Constraints in RNA Secondary structure prediction

Ronny Lorenz
ronny@tbi.univie.ac.at

University of Vienna

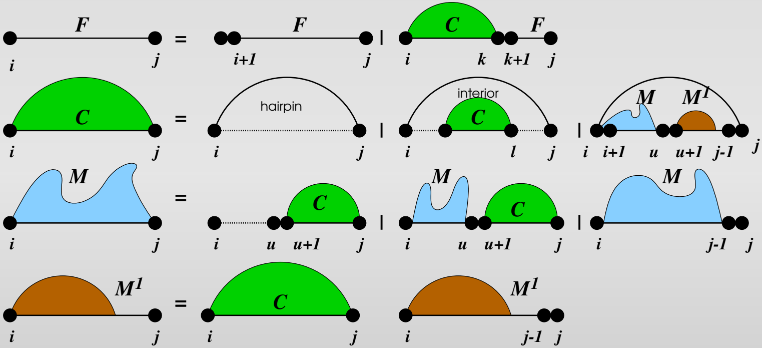
Benasque, Spain, July 27, 2015

RNA secondary structure prediction

- can be done efficiently via DP (typically) in $\mathcal{O}(n^3)$
- very good accuracy for small RNAs
- accuracy drops to 40%-70% for longer sequences
- variation of the same scheme allows one to predict:
  1. MFE
  2. Suboptimals
  3. Partition function → Equilibrium probabilities
  4. Consensus structures
  5. RNA-RNA interactions
  6. Classified DP (DoS, `RNAshapes`, `RNAbor`, `RNA2Dfold`, `RNAheliCes`)
  7. ...

RNA Secondary structure prediction

Recursive decomposition scheme (grammar)



What is constraint folding

What happens during secondary structure prediction:

- Candidate space is generated
- Candidates are evaluated (using Nearest Neighbor Energy parameters)
- Candidate scores are selected (or aggregated)

What is constraint folding

What happens during secondary structure prediction:

- Candidate space is generated
- Candidates are evaluated (using Nearest Neighbor Energy parameters)
- Candidate scores are selected (or aggregated)

But the energy model is not perfect:

- experiment (e.g. SHAPE) may suggest sth. different
- RNA is not ’alone’: bound molecules (proteins, small ligands, etc.) prohibit certain structure features and/or induce change in free energy

What is constraint folding

What happens during secondary structure prediction:

- Candidate space is generated
- Candidates are evaluated (using Nearest Neighbor Energy parameters)
- Candidate scores are selected (or aggregated)

But the energy model is not perfect:

- experiment (e.g. SHAPE) may suggest sth. different
- RNA is not ’alone’: bound molecules (proteins, small ligands, etc.) prohibit certain structure features and/or induce change in free energy

Secondary structure constraints:

- **Hard**: disallow certain parses of the decomposition scheme
- **Soft**: modify the energy contributions of the model

What is constraint folding

Hard Constraints allow for cutting out/ inserting[1] points in the secondary structure energy landscape

[1] circumvention of build-in constraints, e.g canonical base pairs

What is constraint folding

Hard Constraints allow for cutting out/ inserting[1] points in the secondary structure energy landscape

2

[1] circumvention of build-in constraints, e.g canonical base pairs

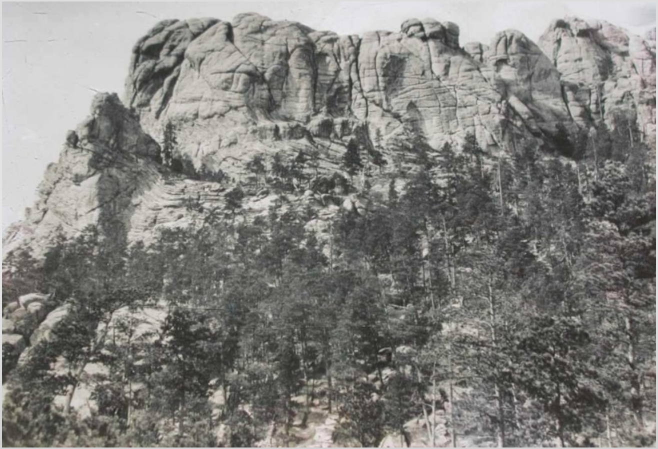
[2] Gobierno de Álvaro Colom, Guatemala

What is constraint folding

Soft Constraints allow for shifting points in the landscape up or down

What is constraint folding

Soft Constraints allow for shifting points in the landscape up or down



Mount Rushmore 1925

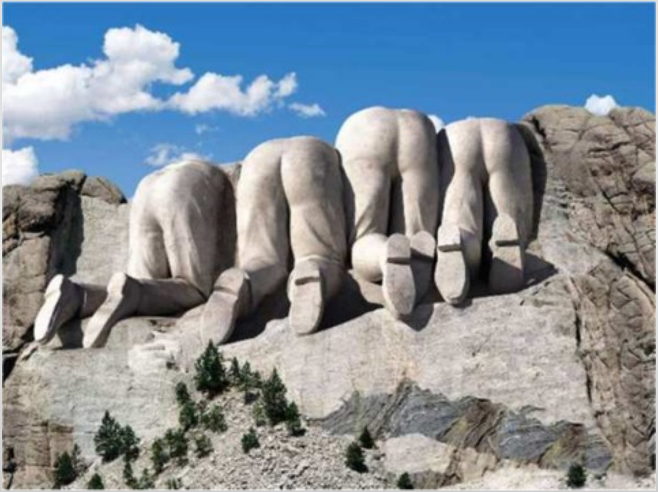
What is constraint folding

Soft Constraints allow for shifting points in the landscape up or down

Mount Rushmore Today

What is constraint folding

Soft Constraints allow for shifting points in the landscape up or down



Mount Rushmore from the back

Secondary Structure constraints

...have been used for decades

Examples

- suboptimal structures *sensu* M. Zuker
- mark modified bases (as unpaired)
- recompute optimal structure given a consensus
- simulations of translocating an RNA through a pore
- incorporate protein/ligand binding
- incorporate probing data (SHAPE, DMS, PARS)
- . . .

Secondary Structure constraints

...have been used for decades

Examples

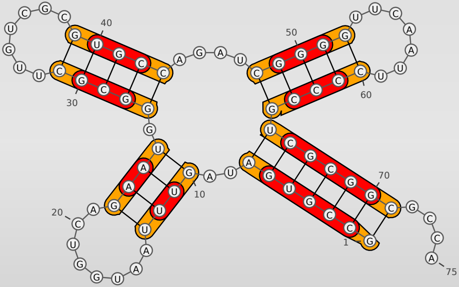
- suboptimal structures *sensu* M. Zuker
- mark modified bases (as unpaired)
- recompute optimal structure given a consensus
- simulations of translocating an RNA through a pore
- incorporate protein/ligand binding
- incorporate probing data (**SHAPE**, DMS, PARS)
- . . .

Soft constraints and SHAPE reactivity

Pseudo energy terms

- Deigan et al. [2009] (stacked pairs)

$$\Delta\ G(i) = m * ln(reactivity[i] + 1) + b$$


1
10
20
30
40
50
60
70
75
G C C G U G A U A G U U U A A U G G U C A G A A U G G G C G C U U G U C G C G U G C C A G A U C G G G G U U C A A U U C C C C G U C G C G G C G C C A

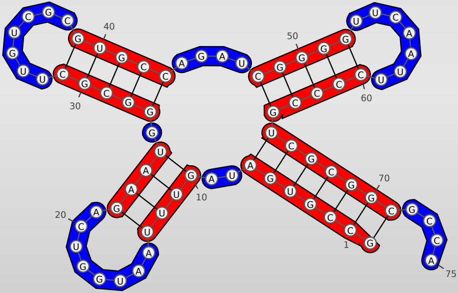
Soft constraints and SHAPE reactivity

Pseudo energy terms

- Zarringhalam et al. [2012] (unpaired bases and base pairs)

$$\Delta\ G(x, i) = \beta\ * |x - q_i|$$

$$x \in [0(unpaired), 1(paired)]$$

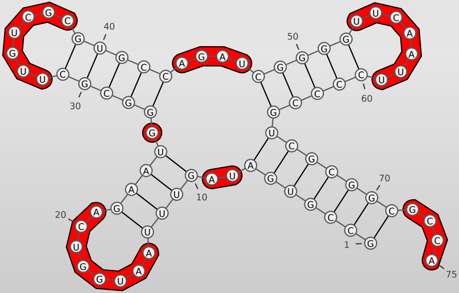


Soft constraints and SHAPE reactivity

Pseudo energy terms

- Washietl et al. [2012] (unpaired bases)
  Objective function

$$F(\vec{\epsilon}) = \sum_{i=1}^{n} \frac{\epsilon_i^2}{\tau^2} + \sum_{i=1}^{n} \frac{(p_i(\vec{\epsilon}) - q_i)^2}{\sigma^2} \rightarrow min$$



Implementations

Constraints aware secondary structure prediction programs:

Hard constraints:

- `UNAfold` *(Markham et al., 2008)*
- `ViennaRNA Package` *(Hofacker et al., 1994, Lorenz et al. 2011)*

Hard and Soft constraints:

- `RNAstructure` (SHAPE) *(Reuter et al., 2010)*
- `RNApbfold` (SHAPE) *(Washietl et al., 2012)*
- `ViennaRNA Package` $\geq$ v2.2 (SHAPE, generalized constraints)

Not to mention all the programs for specific use-cases resulting from

- code-duplication
- from-scratch implementions

What is constraint folding

Where do current implementations apply structure constraints?

- positions that are unpaired
- base pairs
- base pair stacks

Are the above implementations sufficient?

What is constraint folding

Where do current implementations apply structure constraints?

- positions that are unpaired
- base pairs
- base pair stacks

Are the above implementations sufficient?

Of course NOT!

On generalizing Hard constraints

Typical implementations:

$$N_{ij} = X_{ii} N_{i+1,j} + \sum_{k=i+1}^{j} X_{ik} N_{i+1,k-1} N_{k+1,j}$$

On generalizing Hard constraints

Typical implementations:

$$N_{ij} = X_{ii} N_{i+1,j} + \sum_{k=i+1}^{j} X_{ik} N_{i+1,k-1} N_{k+1,j}$$

Add discriminative power:

1. Go beyond Nussinov scheme

   $$\text{Substitute} \quad X \quad \text{with} \quad X^{\tau}$$

   where τ now denotes the different types of loops:

   - exterior loop
   - hairpin loops
   - interior loops (closing, enclosed)
   - components of multi-loops (closing, enclosed)

On generalizing Hard constraints

Typical implementations:

$$N_{ij} = X_{ii} N_{i+1,j} + \sum_{k=i+1}^{j} X_{ik} N_{i+1,k-1} N_{k+1,j}$$

Add discriminative power:

1. Go beyond Nussinov scheme

   $$\text{Substitute} \quad X \quad \text{with} \quad X^{\tau}$$

   where τ now denotes the different types of loops:

   - exterior loop
   - hairpin loops
   - interior loops (closing, enclosed)
   - components of multi-loops (closing, enclosed)

2. Go to full NN scheme
   Express X in terms of a boolean function

   $$f : \mathbb{N}^m \times \mathbb{D} \rightarrow 0|1$$

   with m nucleotide positions, and decomposition step $d \in \mathbb{D}$.

On generalizing Soft constraints

Position dependent pseudo energy:

$$
\begin{aligned}
E(\psi) &= E_0(\psi) + \sum_{i \in \psi^p} b_i^p + \sum_{i \in \psi^u} b_i^u \\
&= E_0(\psi) + \sum_{i=1}^{n} b_i^p + \sum_{i \in \psi^u} (b_i^u - b_i^p) \\
&= E_0(\psi) + E' + \sum_{i \in \psi^u} \delta_i
\end{aligned}
$$

Base pair specific pseudo energies:

$$
\begin{aligned}
E(\psi) &= E_0(\psi) + \sum_{(i,j) \in \psi} b_{ij}^p + \sum_{(i,j) \notin \psi} b_{ij}^u \\
&= E_0(\psi) + \sum_{i<j} b_{ij}^u + \sum_{(i,j) \in \psi} (b_{ij}^p - b_{ij}^u) \\
&= E_0(\psi) + E' + \sum_{(i,j) \in \psi} \Delta_{ij}
\end{aligned}
$$

On generalizing Soft constraints

Combine pseudo energies for single, and paired positions

- $\Delta_{ii} = \delta_i$ (single positions)
- Δ_{ij} (base pairs)

Apply the same ideas as for Hard constraints!

Add discriminative power:

1. Go beyond Nussinov scheme

$$\hat{E}_{ij}^{\tau} = E_{ij}^{\tau} + \Delta_{ij}^{\tau} + \sum_{u \in \tau} \Delta_{uu}^{\tau}$$

2. Go to full NN scheme:
   Express Δ in terms of a Real-valued function

$$f : \mathbb{N}^m \times \mathbb{D} \rightarrow \mathbb{R}$$

with m nucleotide positions, and decomposition step $d \in \mathbb{D}$.

On generalizing constraint folding

Recap: What happens during secondary structure prediction:

- Candidate space is generated
- Candidates are evaluated (using Nearest Neighbor Energy parameters)
- Candidate scores are selected (or aggregated)

On generalizing constraint folding

Recap: What happens during secondary structure prediction:

- Candidate space is generated $\rightarrow$ **Hard constraints**
- Candidates are evaluated (using Nearest Neighbor Energy parameters) $\rightarrow$ **Soft constraints**
- Candidate scores are selected (or aggregated)

Generalized constraints can be efficiently integrated into the DP recursion as a separate additional layer between candidate generation and NN energy evaluation.

On generalizing Soft constraints

What are generalized constraints good for? *(Applications)*

- loop-type dependency of hard constraints
- include protein/ligand binding contributions directly
- include 2.5D structure motifs [3]
- include other models to incorporate probing data
- ...
- **Most importantly:** Use all the above in multiple variations of the RNA secondary structure prediction algorithm (MFE, Subopt, Partition function, Consensus structures, ...)

[3] under certain conditions

On generalizing Soft constraints

What are generalized constraints good for? *(Applications)*

- loop-type dependency of hard constraints
- **include protein/ligand binding contributions directly**
- include 2.5D structure motifs [3]
- include other models to incorporate probing data
- . . .
- **Most importantly:** Use all the above in multiple variations of the RNA secondary structure prediction algorithm (MFE, Subopt, Partition function, Consensus structures, . . .)

[3] under certain conditions

Soft constraints and ligand binding

Incorporate protein-RNA binding to unpaired positions:[4]

Instead of

$$
\begin{aligned}
Q_1(c) &= Q + \hat{Q}_1 \cdot \frac{c}{k_D} \\
Q_2(c) &= Q + \hat{Q}_1 \cdot \frac{c}{k_D} + \hat{Q}_2 \cdot \frac{c}{k_D} + \hat{Q}_{12} \cdot (\frac{c}{k_D})^2 \\
&\vdots
\end{aligned}
$$

directly compute $Q(c)$ via soft constraints:

$$
\begin{aligned}
Q(c) &= \sum_{s \in \Omega} e^{-E(s)/RT} \cdot f(s,c) \\
f(s,c) &= \sum_{a \in A(s)} (\frac{c}{k_D})^{|a|}
\end{aligned}
$$

[4] refers to talk by Ralf Bundschuh

Soft constraints and ligand binding

Incorporate protein-RNA binding to unpaired positions:[4]

Instead of

$$\begin{aligned} Q_1(c) &= Q + \hat{Q}_1 \cdot \frac{c}{k_D} \\ Q_2(c) &= Q + \hat{Q}_1 \cdot \frac{c}{k_D} + \hat{Q}_2 \cdot \frac{c}{k_D} + \hat{Q}_{12} \cdot (\frac{c}{k_D})^2 \\ &\vdots \end{aligned}$$

directly compute $Q(c)$ via soft constraints:

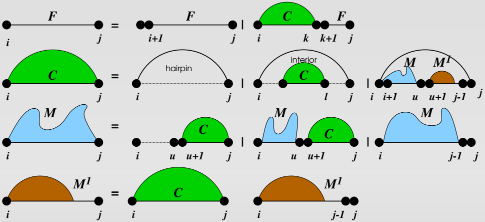
$$\begin{aligned} Q(c) &= \sum_{s\in\Omega} e^{-E(s)/RT} \cdot f(s,c) \\ f(s,c) &= \sum_{a\in A(s)} (\frac{c}{k_D})^{|a|} \end{aligned}$$

Sounds great, but it doesn't work!

[4] refers to talk by Ralf Bundschuh

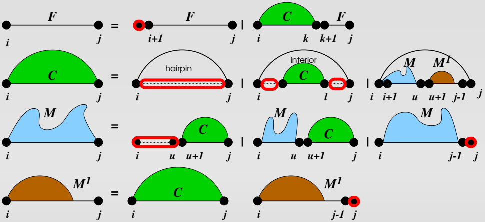
Soft constraints and ligand binding

Nearest Neighbor Model



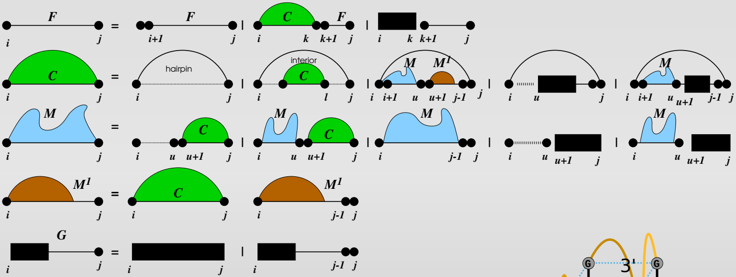
Soft constraints and ligand binding

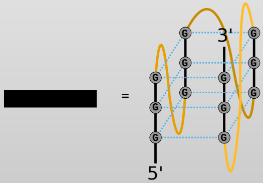
Nearest Neighbor Model



RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes[5]

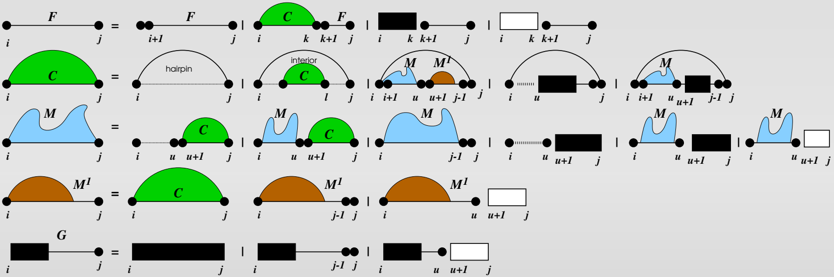




[5] Lorenz et al., (2012, 2013)

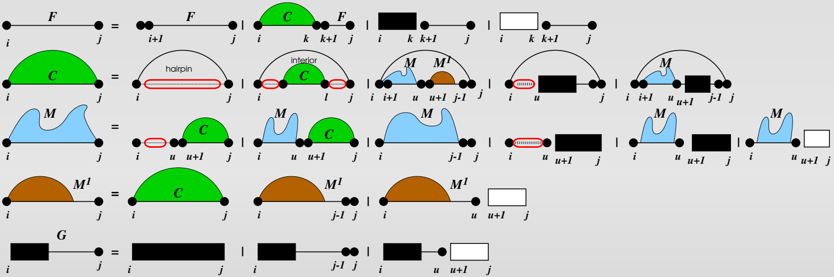
RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes and Ligands



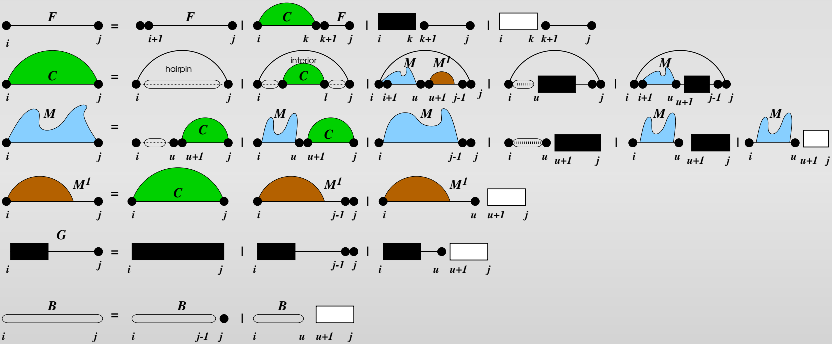
RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes and Ligands



RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes and Ligands[6]



[6] (in preparation)

Constraints within the ViennaRNA Package 2.2

- Extension of the folding grammar to include ligand binding[7]
- Easy to use input for executable programs exposing X^{τ}, and $\Delta^{(\tau)}$
- Convenience input for SHAPE data
- Full NN constraints accessible via `RNAlib` v3.0 API [8]
- Generalized constraints currently available for: `RNAfold`, `RNAcofold`, `RNAsubopt`, and `RNAalifold`

ViennaRNA Package 2.2.0 RC-3 already available

[7] will be part of the final release of v2.2.0

[8] backward compatibility until release of ViennaRNA Package v3.x

Thanks to

- Dominik Luntzer
- Yann Ponty
- Andrea Tanzer
- Peter F Stadler
- Ivo L Hofacker
- remaining TBI team

Thank You for your attention!

This work was funded in parts by the Austrian/French project ‘RNAlands‘, FWF-I-1804-N28 and ANR-14-CE34-0011

Backup slides

Using constraint folding

SHAPE reactivity input file

```
9    -999       # No reactivity information
10   -999
11   0.042816   # normalized SHAPE reactivity
12   0          # also a valid SHAPE reactivity
13   0.15027
...
42   0.16201
```

Constraints definition file (Generalized version of UNAfold constraints)

```
F i 0 k   [TYPE] [ORIENTATION] # Force nucleotides i...i+k-1 to be paired
F i j k   [TYPE] # Force helix of size k starting with (i,j) to be formed
P i 0 k   [TYPE] # Prohibit nucleotides i...i+k-1 to be paired
P i j k   [TYPE] # Prohibit pairs (i,j),...,(i+k-1,j-k+1)
P i-j k-l [TYPE] # Prohibit pairing between two ranges
C i 0 k   [TYPE] # Nucleotides i,...,i+k-1 must appear in context TYPE
C i j k          # Remove pairs conflicting with (i,j),...,(i+k-1,j-k+1)
E i 0 k e        # Add pseudo-energy e to nucleotides i...i+k-1
E i j k e        # Add pseudo-energy e to pairs (i,j),...,(i+k-1,j-k+1)
```

with

```
[TYPE]        = { E, H, I, i, M, m, A }
[ORIENTATION] = { U, D }
```

Using constraint folding

RNAlib v3.0 API usage

```
/* obtain a data structure for folding */
vc = vrna_get_fold_compound(sequence, ...);
/* add hard constraints */
vrna_hc_add(vc, constraints_file, ...);
/* add SHAPE reactivity data and apply Mathews conversion
   for pseudo energies */
vrna_sc_add_mathews(vc, shape_data, ...);
/* fold it */
vrna_fold(vc);
```

Scripting language (Perl/Python) support will follow