Approximate RNA folding kinetics

Gregor Entzian

University of Vienna, Faculty of Chemistry, Department of Theoretical Chemistry

entzian@tbi.univie.ac.at

Supervisor: Univ.-Prof. Dipl.-Phys. Dr. Ivo L. Hofacker

February 14, 2018
RNA Folding Kinetics

\[
\frac{dp_i(t)}{dt} = \sum_{j \neq i} [p_j(t)r_{ji} - p_i(t)r_{ij}]
\]

\[
\vec{p}_t = e^{Kt} \cdot \vec{p}_0
\]
3. = \textit{gradientwalk}(1.)
Global Flooding Algorithm - barriers

Structures sorted by energy

<table>
<thead>
<tr>
<th>x</th>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
<th>f</th>
<th>g</th>
<th>h</th>
<th>i</th>
</tr>
</thead>
</table>

Basin Hash Table

x = a

m = lowest neighbor from neighbors(x)

is E(m) > E(x)?

yes

x is a new basin

Z_k += w(x)

hash_write(x, k)

no

k = hash_lookup(m)

Z_k += w(x)

For all m in neighbors(x) with E(m) < E(x):

o = hash_lookup(m)

If o is not k:

x in contact surface

Z_{o,k} += w(x)

hash_write(x, k)

(Wolfinger et al. 2004)
Global Flooding Algorithm - *barriers*

**Structures sorted by energy**

| i | h | g | f | e | d | c | b |

**Basin Hash Table**

| a --> a |
| b --> a |

If $x = b$:

- $m = \text{lowest neighbor from neighbors}(x)$
- Is $E(m) > E(x)$?

**yes**

- $x$ is a new basin
- $Z_k += w(x)$
- $\text{hash_write}(x, k)$

**no**

- $k = \text{hash_lookup}(m)$
- $Z_k += w(x)$

For all $m$ in neighbors($x$) with $E(m) < E(x)$:

- $o = \text{hash_lookup}(m)$
- If $o$ is not $k$:
  - $x$ in contact surface
  - $Z_{o,k} += w(x)$
  - $\text{hash_write}(x, k)$

(Wolfinger et al. 2004)
Global Flooding Algorithm - barriers

Structures sorted by energy

Basin Hash Table

x = c
x is a new basin
Z_k += w(x)

m = lowest neighbor from neighbors(x)

is E(m) > E(x)?

If yes:

k = hash_lookup(m)
Z_k += w(x)

For all m in neighbors(x) with E(m) < E(x):

o = hash_lookup(m)
If o is not k:

x in contact surface
Z_{o,k} += w(x)

hash_write(x, k)

(Wolfinger et al. 2004)
Global Flooding Algorithm - *barriers*

### Structures sorted by energy

- i
- h
- g
- f
- e
- d

### Basin Hash Table

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
<td>d</td>
</tr>
</tbody>
</table>

- If \( o \) is not \( k \):
  - \( o = \text{hash\_lookup}(m) \)
  - For all \( m \) in \( \text{neighbors}(x) \) with \( E(m) < E(x) \):
    - \( o = \text{hash\_lookup}(m) \)
    - If \( o \) is not \( k \):
      - \( x \) in contact surface
      - \( Z_{\{o,k\}} += w(x) \)
      - \( \text{hash\_write}(x, k) \)

- \( m = \text{lowest neighbor from neighbors}(x) \)
  - \( \text{is } E(m) > E(x) ? \)
  - \( x \) is a new basin
    - \( Z_k += w(x) \)
    - \( \text{hash\_write}(x, k) \)
  - \( k = \text{hash\_lookup}(m) \)
    - \( Z_k += w(x) \)

(Wolfinger et al. 2004)
Global Flooding Algorithm - barriers

Structures sorted by energy

Basin Hash Table

x is a new basin
"e"

x is in contact surface

Z_k += w(x)

m = lowest neighbor from neighbors(x)

is E(m) > E(x)?

Yes:

k = hash_lookup(m)
Z_k += w(x)
hash_write(x, k)

No:

For all m in neighbors(x) with E(m) < E(x):

o = hash_lookup(m)
If o is not k:

x in contact surface

Z_{o,k} += w(x)
hash_write(x, k)

(Wolfinger et al. 2004)
Global Flooding Algorithm - *barriers*

**Global Flooding Algorithm**

**Basin Hash Table**

<table>
<thead>
<tr>
<th>Structures sorted by energy</th>
<th>Basin Hash Table</th>
</tr>
</thead>
<tbody>
<tr>
<td>a  --&gt; a</td>
<td>a  --&gt; a</td>
</tr>
<tr>
<td>b  --&gt; a</td>
<td>b  --&gt; a</td>
</tr>
<tr>
<td>c  --&gt; a</td>
<td>c  --&gt; a</td>
</tr>
<tr>
<td>d  --&gt; d</td>
<td>d  --&gt; d</td>
</tr>
<tr>
<td>e  --&gt; a</td>
<td>e  --&gt; a</td>
</tr>
<tr>
<td>f  --&gt; d</td>
<td>f  --&gt; d</td>
</tr>
</tbody>
</table>

- **x** is a new basin
- **Z**
  - **k** += **w**(x)
  - **hash_write**(x, **k**)

- **m** = lowest neighbor from **neighbors**(x)
- is **E**(m) > **E**(x)?

  - **yes**
    - **k** = **hash_lookup**(m)
    - **Z**
      - **k** += **w**(x)
    - **hash_write**(x, **k**)

  - **no**
    - For all **m** in **neighbors**(x) with **E**(m) < **E**(x):
      - **o** = **hash_lookup**(m)
      - If **o** is not **k**: **x** in contact surface
      - **Z**
        - **{o,k}** += **w**(x)
      - **hash_write**(x, **k**)

*(Wolfinger et al. 2004)*
Global Flooding Algorithm - \textit{barriers}

Structures sorted by energy

<table>
<thead>
<tr>
<th>Structures</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
</tr>
<tr>
<td>h</td>
</tr>
<tr>
<td>g</td>
</tr>
</tbody>
</table>

Basin Hash Table

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>a</td>
<td>a</td>
<td>d</td>
</tr>
<tr>
<td>b</td>
<td>a</td>
<td>a</td>
<td>d</td>
</tr>
<tr>
<td>c</td>
<td>a</td>
<td>a</td>
<td>d</td>
</tr>
<tr>
<td>d</td>
<td>d</td>
<td>d</td>
<td>a</td>
</tr>
<tr>
<td>e</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>f</td>
<td>d</td>
<td>a</td>
<td>a</td>
</tr>
<tr>
<td>g</td>
<td>a</td>
<td>a</td>
<td>a</td>
</tr>
</tbody>
</table>

$m = \text{lowest neighbor from neighbors}(x)$

\[ \text{is } E(m) > E(x)? \]

yes

- $x$ is a new basin
- $Z_k += w(x)$
- hash\_write($x$, $k$)

no

- $k = \text{hash\_lookup}(m)$
- $Z_k += w(x)$
- For all $m$ in neighbors($x$) with $E(m) < E(x)$:
  - $o = \text{hash\_lookup}(m)$
  - If $o$ is not $k$:
    - $x$ in contact surface
    - $Z_{\{o,k\}} += w(x)$
  - hash\_write($x$, $k$)

(Wolfinger et al. 2004)
Global Flooding Algorithm - *barriers*

**Structures sorted by energy**

- i
- h

**Basin Hash Table**

- a --> a
- b --> a
- c --> a
- d --> d
- e --> a
- f --> d
- g --> a
- h --> a

x = h

- m = lowest neighbor from neighbors(x)
- is E(m) > E(x)?

**yes**

- x is a new basin
- $Z_k += w(x)$
- hash_write(x, k)

**no**

- k = hash_lookup(m)
- $Z_k += w(x)$

For all m in neighbors(x) with E(m) < E(x):

- o = hash_lookup(m)
- If o is not k: x in contact surface
- $Z_{o,k} += w(x)$
- hash_write(x, k)

(Wolfinger et al. 2004)
Global Flooding Algorithm - *barriers*

(Wolfinger et al. 2004)
Local Flooding Algorithm - *pourRNA*

TodoList (PQ)  
DoneList  

\[ x \text{ in Basin} \quad Z_b += w(x) \quad \text{DoneList} += [x] \]

For all \( m \) in neighbors(x):
  
  If \( E(m) > E(x) \):
    
    TodoList += \( m \)

Descending transitions leaving \( b \):

For all \( m \) in neighbors(x):
  
  If \( E(m) < E(x) \) and \( m \) not in DoneList:
    
    \[ Z_{\text{grad. walk}(m)_b} += w(x) \]

Descending transitions entering \( b \):

For all \( m \) in neighbors(x):
  
  If \( E(m) < E(x) \) and \( m \) in DoneList:
    
    \[ Z_{\text{grad. walk}(x)_b} += w(x) \]

\[
\text{y = neighbor where } E(m) \text{ is minimal for all } m \text{ in neighbors(x)}
\]

\[
\text{y in DoneList?}
\]

yes

\[
x \text{ in Basin}
\]

no

\[
x \text{ in Contact Surface}
\]

(Mann et al. 2014)

Gregor Entzian (TBI)  
Approximate RNA folding kinetics  
February 14, 2018 7 / 11
Local Flooding Algorithm - *pourRNA*

**TodoList (PQ)**

- b
- c

**DoneList**

- a

---

x in Basin

\[ Z_b += w(x) \]

DoneList += \([x]\)

**For all** m **in** neighbors**(x)**:

**If** \( E(m) > E(x) \):

TodoList += m

Descending transitions leaving b:

**For all** m **in** neighbors**(x)**:

**If** \( E(m) < E(x) \) **and** m **not in** DoneList:

\[ Z_{b, \text{grad. walk}(m)} += w(x) \]

---

x in Contact Surface

*Descending transitions entering b:*

**For all** m **in** neighbors**(x)**:

**If** \( E(m) < E(x) \) **and** m **in** DoneList:

\[ Z_{\text{grad. walk}(x), b} += w(x) \]

---

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

**TodoList (PQ)**

| c | e |

**DoneList**

| a | b |

- $Z_b \leftarrow w(x)$
- $\text{DoneList} \leftarrow [x]$
- For all $m$ in $\text{neighbors}(x)$:
  - If $E(m) > E(x)$:
    - $\text{TodoList} \leftarrow m$
- Descending transitions entering $b$:
  - For all $m$ in $\text{neighbors}(x)$:
    - If $E(m) < E(x)$ and $m$ in $\text{DoneList}$:
      - $Z_{\text{grad. walk}(x), b} \leftarrow w(x)$
  - Otherwise:
    - For all $m$ in $\text{neighbors}(x)$:
      - If $E(m) < E(x)$ and $m$ not in $\text{DoneList}$:
        - $Z_{b, \text{grad. walk}(m)} \leftarrow w(x)$

- $x$ in Basin
  - If $E(m) > E(x)$:
    - $\text{TodoList} \leftarrow m$
  - Descending transitions entering $b$:
    - For all $m$ in $\text{neighbors}(x)$:
      - If $E(m) < E(x)$ and $m$ in $\text{DoneList}$:
        - $Z_{\text{grad. walk}(x), b} \leftarrow w(x)$
      - Otherwise:
        - For all $m$ in $\text{neighbors}(x)$:
          - If $E(m) < E(x)$ and $m$ not in $\text{DoneList}$:
            - $Z_{b, \text{grad. walk}(m)} \leftarrow w(x)$
  - Otherwise:
    - $x$ in Contact Surface

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

x in Basin  
\[ Z_b += w(x) \]  
\[ \text{DoneList} += \{x\} \]  
\[ \text{TodoList} += m \]

y in DoneList?  
yes  
no  
\[ x = c \]

y = neighbor where \( E(m) \) is minimal for all \( m \) in neighbors(\( x \))

x in Contact Surface

Descending transitions entering \( b \):

For all \( m \) in neighbors(\( x \)):
\[ \text{if } E(m) < E(x) \text{ and } m \text{ in DoneList}: \]
\[ Z_{\text{grad. walk}(x),b} += w(x) \]

Descending transitions leaving \( b \):

For all \( m \) in neighbors(\( x \)):
\[ \text{if } E(m) < E(x) \text{ and } m \text{ not in DoneList}: \]
\[ Z_{b, \text{grad. walk}(m)} += w(x) \]

---

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

- **TodoList (PQ)**
  - g

- **DoneList**
  - a
  - b
  - c
  - e

---

- **x in Basin**
  - $Z_b \leftarrow w(x)$
  - DoneList $\leftarrow \{x\}$

- For all m in neighbors(x):
  - If $E(m) > E(x)$:
    - TodoList $\leftarrow m$

- **Descending transitions leaving b:**
  - For all m in neighbors(x):
    - If $E(m) < E(x)$ and m not in DoneList:
      - $Z_{b, \text{grad. walk}(m)} \leftarrow w(x)$

---

- **x in Contact Surface**
  - Descending transitions entering b:

  - For all m in neighbors(x):
    - If $E(m) < E(x)$ and m in DoneList:
      - $Z_{\text{grad. walk}(x), b} \leftarrow w(x)$

---

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

- **TodoList (PQ)**
  - h

- **DoneList**
  - a
  - b
  - c
  - e
  - g

**x in Basin**
- $Z_b += w(x)$
- DoneList += [x]

For all m in neighbors(x):
- If $E(m) > E(x)$:
  - TodoList += m

Descending transitions leaving b:

For all m in neighbors(x):
- If $E(m) < E(x)$ and m not in DoneList:
  - $Z_{b, \text{grad. walk}(m)} += w(x)$

**x in Contact Surface**

Descending transitions entering b:

For all m in neighbors(x):
- If $E(m) < E(x)$ and m in DoneList:
  - $Z_{\text{grad. walk}(x), b} += w(x)$

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

**TodoList (PQ)**
- i

**DoneList**
- a
- b
- c
- e
- g
- h

- y = neighbor where $E(m)$ is minimal for all $m$ in neighbors($x$)
- y in DoneList?
  - yes
  - no

- if $x$ in Basin
  - $Z_b += w(x)$
  - DoneList += [x]
  - For all $m$ in neighbors($x$):
    - If $E(m) > E(x)$:
      - TodoList += m

- Descending transitions leaving $b$:
  - For all $m$ in neighbors($x$):
    - If $E(m) < E(x)$ and $m$ not in DoneList:
      - $Z_{b, \text{grad. walk}(m)} += w(x)$

- if $x$ in Contact Surface
  - Descending transitions entering $b$:
    - For all $m$ in neighbors($x$):
      - If $E(m) < E(x)$ and $m$ in DoneList:
        - $Z_{\text{grad. walk}(x),b} += w(x)$

(Mann et al. 2014)
Local Flooding Algorithm - *pourRNA*

TodoList (PQ)  
DoneList

- a
- b
- c
- e
- g
- h

x = i

y = neighbor where E(m) is minimal for all m in neighbors(x)

y in DoneList?

- yes
- no

x in Basin

Z\(_b\) += w(x)
DoneList += [x]

For all m in neighbors(x):
  - If E(m) > E(x):
    - TodoList += m

Descending transitions leaving b:

For all m in neighbors(x):
  - If E(m) < E(x) and m not in DoneList:
    - \(Z_{\text{grad. walk}(m),b} += w(x)\)

x in Contact Surface

Descending transitions entering b:

For all m in neighbors(x):
  - If E(m) < E(x) and m in DoneList:
    - \(Z_{\text{grad. walk}(x),b} += w(x)\)

(Mann et al. 2014)
Memory Trend Lines

9 sequences

<table>
<thead>
<tr>
<th>sequence</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>xbix</td>
<td>20</td>
</tr>
<tr>
<td>boris1</td>
<td>20</td>
</tr>
<tr>
<td>boris2</td>
<td>20</td>
</tr>
<tr>
<td>d25</td>
<td>25</td>
</tr>
<tr>
<td>d33</td>
<td>33</td>
</tr>
<tr>
<td>ire</td>
<td>35</td>
</tr>
<tr>
<td>bhg33</td>
<td>33</td>
</tr>
<tr>
<td>d45</td>
<td>45</td>
</tr>
<tr>
<td>SL (stem 1)</td>
<td>56</td>
</tr>
</tbody>
</table>

Approximate RNA folding kinetics

filter: $\Delta E = 5\, \text{kcal/mol}$

Memory peak per sequence length

Gregor Entzian (TBI)
Sequence: Spliced Leader
Nucleotides: 56
1 Thread,
max. Energy = 5kcal/mol:

<table>
<thead>
<tr>
<th></th>
<th>barriers</th>
<th>pourRNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory [GB]</td>
<td>8.31</td>
<td>0.06</td>
</tr>
<tr>
<td>time [m]</td>
<td>44.9</td>
<td>6.3</td>
</tr>
</tbody>
</table>

filter: $\Delta E = 5kcal/mol$
Approximate Kinetics Quality

Computing time: 0.5h

unfiltered

Delta-Energie filter with 10 kcal/mol, K-Best filter = 4, DeltaMinE filter = 1

Computing time: 0.5m

macro-states: 2913

macro-states: 35
## Advantages and Disadvantages

<table>
<thead>
<tr>
<th>barriers</th>
<th>pourRNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>optimization focus</td>
<td>runtime</td>
</tr>
<tr>
<td>runtime</td>
<td>$O(N \times n)$ † ‡</td>
</tr>
<tr>
<td>parallelization</td>
<td>difficult</td>
</tr>
<tr>
<td>bottleneck</td>
<td>memory</td>
</tr>
<tr>
<td>local filters</td>
<td>difficult</td>
</tr>
<tr>
<td>detects missing sublandscape</td>
<td>yes</td>
</tr>
<tr>
<td>connected component only</td>
<td>no</td>
</tr>
<tr>
<td>many start structures</td>
<td>no</td>
</tr>
</tbody>
</table>

$^\dagger N = \text{all secondary structures, } n = \text{number of neighbors}$  
$^\ddagger \text{RNAsubopt } O(m^3 + ml + L \log L)$  
$m = \text{sequence length, } L = \text{number of lowest structures}$
Acknowledgements

- Martin Raden
- Team Freiburg
- Team Vienna

Thank you!

pourRNA
https://github.com/ViennaRNA/pourRNA
Delta Energy Stair Climb Effect

(a) $b_1 \quad b_2$

(b) $b$

$\Delta E$

Approximate RNA folding kinetics

February 14, 2018 13 / 11
Delta Energy Filter - Partition Function and Number of States

(d) Partition Function

(e) Number of States
Delta Energy Filter - Time and Memory

(f) Partition Function

(g) Number of States
Figure: Folding kinetics of SL when using (a) exact macroscopic transition rates and (b) approximate rates resulting from restricted local flooding with deltaE=5 kcal/mol.