed-up in computation time while keeping prediction accuracy [7]. The topology of predictable joint structures can be modeled by a set of constraints. Furthermore, motivated by some accessibility-based approaches that compute a free energy to make a sequence stretch open for RNA–RNA binding [8, 9], RactIPAce (RactIP with precomputed accessibility) was developed to incorporate interaction site accessibility to improve prediction accuracy [10].

In what follows, we will describe a practical procedure to install and run RactIP (RactIPAce) for predicting joint structures based on IP. Note that RactIP includes RactIPAce because the function of RactIPAce can be invoked as a command line option when using the latest version of RactIP.

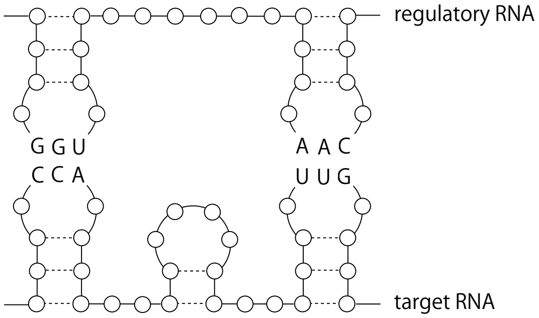

Fig. 1 An example of two kissing hairpin structures formed by a regulatory RNA and its target. A white circle indicates a base, and a dashed line shows an internal base pair. To emphasize external base pairs underling the RNA–RNA interaction, actual bases in the hairpin loop regions are shown to illustrate Watson–Crick base pairings.

2 Materials

2.1 Software Requirements

RactIP can run by command line interface on Linux, Windows Subsystem for Linux (WSL), and macOS. Before installing RactIP onto user's computer, a few tools described below need to be installed. Although we illustrate detailed procedures on Linux (Ubuntu 22.04) as an example, we believe that these can be performed similarly in other computational environments. In the following, assume that we are in `/home/username` directory.

2.1.1 ViennaRNA Package

The ViennaRNA package [11] version 2.2.0 or later, a toolkit for analyzing RNA secondary structures, is required by RactIP. The ViennaRNA package can be installed in your terminal by the following steps:

Command

```
wget https://www.tbi.univie.ac.at/RNA/download/\
sourcecode/2_5_x/ViennaRNA-2.5.1.tar.gz
tar zxf ViennaRNA-2.5.1.tar.gz
cd ViennaRNA-2.5.1
./configure
make
sudo make install
cd ..
```

The above commands will install the ViennaRNA package to `/usr/local` (see Note 1).

2.1.2 Optimization Solvers

Any of the optimization solvers GNU Linear Programming Kit (GLPK) (≥ 4.41) [12], Gurobi Optimizer (≥ 8.0) [13], IBM ILOG CPLEX (≥ 12.0) [14], SCIP (≥ 8.0.3) [15], and HiGHS (≥ 1.5.0) [16] is necessary to install to solve IP problems designed in the RactIP framework. For example, the free package GLPK 5.0 can be installed by the commands shown below (see Note 2):

Command

```
wget http://ftp.gnu.org/gnu/glpk/glpk-5.0.tar.gz
tar zxf glpk-5.0.tar.gz
cd glpk-5.0
./configure
make
sudo make install
cd ..
```

Refer to an instruction of another solver to install it (see Note 3).

2.2 Installing RactIP

2.2.1 Building an Executable Program from Source Codes

We demonstrate how RactIP 1.1.1 can be built with GLPK. Make sure that a C++17 compatible compiler, and the commands `cmake` and `pkg-config` work on your environment, otherwise install them. The GitHub website at `https://github.com/satoken/ractip` is also helpful to install RactIP.

Command

```
wget https://github.com/satoken/ractip/archive/\
refs/tags/v1.1.1.zip
unzip v1.1.1.zip
cd ractip-1.1.1
export PKG_CONFIG_PATH=/usr/local/lib/pkgconfig:\
$PKG_CONFIG_PATH
cmake -DCMAKE_BUILD_TYPE=Release -S . -B build
cmake --build build
sudo cmake --install build
```

For Gurobi, add `-DENABLE_GUROBI` to the above configure step:

Command

```
cmake -DENABLE_GUROBI=true \
-DCMAKE_BUILD_TYPE=Release -S . -B build
```

For CPLEX, add `-DENABLE_CPLEX` to the configure step (see Note 4):

Command

```
cmake -DENABLE_CPLEX=true \
-DCMAKE_BUILD_TYPE=Release -S . -B build
```

For SCIP, add `-DENABLE_SCIP` to the configure step (see Note 5):

Command

```
cmake -DENABLE_SCIP=true \
-DCMAKE_BUILD_TYPE=Release -S . -B build
```

For HiGHS, add `-DENABLE_HIGHS` to the configure step:

Command

```
cmake -DENABLE_HIGHS=true \
-DCMAKE_BUILD_TYPE=Release -S . -B build
```

2.2.2 Using a Precompiled Binary

Users can also install RactIP precompiled binary if they use Linux or WSL. Assume that we are in `/home/username` directory.

Command

```
wget https://github.com/satoken/ractip/releases/\
download/v1.1.1/ractip-1.1.1-x86_64-linux.zip
unzip ractip-1.1.1-x86_64-linux.zip
```

The executable file `ractip` can be found in `ractip-1.1.1-x86_64-linux` directory.

3 Methods

3.1 Data Preparation

Example input sequences in FASTA format for performing joint secondary structure prediction are available in `ractip-1.1.1/data` directory or `ractip-1.1.1-x86_64-linux/data`, which have already been downloaded through the installation of RactIP described above.

3.2 Predicting Joint Secondary Structures

The current directory is assumed to be `/home/username/ractip-1.1.1`. Taking sequences `CopA.fa` and `CopT.fa` in FASTA format (see Note 6), which are known to form a kissing hairpin structure [17], RactIP can run by the following command:

Command

```
ractip data/CopA.fa data/CopT.fa
```

A predicted joint secondary structure will soon be returned in dot-bracket representation in standard out as shown in Fig. 2. To evaluate the prediction quantitatively, RactIP has the option `-e` to calculate the free energy of the predicted joint structure. For example, free energy -37.88 kcal/mol of the joint structure formed by CopA–CopT will be shown at the bottom of the prediction.

3.3 Running with Docker

If Docker has been installed, the following commands can run RactIP:

Command

```
docker build . -t ractip
docker run -it --rm -v $(pwd):$(pwd) -w $(pwd) \
ractip ractip CopA.fa CopT.fa
```

3.4 Using a Web Server

To avoid installation for some reason, or see an arc representation of a predicted joint structure explained below, we provide a web server Rtips [18] online at `http://ws.sato-lab.org/rtips/ractip/`. In the web server, CopA and CopT sequences will appear as Example 3 in the top page, and just clicking "Predict" button will perform joint structure prediction as shown in the command line case. Unlike the standalone program, the web server will output not only a dot-bracketed joint structure with a predicted free energy, but also an arc representation of the prediction shown in Fig. 3 by virtue of a drawing tool VARNA [19].

4 Notes

1. A user without root privileges on his/her computer cannot edit `/usr/local`, which means that the final command `sudo make install` will fail. To resolve this issue, create a directory under the home directory such as `/home/username/ViennaRNA`, and run the configure script as follows:

```
>CopA
CGGUUUAAGUGGGCCCCGGUAAUCUUUUCGUACUCGCCAAAGUUGAAGAAGAUUAUCGGGGUUUUUGCUU
.......(((.((((((((((((((((.[[[[[[[[[[[[[[[....))))))))))))))))...))).
>CopT
AAGCAAAAACCCCGAUAAUCUUCUUCAACUUUGGCGAGUACGAAAAGAUUACCGGGGCCCACUUAAACCG
.........(((((.(((((((.....]]]]]]]]]]]]]]]..))))))).))))).............
```

Fig. 2 A predicted kissing hairpin structure formed by antisense RNA CopA and its target RNA CopT. Parentheses '( )' denote an internal base pair, whereas brackets '[ ]' show an external base pair.

```
./configure --prefix=/home/username/ViennaRNA
```

2. A non-root user should install GLPK by the command like this:

```
./configure --prefix=/home/username/glpk
```

   where the directory `glpk` has been made under the home directory.
3. Gurobi Optimizer and IBM ILOG CPLEX are commercial software but grant free academic license.
4. If the optimization solver has been installed to the directory other than `/usr` or `/usr/local`, add an option that indicates the root directory of the solver installed to the configuration step. Here is an example of CPLEX-incorporated installation:

```
cmake -DCPLEX_ROOT_DIR=/path/to/CPLEX \
-DENABLE_CPLEX=true -DCMAKE_BUILD_TYPE=Release \
-S . -B build
```

   As for the case of Gurobi, add `-DGROBI_ROOT_DIR=/path/to/Gurobi` to the configuration.
5. If SCIP has been installed to the directory other than `/usr` or `/usr/local`, specify the option `-DCMAKE_MODULE_PATH=/path/to/scip/lib/cmake` in the configuration step.
6. The FASTA formatted file begins with '>' including its sequence ID, followed by an actual sequence from the second row.

Acknowledgements This work was supported by the Japan Society for the Promotion of Science KAKENHI [24700296 and 15K00401 to YK, 16K00404 to KS].

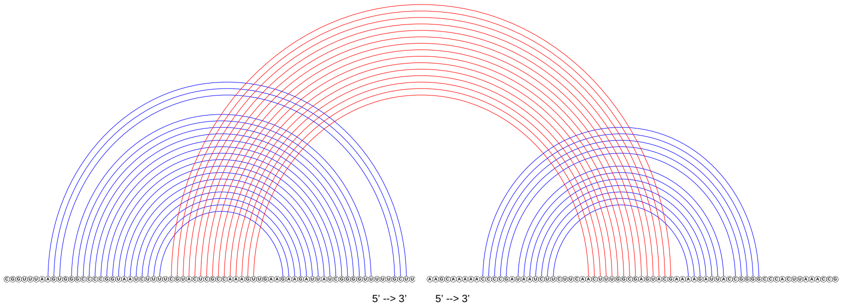

Fig. 3 An arc representation of the predicted joint structure formed by CopA and CopT. Note that both sequences are oriented in the 5′ to 3′ direction from left to right. A blue arc indicates an intramolecular base pair, and a red arc shows an interaction between two RNAs.

References

1. Backofen R (2014) Computational prediction of RNA–RNA interactions. Methods Mol Biol 1097:417–435
2. Alkan C, Karakoç E, Nadeau JH, et al (2006) RNA–RNA interaction prediction and antisense RNA target search. J Comput Biol 13:267–282
3. Pervouchine DD (2004) IRIS intermolecular RNA interaction search. Genome Inform 15:92–101
4. Chitsaz H, Salari R, Sahinalp SC, othres (2009) A partition function algorithm for interacting nucleic acid strands. Bioinformatics 25:i365–i73
5. Huang FWD, Qin J, Reidys CM, othres (2009) Partition function and base pairing probabilities for RNA–RNA interaction prediction. Bioinformatics 25:2646–2654
6. Kato Y, Akutsu T, Seki H (2009) A grammatical approach to RNA–RNA interaction prediction. Pattern Recognit 42:531–538
7. Kato Y, Sato K, Hamada M, et al (2010) RactIP: fast and accurate prediction of RNA–RNA interaction using integer programming. Bioinformatics 26:i460–i466
8. Mückstein U, Tafer H, Hackermüller J, et al (2006) Thermodynamics of RNA–RNA binding. Bioinformatics 22:1177–1182
9. Busch A, Richter AS, Backofen R (2008) IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics 24:2849–2856
10. Kato Y, Mori T, Sato K, et al (2017) An accessibility-incorporated method for accurate prediction of RNA–RNA interactions from sequence data. Bioinformatics 33:202–209
11. Lorenz R, Bernhart SH, Höner Zu Siederdissen C, et al (2011) ViennaRNA package 2.0. Algorithms Mol Biol 6:26
12. GLPK (GNU Linear Programming Kit). `https://www.gnu.org/software/glpk/`
13. Gurobi Optimization. `https://www.gurobi.com/`
14. IBM ILOG CPLEX Optimization Studio. `https://www.ibm.com/products/ilog-cplex-optimization-studio`
15. SCIP (Solving Constraint Integer Programs). `https://scipopt.org/`
16. HiGHS (high performance software for linear optimization). `https://highs.dev/`
17. Kolb FA, Malmgren C, Westhof E, et al (2000) An unusual structure formed by antisense-target RNA binding involves an extended kissing complex with a four-way junction and a side-by-side helical alignment. RNA 6:311–324
18. Kato Y, Sato K, Asai K, et al (2012) Rtips: fast and accurate tools for RNA 2D structure prediction using integer programming. Nucleic Acids Res 40:W29–W34
19. Darty K, Denise A, Ponty Y (2009) VARNA: interactive drawing and editing of the RNA secondary structure. Bioinformatics 25:1974–1975