

Manual for DotKnot 1.3.2

A tool for RNA pseudoknot prediction

Jana Sperschneider

September 17, 2015

1 What is DotKnot?

DotKnot is a heuristic method for pseudoknot prediction in a given RNA sequence. DotKnot extracts stems from the secondary structure probability dot plot calculated by RNAfold¹. H-type pseudoknots and kissing hairpins are constructed and their presence in the sequence is verified. The predicted pseudoknots can then be further analysed using bioinformatics or laboratory techniques.

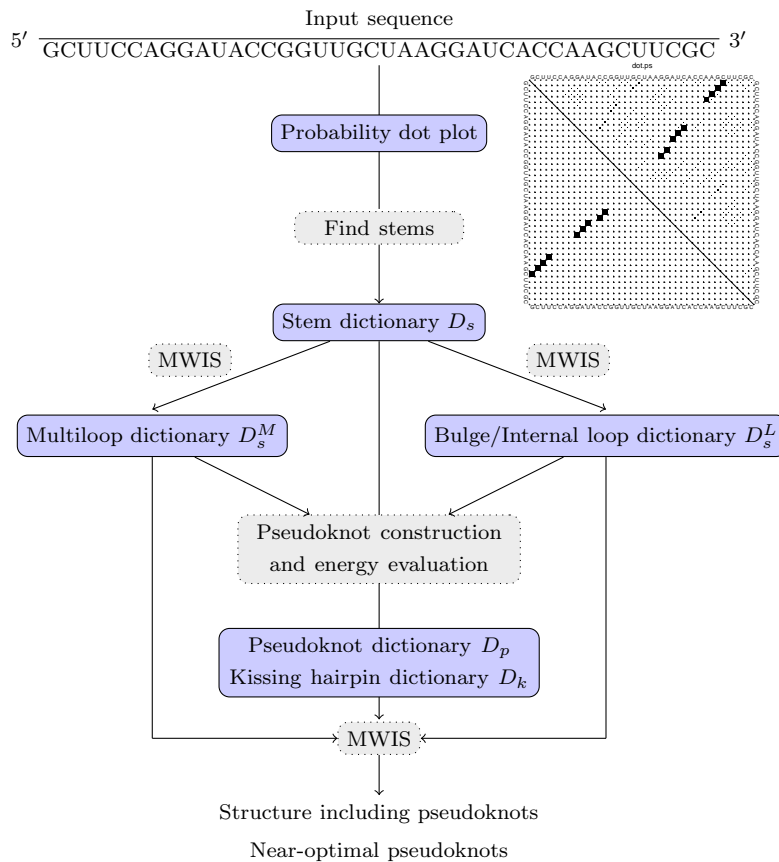


Figure 1: Workflow for the prediction of pseudoknots in a given RNA sequence. The structure returned as an output may contain both H-type pseudoknots and kissing hairpin type pseudoknots. A number of near-optimal H-type pseudoknots and kissing hairpins may also be reported. MWIS stands for maximum weight independent set calculation.

¹<http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi>

2 Installation

DotKnot has been written in Python and uses RNAfold and RNAeval from the Vienna RNA Package for producing the probability dot plot and for free energy evaluation of secondary structure elements. To get DotKnot to work on your local machine, you need to get Python and the Vienna RNA package running. DotKnot has been tested on Linux, Windows using Cygwin and Mac OS.

2.1 First step

Unpack DotKnot in your desired location:

```
tar -zxvf DotKnot_1.3.2.tar.gz
```

2.2 Second step

If you already have Python on your local machine, go to the next step. Otherwise, you need to install Python to run DotKnot, see <http://www.python.org>².

2.3 Third step

You need to install the Vienna RNA package from source. The Vienna RNA package is already provided in the DotKnot folder for one reason. The default cutoff in RNAfold for the base pair probabilities which are displayed in the dotplot is 1E-5, however for pseudoknot prediction we need a much lower cutoff of 1E-11. In the Vienna RNA package which comes with DotKnot, this cutoff has already been changed by hand in the C source code.

You need to switch to the directory and unpack, configure and make:

```
cd DotKnot_1.3.2/src
tar -zxvf ViennaRNA-1.8.5.tar.gz
cd ViennaRNA-1.8.5
./configure
make
and (as root)
make install
```

For problems with installing the Vienna RNA package, see <http://www.tbi.univie.ac.at/~ivo/RNA/> and <http://www.tbi.univie.ac.at/~ivo/RNA/INSTALL.html>.

²Note that DotKnot has been tested with Python 2.6 and 2.7, not 3.x.

3 Usage

Switch to the DotKnot working directory:

```
cd DotKnot_1.3.2/scr
```

DotKnot needs to be called with an input FASTA file and optional arguments, if desired.

```
python dotknot.py <input_file> [-k] [-l] [-g]
```

3.1 Input File

The FASTA file must contain a comment line starting with > followed by the sequence. It can also contain several consecutive sequences and DotKnot will be executed for each sequence in the file.

```
> Arc-Ful-SRP short
GGGGGGUUCGGCGUCCCCUGUAACCCGAAACCGCCGAUACGCGGG
> MMTV
AAAAAACUUGUAAAGGGGCAGUCCCUAGCCCCGCUAAAAGGGGGAUG
```

For example:

```
python dotknot.py ../testdata/TMV.fasta
```

3.2 Optional Arguments

-k: Includes **kissing hairpins**. Kissing hairpins are complex and biologically relevant types of pseudoknots. Inclusion of kissing hairpins will lead to increased run time, yet produce more meaningful results.

-l: Shows top five **near-optimal local pseudoknots** in terms of two criteria: estimated free energy to length ratio and lowest estimated free energy. This can help to identify promising pseudoknot foldings and may compensate for the limitations of the energy parameters.

-g: Shows predicted **global structure** in addition to predicted pseudoknots.

For example:

```
python dotknot.py ../testdata/TMV.fasta -k -g
```

3.3 Output

Each pseudoknot is displayed in dot-bracket notation. Unpaired bases are indicated by dots. Base pairs are written as bracket pairs. The first stem of a pseudoknot is indicated by round brackets, i.e. (and). The second stem of a pseudoknot is indicated by square brackets, i.e. [and]. We also give the start and end positions of the pseudoknot with respect to the input sequence.

If the global structure option is chosen (**-g**), a CT file will be created which contains the global structure. The file name will start with the identifier given in the FASTA file.

If you want to redirect the command line output to a file, you could use

```
python dotknot.py ../testdata/TMV.fasta -k -g > output.txt
```

4 Background

DotKnot is a heuristic algorithm for RNA structure prediction including pseudoknots. Pseudoknot prediction is an intricate problem, mostly because of computational complexity and a lack of experimentally measured pseudoknot free energy parameters. DotKnot is able to predict H-type pseudoknots and kissing hairpins and uses state-of-the-art pseudoknot free energy parameters.

4.1 H-Type Pseudoknots

The most basic and prominent pseudoknots belong to the hairpin type (H-type) class (Figure 2). H-type pseudoknots are the best-studied group of pseudoknots in the literature. This relatively simple folding principle allows for an astonishingly diverse range of functional H-type pseudoknots in the cell and in viruses. DotKnot predicts recursive H-type pseudoknots where one of the stems may be interrupted by bulges or internal loops.

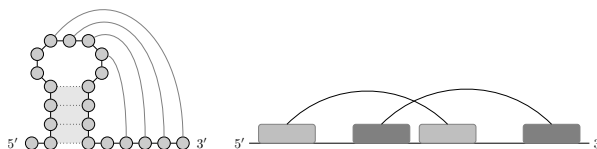


Figure 2: Example for a simple H-type pseudoknot.

4.2 Kissing Hairpins

A special type of H-type pseudoknot folds when unpaired bases in a hairpin loop bond with unpaired bases in another hairpin loop. This pseudoknot type is called an intramolecular kissing hairpin (Figure 3). It has been found in a large number of RNA molecules and participates in a range of biological functions. In many cases, the kissing base pairs are a long-range interaction which may span hundreds of nucleotides.

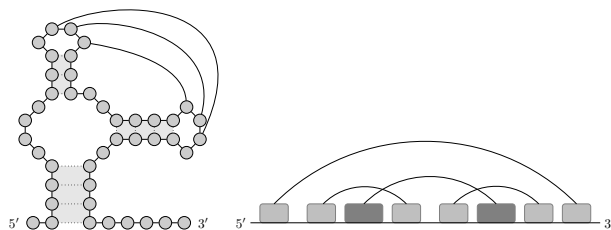


Figure 3: Example for a kissing hairpin type of pseudoknot.

4.3 Free Energy Model

Due to the complexity of pseudoknot prediction based on free energy minimization, most methods use severe restrictions of allowed pseudoknot topologies and/or unrealistic simplified energy parameters. *DotKnot* separates the steps of secondary structure formation and pseudoknot folding in its algorithm, which results in a set of pseudoknot candidates. This allows for easy integration of non-additive pseudoknot energy models, which can drastically improve prediction accuracy. Non-additive H-type pseudoknot energy models are based on the important interference between opposite stems and loops. *DotKnot* makes full use of the non-additive virtual bond model described in ?? for H-type pseudoknots with interhelix loop size ≤ 6 nt. For all other types of pseudoknots, a heuristic free energy estimation is used as no experimentally measured data is available.

4.4 Local Pseudoknots

The sets of pseudoknot candidates D_p and D_k (Figure 1) contain all recursive H-type pseudoknot and kissing hairpin candidates with negative free energy for a given sequence and can be easily scanned for promising local pseudoknots. This can help to identify near-optimal pseudoknot foldings in addition to the best global folding and may compensate for the limitations of the energy parameters.

The best local H-type pseudoknots in terms of two criteria are displayed if desired by the user. First, *DotKnot* returns the best pseudoknots in terms of estimated free energy to length ratio. This helps to identify local pseudoknots and will favour pseudoknots with compact structure and low free energy. Second, *DotKnot* returns pseudoknots with lowest estimated free energy. This returns pseudoknots with lowest free energy, regardless of their lengths.

For each criterion, a default number of five pseudoknots is returned. However, you may choose to set this threshold to a higher number by changing the value of global variable `NUMBER_OF_BEST_PKS` in the source file `dotknot.py`.

4.5 Global Structure

In addition to the predicted pseudoknots, the user can choose to see the global predicted structure. Note that *DotKnot* is a specialized pseudoknot folding tool and does not aim to compete with free energy minimization algorithms for predicting pseudoknot-free structures such as:

- `mfold` (<http://mfold.rna.albany.edu/?q=mfold>) or
- `RNAfold` (<http://rna.tbi.univie.ac.at/cgi-bin/RNAfold.cgi>).

5 Examples

5.1 First Example

Consider the 3'-UTR of the tobacco mosaic virus (TMV). Calling DotKnot gives the following result:

```
> python dotknot.py ../testset_negative/TMV.fasta -k -g
Include kissing hairpins.
Show predicted global structure.
```

```
>TMV
UGCAACUUGAGGUAGUCAAGAUGCAUAAUAAUAAACGGAUUGUGUCCGUAAUCACACGUGGUGCGUACGAUAACGCAUAG
UGUUUUUCCCUCCACUUAUAAUCGAAGGGUUGUGUCUUGGAUCGCGCGGGUCAAAUGUAUAUGGUUCAUAUACAUCGCGCAG
GCACGUAAUAAAGCGAGGGGUUCGAAUCCCCCGUUAACCCCGGUAGGGGCCCA
Sequence length: 214
```

```
DotKnot is running...
Predicting pseudoknots...
Predicting kissing hairpins...
```

Detected pseudoknots and kissing hairpins:

```
35 56 -10.97
ACGGAUUGUGUCCGUAAUCACA
((((([[[[[]]))))...]]]]
```

```
57 78 -12.92
CGUGGUGCGUACGAUAACGCAU
(((.[[[[[]]))...]]]]]
```

```
79 108 -8.16
AGUGUUUUUCCCUCCACUUAUAAUCGAAGGG
((((....[[[[]])))...]]]]
```

```
111 176 -26.94
GUGUCUUGGAUCGCGCGGGUCAAAUGUAUAUGGUUCAUAUACAUCGCGAGGCACGUAAUAAAGCGA
((((....[[[[( (((((((....((((((...))))))))))))))....]]]]
```

```
192 210 -13.83
CCGUUACCCCGGUAGGGG
(((...[[[[]]))..]]]]
```

Predicted global structure

```
UGCAACUUGAGGUAGUCAAGAUGCAUAAUAAUAAACGGAUUGUGUCCGUAAUCACACGUGGUGCGUACGAUAACGCAUAG
UGUUUUUCCCUCCACUUAUAAUCGAAGGGUUGUGUCUUGGAUCGCGCGGGUCAAAUGUAUAUGGUUCAUAUACAUCGCGCAG
GCACGUAAUAAAGCGAGGGGUUCGAAUCCCCCGUUAACCCCGGUAGGGGCCCA
((((((((....))))))....((((([[[[[]]))...]]]]((([[[[[]]))...]]]]((
((....[[[[]]))....]]]]..((((....[[[[( (((((((....((((((...))))))))))))))
))))....]]]]((((....))))((....[[[[]]))...]]]]....
```

A visualization using VARNA (?) gives the following global structure.

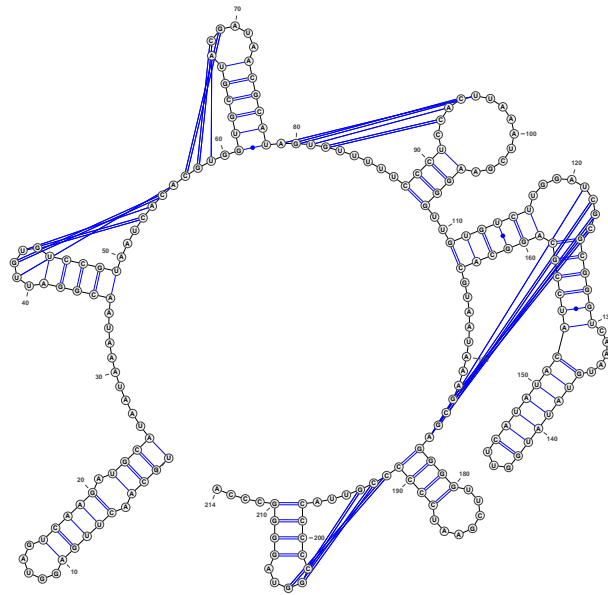


Figure 4: DotKnot's prediction of the 3'-UTR of the tobacco mosaic virus (TMV).

5.2 Second Example

Consider the PreQ1 riboswitch (PseudoBase entry PKB304). Calling DotKnot with the local pseudoknot option gives the following result:

```
> python dotknot.py ../testset_negative/PreQ1_Riboswitch.fasta -k -g -l
Include kissing hairpins.
Show predicted global structure.
Show best local pseudoknots.
```

```
>PseudoBase entry PKB304 PreQ1 riboswitch
AGAGGUUCUAGCUACACCCUCUAUAAAAAACUAA
Sequence length: 34
```

```
DotKnot is running...
Predicting pseudoknots...
Predicting kissing hairpins...
```

No pseudoknots or kissing hairpins were detected.

Best 5 pseudoknots and kissing hairpins in terms of energy to length ratio:

```
4 23 -4.84
GGUUCUAGCUACACCCUCUA
(((...[[[...]])...]])
1 33 -6.97
AGAGGUUCUAGCUACACCCUCUAUAAAAAACUA
((((...[[[.....]])...]])
4 33 -3.3
GGUUCUAGCUACACCCUCUAUAAAAAACUA
(((...[[[...]])...]])
3 33 -0.5
AGGUUCUAGCUACACCCUCUAUAAAAAACUA
(((...[[[.....]])...]])
```

Best 5 pseudoknots and kissing hairpins in terms of free energy:

```
1 33 -6.97
AGAGGUUCUAGCUACACCCUCUAUAAAAACUA
((((...[[[.....]])))).]]]
4 23 -4.84
GGUUCUAGCUACACCCUCUA
(((...[[[.....]]))..]]]
4 33 -3.3
GGUUCUAGCUACACCCUCUAUAAAAACUA
(((...[[[.....]])))).]]]
3 33 -0.5
AGGUUCUAGCUACACCCUCUAUAAAAACUA
((((...[[[.....]])))).]]]
```

Predicted global structure

```
AGAGGUUCUAGCUACACCCUCUAUAAAAACUAA
((((.....)))).]]]
```

DotKnot does not detect pseudoknots in the global structure. Looking at the local near-optimal pseudoknots with best length-normalized free energy, the true positive pseudoknot structure is found:

```
Position: 1 ... 33
AGAGGUUCUAGCUACACCCUCUAUAAAAACUA
((((...[[[.....]])))).]]]
Estimated free energy: -6.97 kcal/mol
```

6 Support

For comments, suggestions and reporting bugs please contact janaspe@csse.uwa.edu.au.

7 If You Find DotKnot Useful

Please cite:

- Sperschneider, J. & Datta, A. (2010). DotKnot: pseudoknot prediction using the probability dot plot under a refined energy model. *Nucleic Acids Res*, 38 (7): e103.
- Sperschneider, J., Datta, A., & Wise, M.J. (2011). Heuristic RNA pseudoknot prediction including intramolecular kissing hairpins. *RNA*, 17 (1): 27-38.

8 Licence

DotKnot is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

For a copy of the full text of the GNU General Public License, see www.gnu.org/licenses.