Constraints in RNA Secondary structure prediction

Ronny Lorenz
ronny@tbi.univie.ac.at

University of Vienna

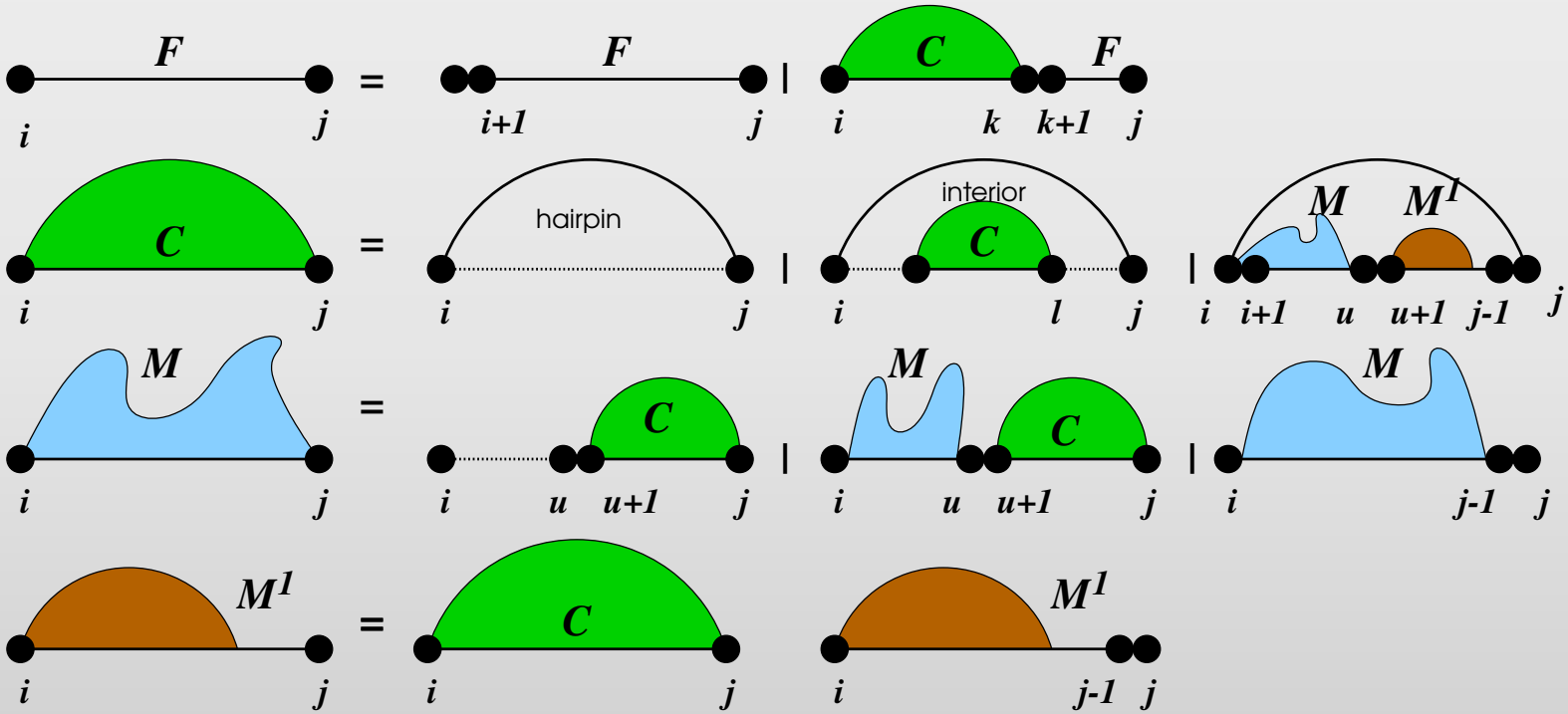
Ljubljana, Slovenia, February 17, 2016

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 $\rightarrow$ 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 $\rightarrow$ **Hard constraints**
- Candidates are evaluated (using Nearest Neighbor Energy parameters) $\rightarrow$ **Soft constraints**
- 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**: determine the candidate space
- **Soft**: act on candidate evaluation

Secondary Structure constraints

...have been used for decades

Examples

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

Constraints aware secondary structure prediction programs

Most implementations are for specific use-cases:

- constraints on positions that are unpaired, base pairs, base pair stacks
- code-duplication
- from-scratch implementions

Examples:

- `UNAfold` [1] (**hard**)
- `ViennaRNA Package` [2] (**hard**)
- `RNAstructure` [3] (**hard** + **soft**, SHAPE)
- `RNApbfold` [4] (**hard** + **soft**, SHAPE)

Are the above implementations sufficient?

---

[1] (Markham et al., 2008)
[2] (Hofacker et al., 1994, Lorenz et al. 2011)
[3] (Reuter et al., 2010)
[4] (Washietl et al., 2012)

Constraints aware secondary structure prediction programs

Most implementations are for specific use-cases:

- constraints on positions that are unpaired, base pairs, base pair stacks
- code-duplication
- from-scratch implementions

Examples:

- `UNAfold` [1] (**hard**)
- `ViennaRNA Package` [2] (**hard**)
- `RNAstructure` [3] (**hard** + **soft**, SHAPE)
- `RNApbfold` [4] (**hard** + **soft**, SHAPE)

Are the above implementations sufficient? Of course NOT!

---

[1] (Markham et al., 2008)
[2] (Hofacker et al., 1994, Lorenz et al. 2011)
[3] (Reuter et al., 2010)
[4] (Washietl et al., 2012)

On generalizing Hard constraints

Typical implementations:

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

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

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

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

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 [5]
- easy adaptation to new models of incorporating probing data
- . . .
- **Most importantly:** Use all the above in multiple variations of the RNA secondary structure prediction algorithm (MFE, Subopt, Partition function, Consensus structures, . . .)

[5] 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 [5]
- easy adaptation to new models of incorporating probing data
- . . .
- **Most importantly:** Use all the above in multiple variations of the RNA secondary structure prediction algorithm (MFE, Subopt, Partition function, Consensus structures, . . .)

[5] under certain conditions

Soft constraints and ligand binding

Ligand binding to an aptamer motif:

$$Q = \sum_{s \in \Omega} e^{-E(s)/RT}, \quad \text{and} \quad p(M) = \frac{\sum_{s|M \in s} e^{-E(s)/RT}}{Q}$$

Adding the contribution of one ligand L bound to a single aptamer motif A:

$$Q_L = Q + Q^A \cdot e^{-\Delta G/RT}, \quad \text{with} \quad Q^A = \sum_{s|A \in s} e^{-E(s)/RT}, \Delta G = RT \ln \frac{K_d}{c}$$

More than one aptamer motif $A_1, A_2, ..$ per sequence:

$$Q_L = Q + (Q^{A_1} + Q^{A_2}) \cdot e^{-\Delta G/RT} + Q^{A_1 A_2} \cdot e^{-2\Delta G/RT} + \ldots$$

Soft constraints and ligand binding

Ligand binding to an aptamer motif:

With generic soft-constraints:

$$
\begin{aligned}
Q_L &= \sum_{s \in \Omega} e^{-E(s)/RT} \cdot f(s) \\
f(s) &= \sum_{a \in \mathcal{P}(\{A_1, A_2, \ldots\}) \cap s} e^{-|a| \Delta G / RT}
\end{aligned}
$$

Soft constraints and ligand binding

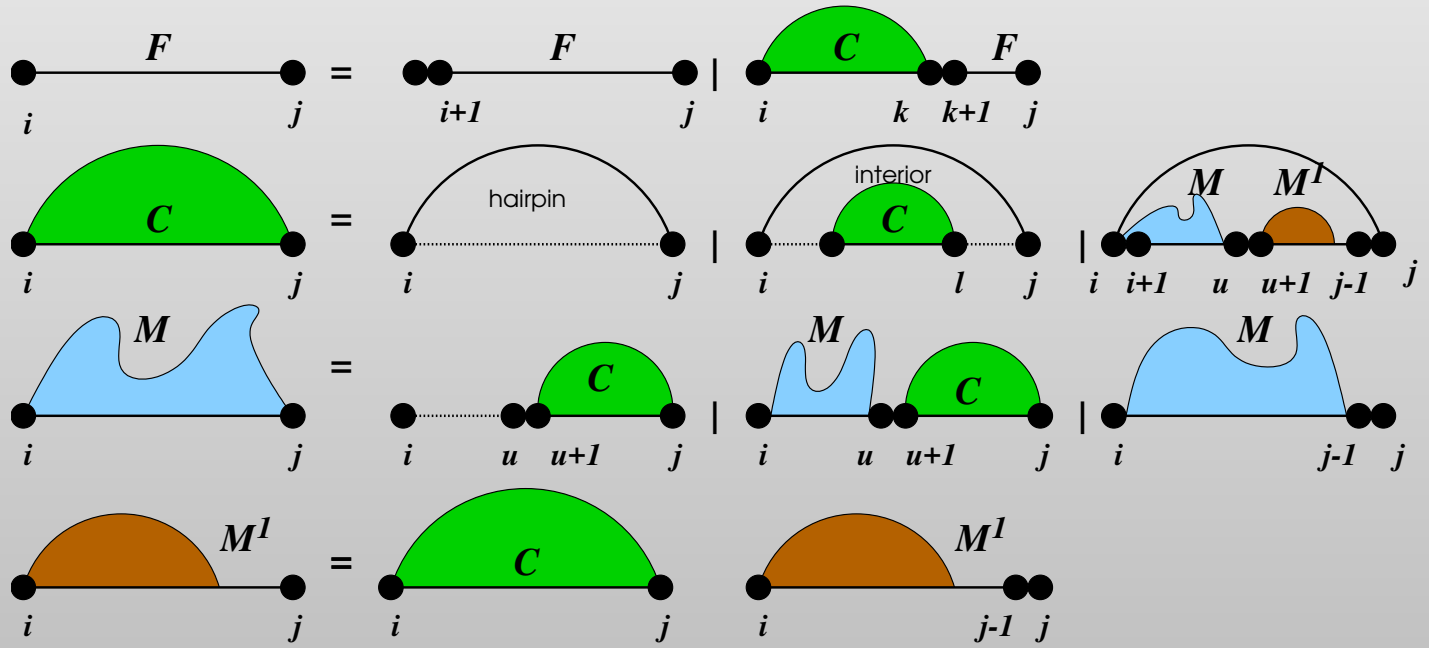
Ligand binding to an aptamer motif:

With generic soft-constraints:

$$Q_L = \sum_{s \in \Omega} e^{-E(s)/RT} \cdot f(s)$$

$$f(s) = \sum_{a \in \mathcal{P}(\{A_1, A_2, \ldots\}) \cap s} e^{-|a| \Delta G/RT}$$

Sounds great, but it doesn't work in general!



Soft constraints and ligand binding

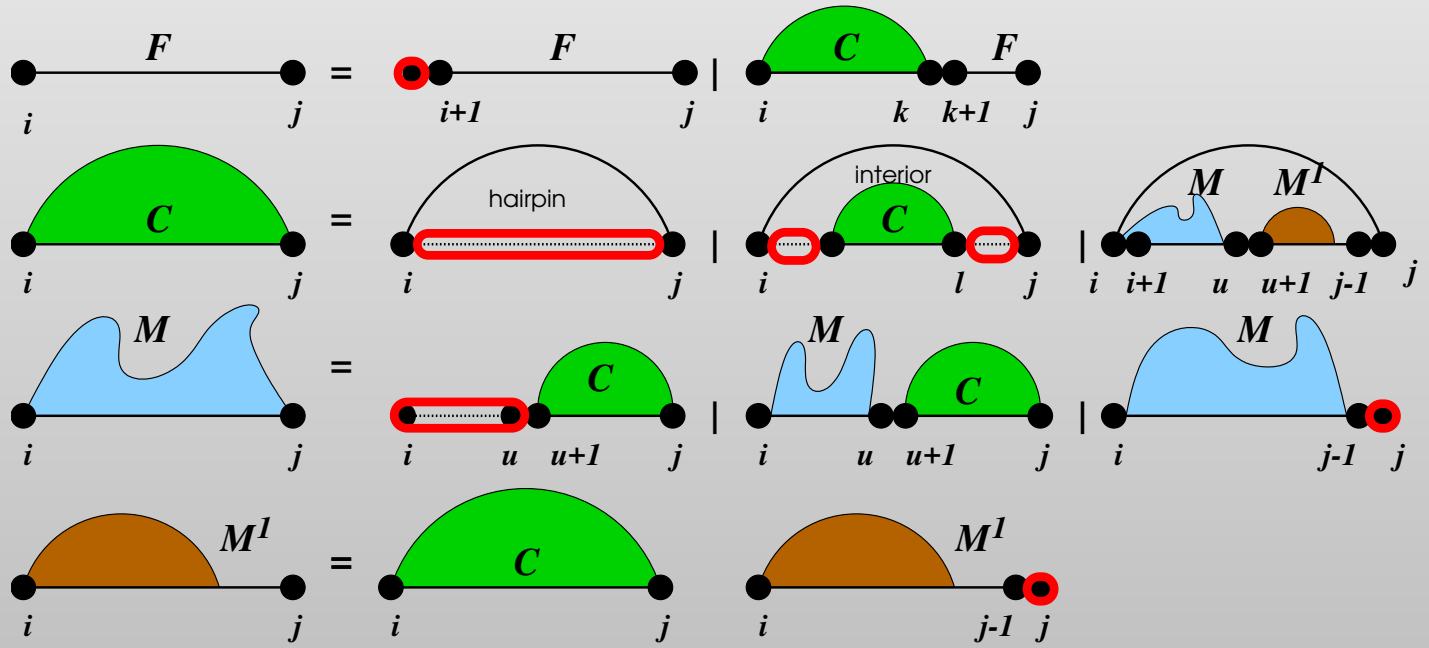
Ligand binding to an aptamer motif:

With generic soft-constraints:

$$Q_L = \sum_{s \in \Omega} e^{-E(s)/RT} \cdot f(s)$$

$$f(s) = \sum_{a \in \mathcal{P}(\{A_1, A_2, \ldots\}) \cap s} e^{-|a| \Delta G/RT}$$

Sounds great, but it doesn't work in general!



Soft constraints and ligand binding

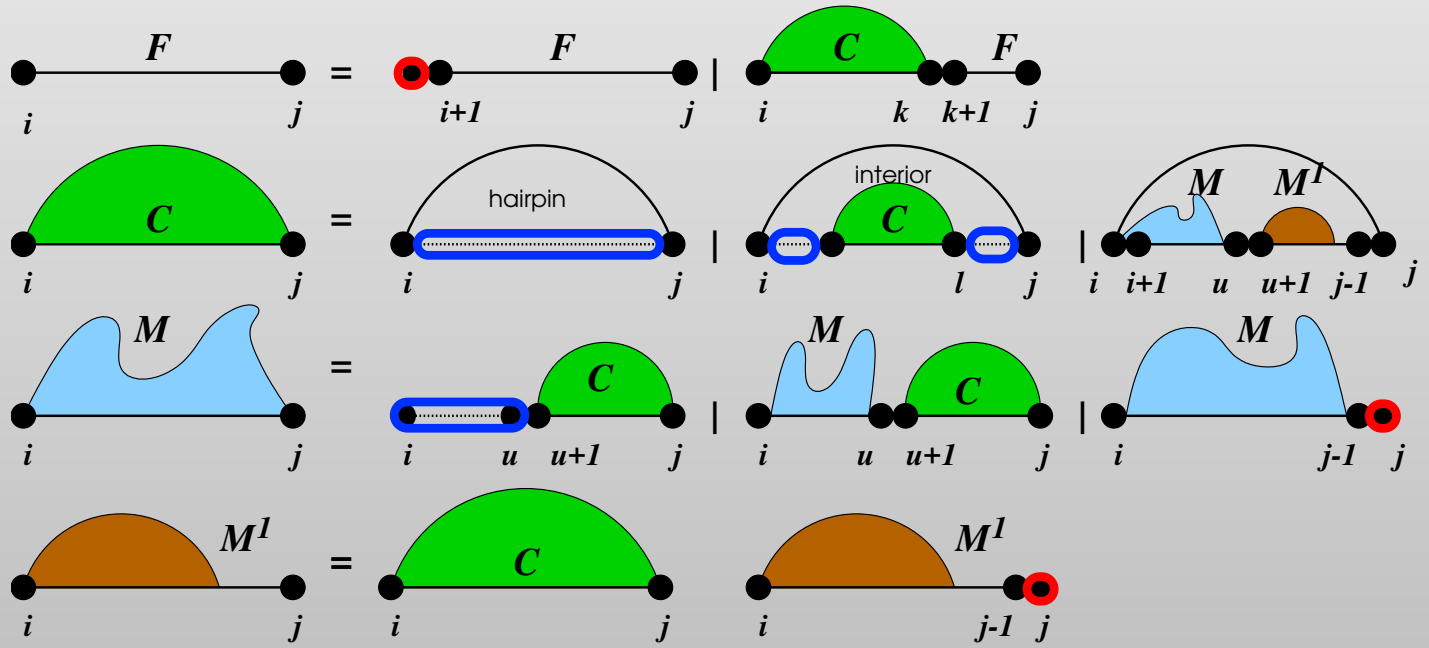
Ligand binding to an aptamer motif:

With generic soft-constraints:

$$Q_L = \sum_{s \in \Omega} e^{-E(s)/RT} \cdot f(s)$$

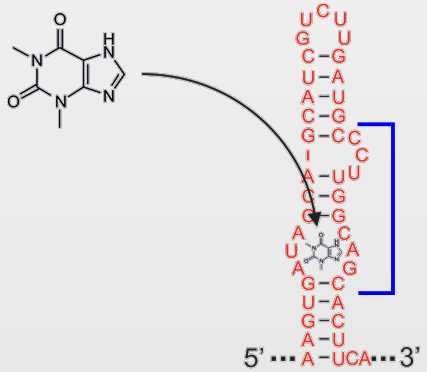
$$f(s) = \sum_{a \in \mathcal{P}(\{A_1, A_2, \ldots\}) \cap s} e^{-|a| \Delta G/RT}$$

Sounds great, but it doesn't work in general!



Soft constraints and ligand binding - hairpin/interior loop motifs

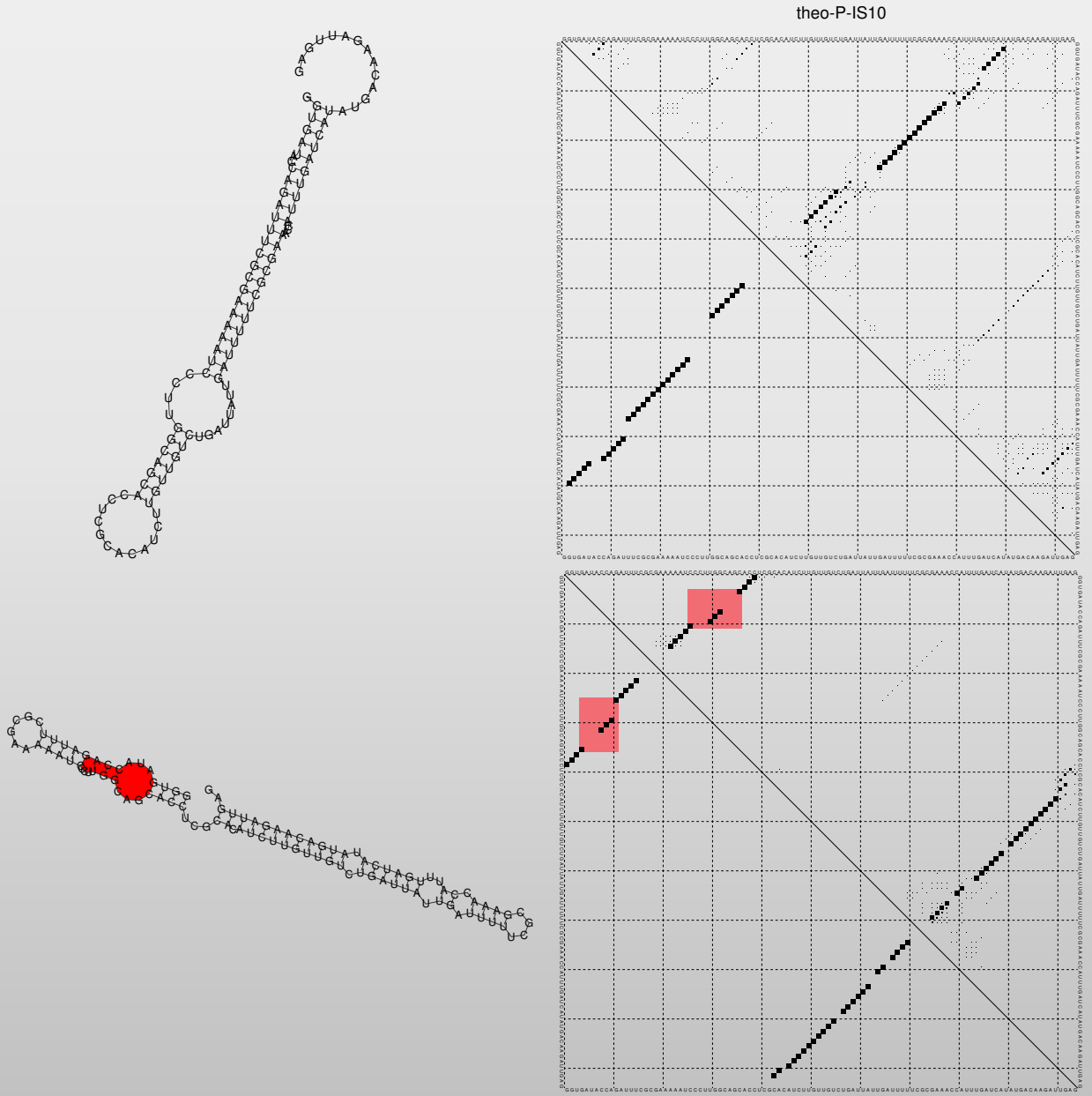
Theophylline $K_d = 0.32\mu M$ (Jenison et al. 1994):



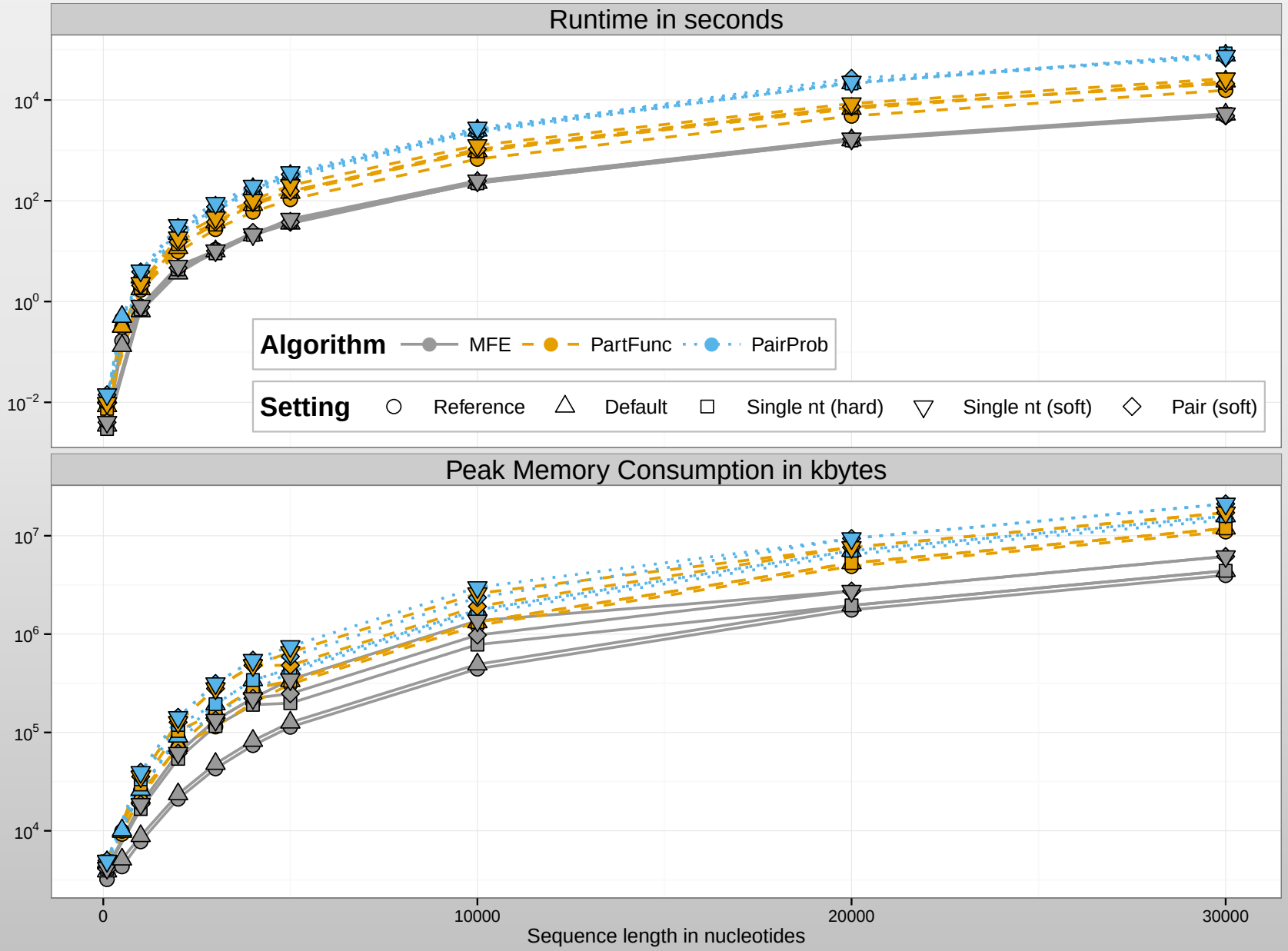

```
$ cat theo.fa
>theo-P-IS10
GGUGAUACCAGAUUUCGCGAAAAAUCCCUUGGCAGCACCUCGCACAUCUUGUUGUCUGAUUAUUGAUUUUUCGCGAAACCAUUUGAUCAUAUGACAAGAUUGAG

$ RNAfold -p < theo.fa
>theo-P-IS10
GGUGAUACCAGAUUUCGCGAAAAAUCCCUUGGCAGCACCUCGCACAUCUUGUUGUCUGAUUAUUGAUUUUUCGCGAAACCAUUUGAUCAUAUGACAAGAUUGAG
.(((((..(((((((((((((((((....(((((((.............)))))))........)))))))))))))...)))))))))..... (-26.50)
.((((.{..{{{{((((((((((((((....(((((((.............)))))))),,......)))))))))))))}..,}}}}}))))}..... [-28.18]
.(((((...(((((((((((((((....(((((((.............)))))))........)))))))))))))...))).)))))..... {-23.80 d=13.66}
 frequency of mfe structure in ensemble 0.0656727; ensemble diversity 20.37

$ RNAfold -p --motif="GAUACCAG&CCCUUGGCAGC,(...((((&)...)))...),-9.22" --verbose
>theo-P-IS10
read ligand motif: GAUACCAG&CCCUUGGCAGC, (...((((&)...)))...), -9.220000
GGUGAUACCAGAUUUCGCGAAAAAUCCCUUGGCAGCACCUCGCACAUCUUGUUGUCUGAUUAUUGAUUUUUCGCGAAACCAUUUGAUCAUAUGACAAGAUUGAG
((((...(((((((.....)))))...)))...))))...((.((((((((((.((((((.((.((((....)))).)).)).)))))).)))))))))).. (-33.82)
specified motif detected in MFE structure: (4,36) (11,26)
((((...(((((((.....)))))...)))...))))...,,.((((((((((.((((((.((.((((....)))).)).)).)))))).)))))))))).. [-35.14]
((((...(((((((.....)))))...)))...))))......((((((((((.((((((.((.((((....)))).)).)).)))))).))))))))).... {-24.20 d=4.35}
specified motif detected in centroid structure: (4,36) (11,26)
 frequency of mfe structure in ensemble 0.116952; ensemble diversity 6.71
```

Soft constraints and ligand binding

Theophylline binding to an aptamer motif:



Computational Overhead of Constraints Framework Implementation



RNA folding with Hard and Soft constraints[6]

- efficiently integrated as separate additional layer between candidate generation and NN energy evaluation
- Easy to use input for executable programs exposing X^{τ}, and Δ
- Convenience input for SHAPE data
- Convenience input for ligand binding to hairpin/interior loops
- Extension for ligands binding to consecutive stretches of unpaired nucleotides (similar to G-Quadruplex feature)[7]
- Full NN constraints accessible via `RNAlib` v3.0 API [8]
- Generalized constraints currently available for: `RNAfold`, `RNAcofold`, `RNAsubopt`, and `RNAalifold`
- available since ViennaRNA Package 2.2.0

---

[6] submitted

[7] Scheduled for ViennaRNA Package 2.3

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

Thanks to

- Caipirinha and Crocodile Burgers
- Peter F Stadler
- Dominik Luntzer
- Yann Ponty
- Andrea Tanzer
- 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 }
```

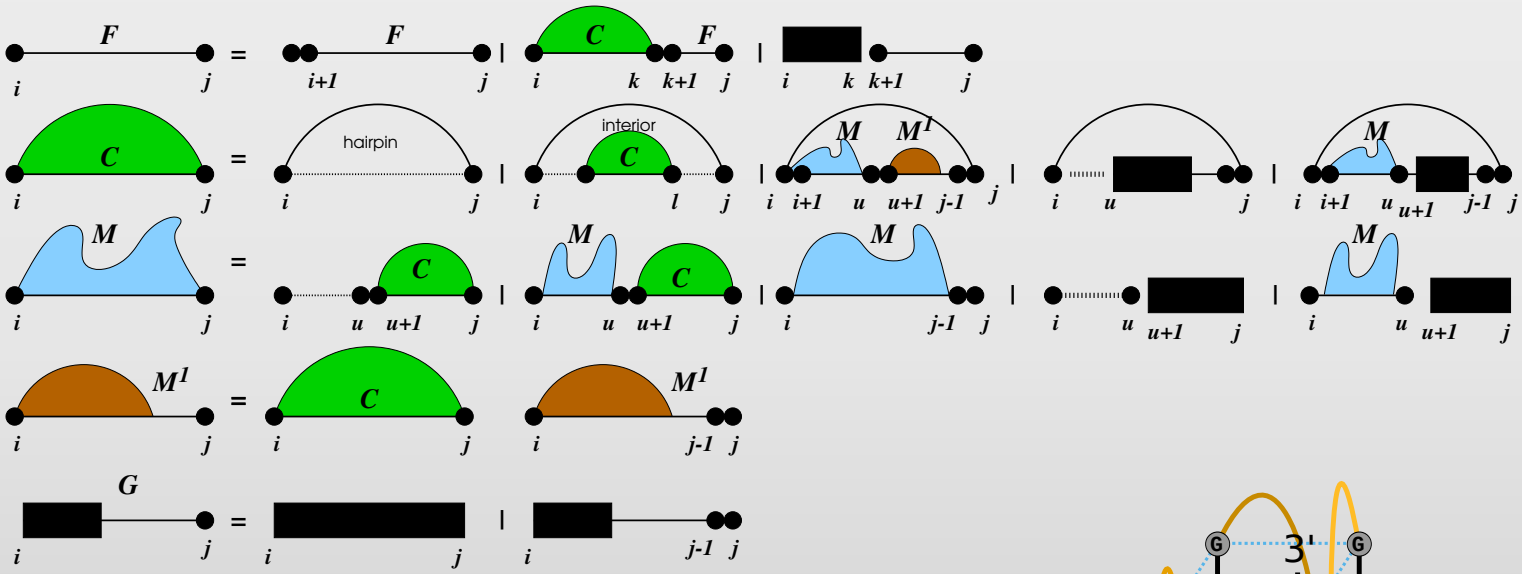
Using constraint folding
RNAlib v3.0 API usage

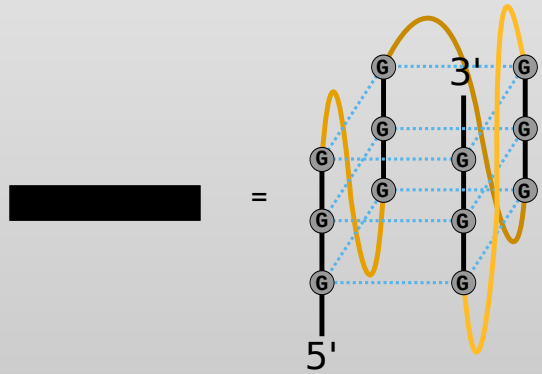
```
/* obtain a data structure for folding */
vc = vrna_fold_compound(sequence, ...);
/* add hard constraints */
vrna_hc_add(vc, constraints_file, ...);
/* add SHAPE reactivity data and apply Deigan et al. conversion
   for pseudo energies */
vrna_sc_add_SHAPE_deigan(vc, shape_data, ...);
/* fold it */
vrna_mfe(vc);
```

Scripting language (Perl/Python) support will follow

RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes[9]

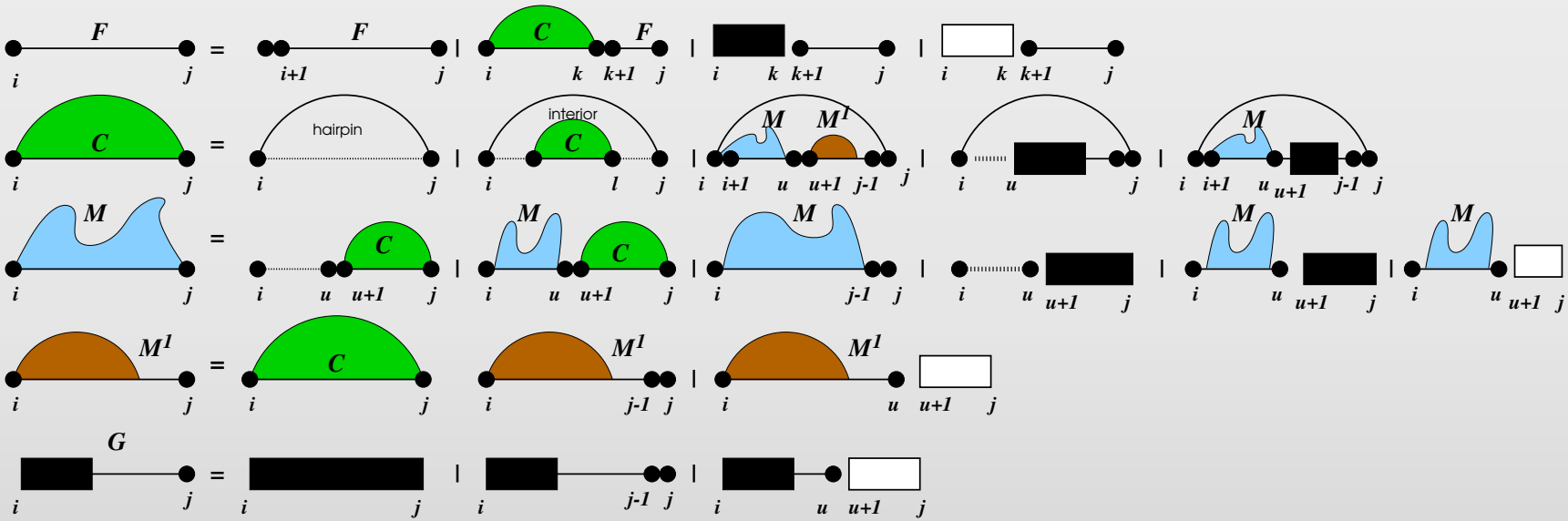




[9]Lorenz et al., (2012, 2013)

RNA Secondary structure prediction

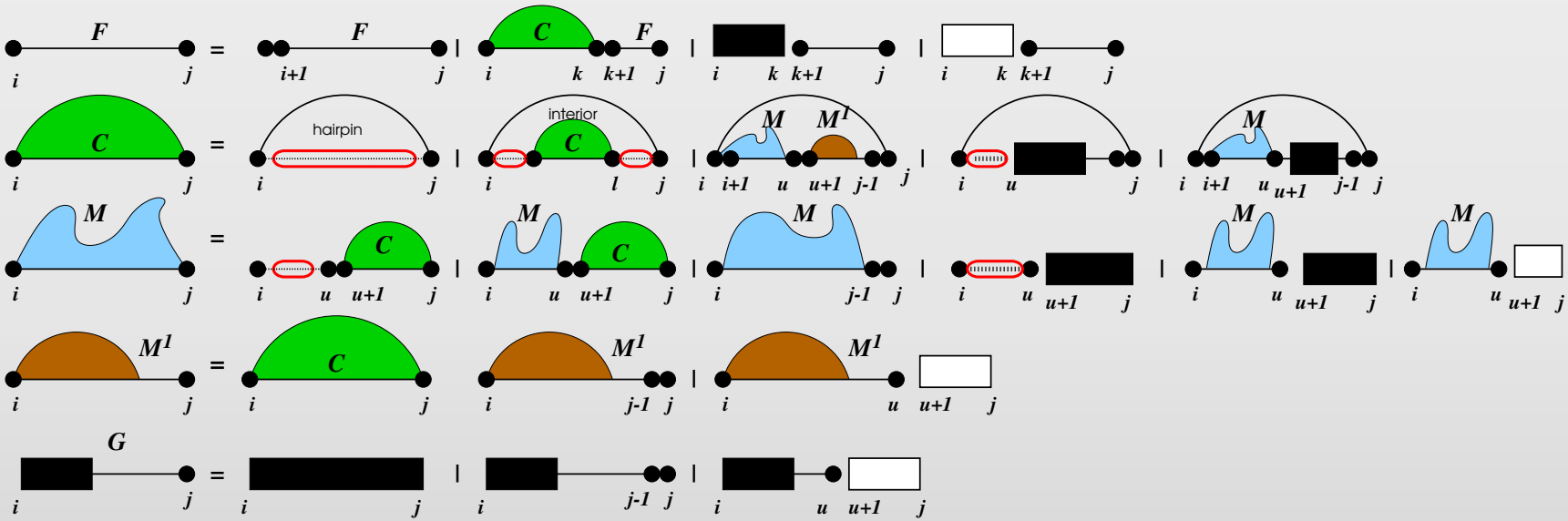
Nearest Neighbor Model with GQuadruplexes and Ligands[10]



[10] Ligands that bind to a set of consecutive unpaired nucleotides

RNA Secondary structure prediction

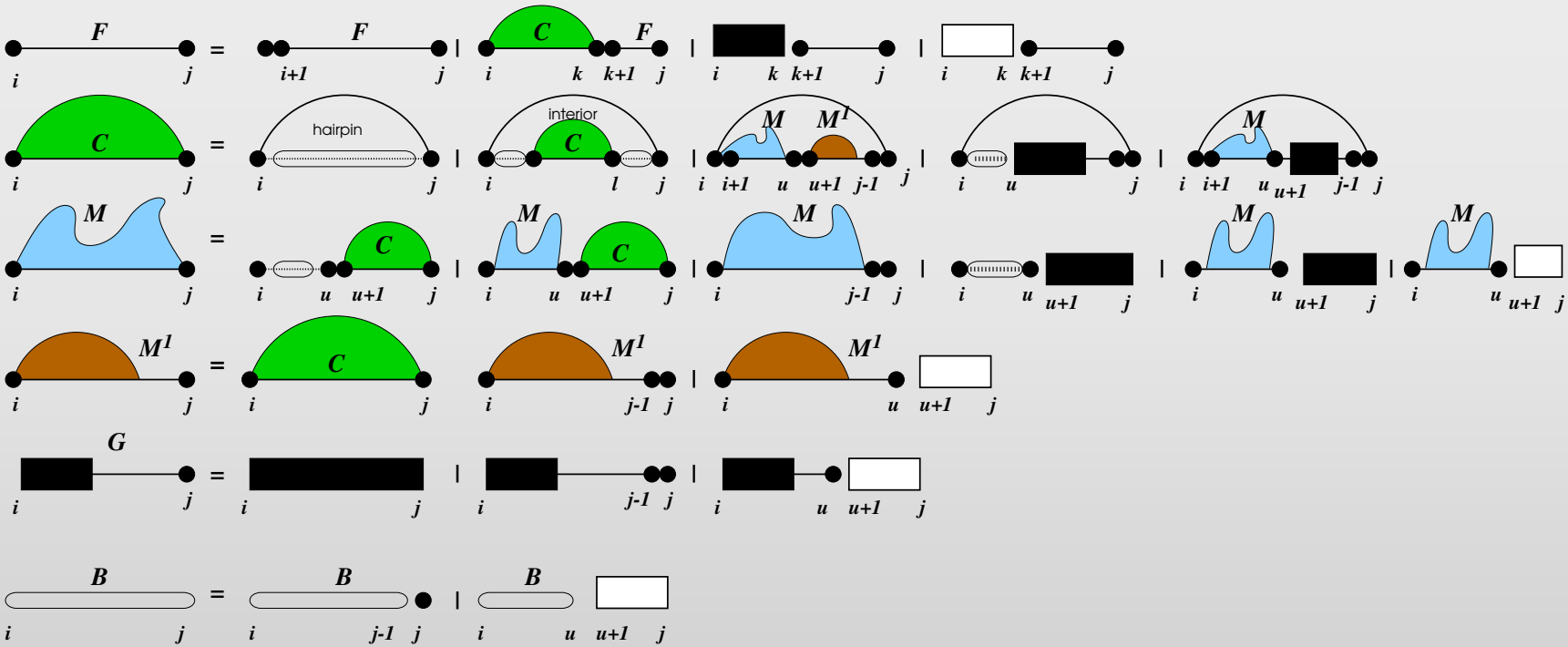
Nearest Neighbor Model with GQuadruplexes and Ligands[10]



[10] Ligands that bind to a set of consecutive unpaired nucleotides

RNA Secondary structure prediction

Nearest Neighbor Model with GQuadruplexes and Ligands[10]



[10] Ligands that bind to a set of consecutive unpaired nucleotides

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:

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