Kinfold 2.0

Efficient Computation of RNA Folding Kinetics

Daniel Oser, B.Sc.

Theoretical Biochemistry Group
University of Vienna

31^{st} TBI Winterseminar in Bled, 2016

Why Kinfold was implemented

GGGAUACUCAAAAUGGGGGCGCUUUCCUUUUUGUCUGUACUGGGA...

↘

.....((((((((.....(((...(((((....)))))..)))....)))))))).

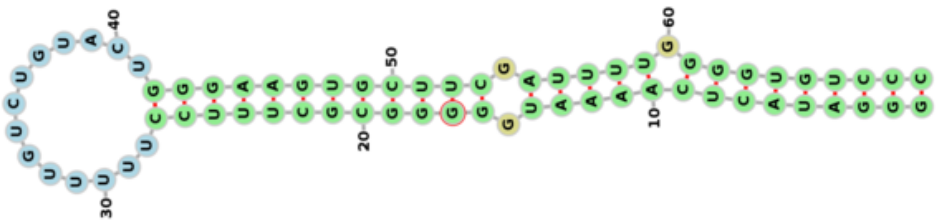
(((...))).....((((((((((....(((...(((((....)))))..)))....)))))))))).

↓ MFE

((((((((((((.(((((.........(((.((.....)).)))..)))).)))))).)))))).

((((((((((((.((((((((((((.....))))))))))))).)))))).)))))).

↙



The old Kinfold

Usage:

```
Kinfold --log LogFile --num 10 < ./InputFile.in >& Trajectorie.out
```

Input File:

```
AUCCUAUGGU  (sequence)
..........  (start structure)
(((....)))  (stop structure)
```

Output Files:

- Log:

```
#Simulation data and variables
#AUCCUAUGGU
(55884 10677 1929) X01 0.959
(26598 5835 49747) X01 2.323
```

- Traj:

```
..........  0.00 0.192
..(....)..  3.30 55.095
.((....)).  3.20 55.095
(((....)))  2.40 55.465 X1
```

Motivation of the Master Thesis

Kinfold - 'Old but gold'

- Originally implemented by Christoph Flamm
- Updated piece by piece
- Heterogenic code
- Hard to read and update
- Not as fast as it could be
- Therefore - reimplementation

Disadvantages of Kinfold

- Memory and computational time constraints for RNA > 100 nucleotides
- led to other algorithms:
  - kinetics of RNA using a more macroscopic view
    - barriers between local minima
    - simulated using numerical integration
  - coarse grained methods (KINEFold)
    faster, but rates hard to compute
  - memorization of previous states

$\rightarrow$ Still limited to a few 100 nucleotides.

Kfold

- Eric C. Dykeman introduced Kfold in 2015
- Also Gillespie Algorithm with greater efficiency
  - Divide structure into 'loops'
  - Search neighbours, moves and rates for loops
  - Unchanged loops keep same properties. O(m) $\rightarrow$ O(1)
  - $L_i = \sum_{l=1}^{i} \phi_l > r_1 \Phi$
  - $m_j = \sum_{j=1}^{u} k_{0j} \geq \bar{\Phi} \rightarrow S_\mu$

1

1

E.C. Dykeman.

An Implementation of the Gillespie Algorithm for RNA kinetics with Sub-Linear Time Update.

Nucleic Acids Research, 43(12):5708–5715, 2015.

Loop definition

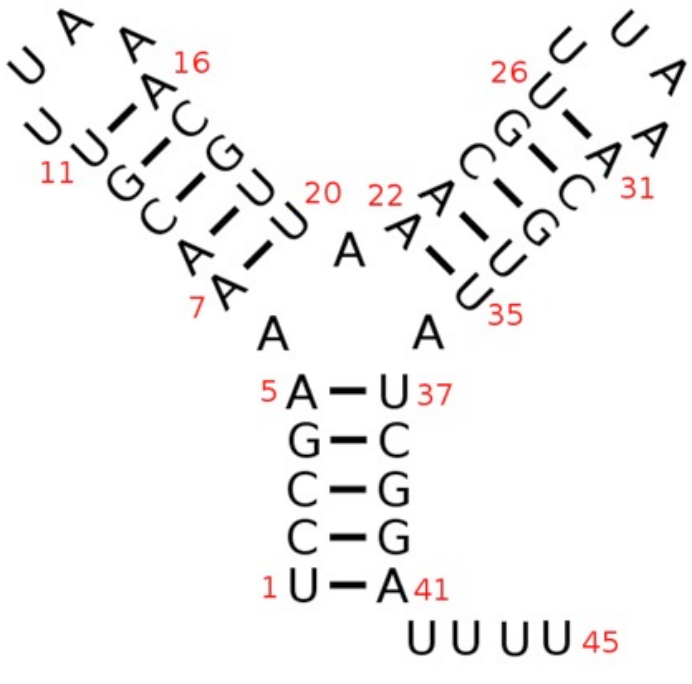


Figure: Loops in Kfold.

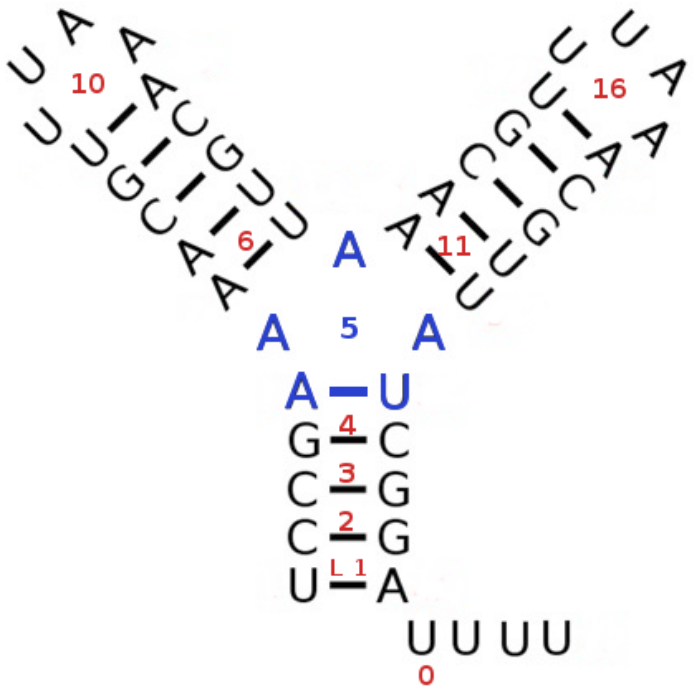
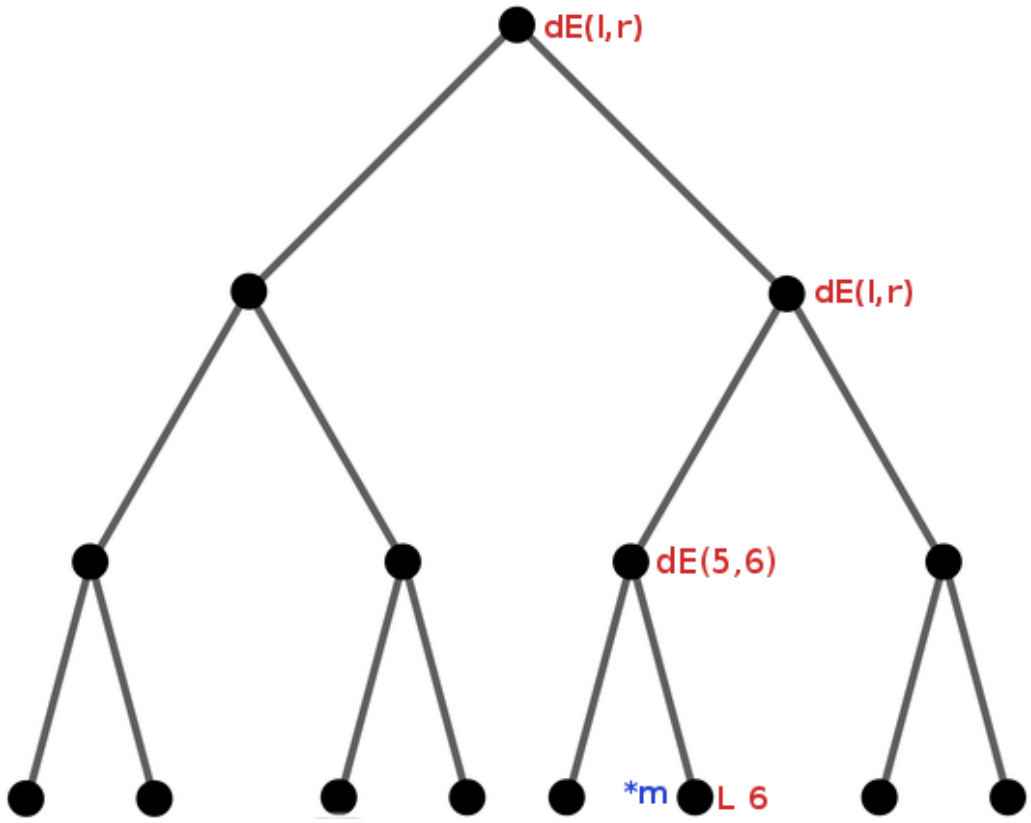


Figure: Loops in Kinfold 2.

Kinfold $\xrightarrow{\text{Improvement}}$ Kinfold 2.0

- ΔE instead of E_{tot}
- keep neighbours and rates of unchanged loops
- faster recalculation of rates
- more efficient memory - save loops to tree
- later on:
  - add shifts to moveset
  - keep loops
  - threadsave
  - add pseudoknots

BSTree and recalculation



Moves *m: [-6,-37 | 7,21 | 7,36]

My work so far

- Commandline options
- Read data
- Get BP list
- Get loops
- Find moves
- Currently: Save to tree

```
AGCGAACGCCUAAGUACAAGAAAACUUGUACUUAGGCGAAG
 0       1        u        -1       0
 1       3        p        60       1
 2       2        u        -1       1
 3       3        p        58       2
```

```
loop 0   2  0  61
loop 1   4  1  2  59  60
loop 2   8  3  4  5  38  39  56  57  58
loop 3   2  6  37
loop 4   2  7  36
```

```
addition possible between 0 - A and 61 - U
deletion possible between 1 and 60
addition possible between 2 - C and 59 - G
deletion possible between 3 and 58
```

Figure: BP, Loops, Moves

Outlook

- finish BSTree
- calculation of energies and rates
- do several trajectories
- test-run
- optimize

THANK YOU
for your
ATTENTION!

Any questions?

Further Reading I


C. Flamm, W. Fontana, I.L. Hofacker, et al.
RNA folding at elementary step resolution.
RNA (Journal), 6:325–338, 2000.


M.T. Wolfinger, W.A. Svrcek-Seiler, C. Flamm, I.L. Hofacker, P.F. Stadler.
Efficient computation of RNA folding dynamics.
Journal of Physics A, 37(17):4731–4741, 2004.


C. Flamm, I.L. Hofacker.
Beyond energy minimization: approaches to the kinetic folding of RNA.
Chemical Monthly, 139(4):447–457, 2008.