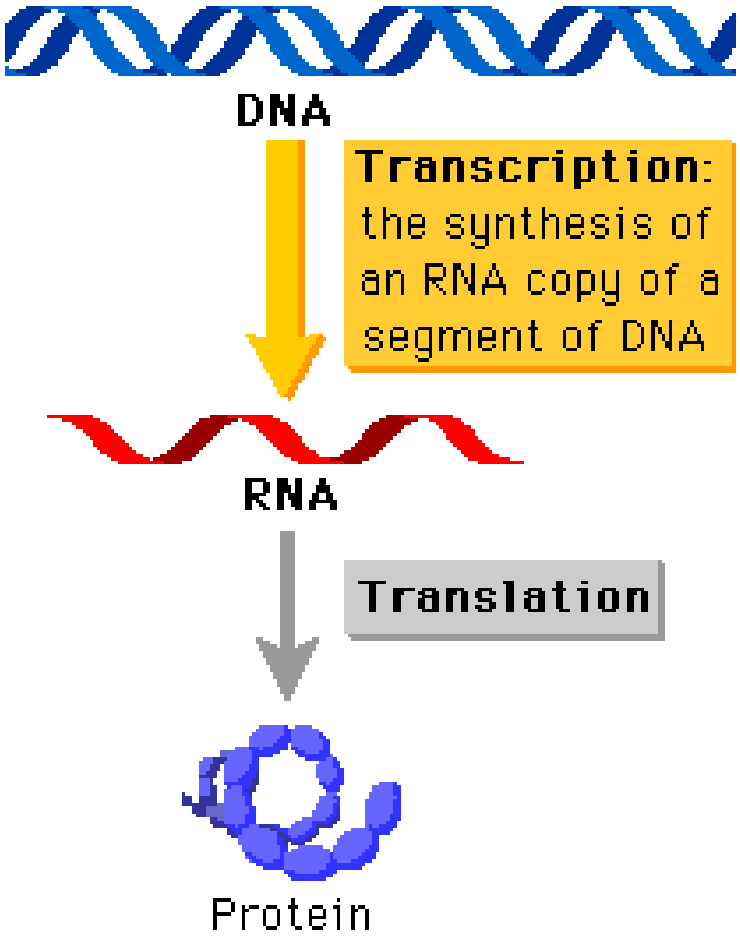
Constraint Programming and Biology: RNA secondary structure

Agostino Dovier

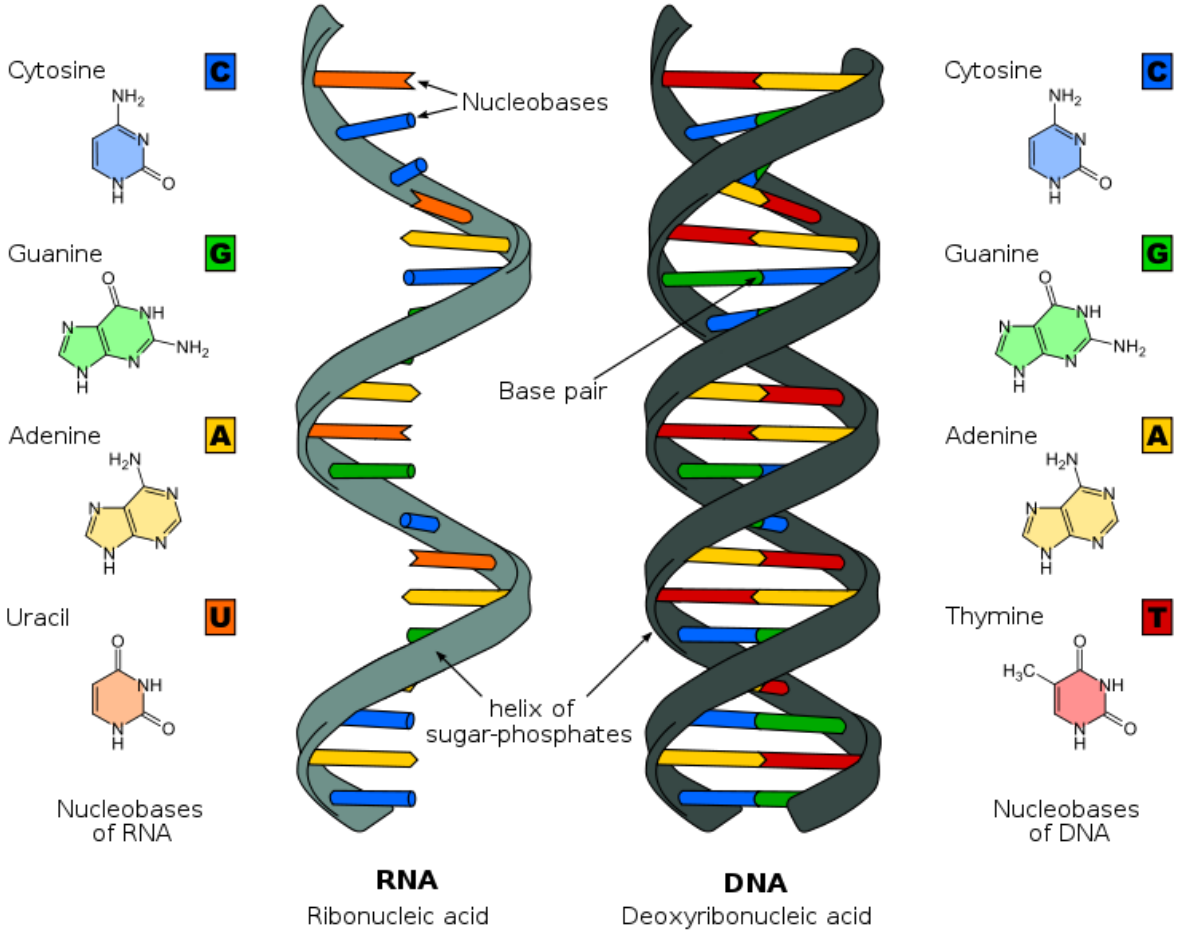
Dept. Math and Computer Science, Univ. of Udine, Italy

ACP Summer School in Constraint Programming
Wrocław, September 2012

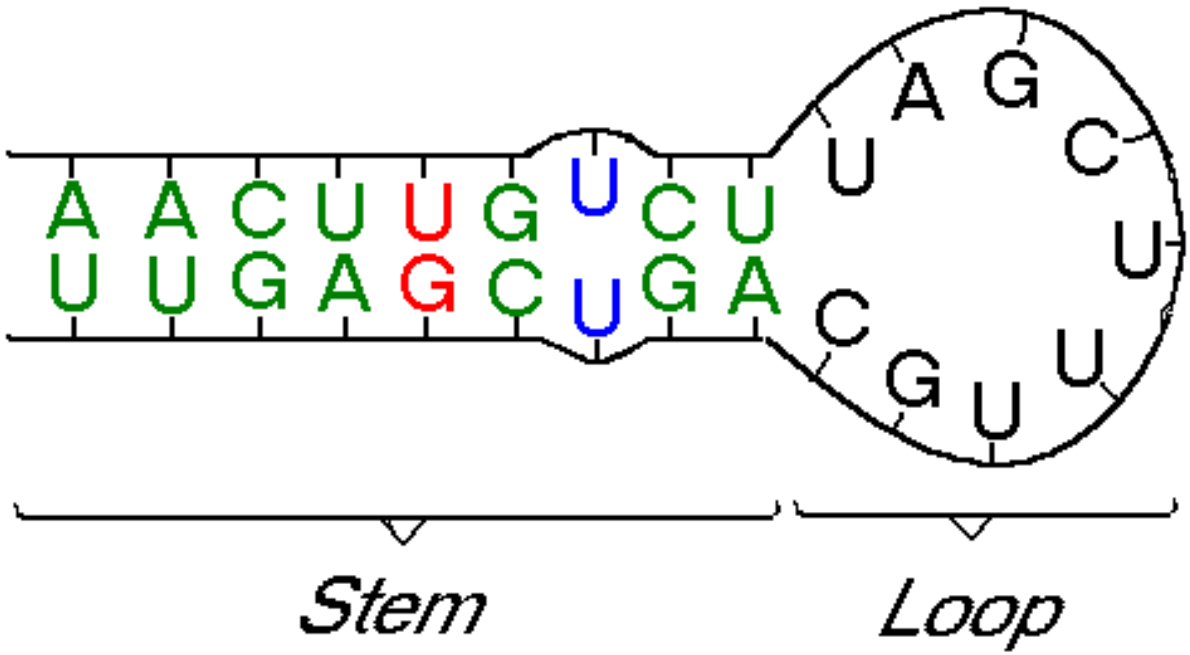
The central dogma



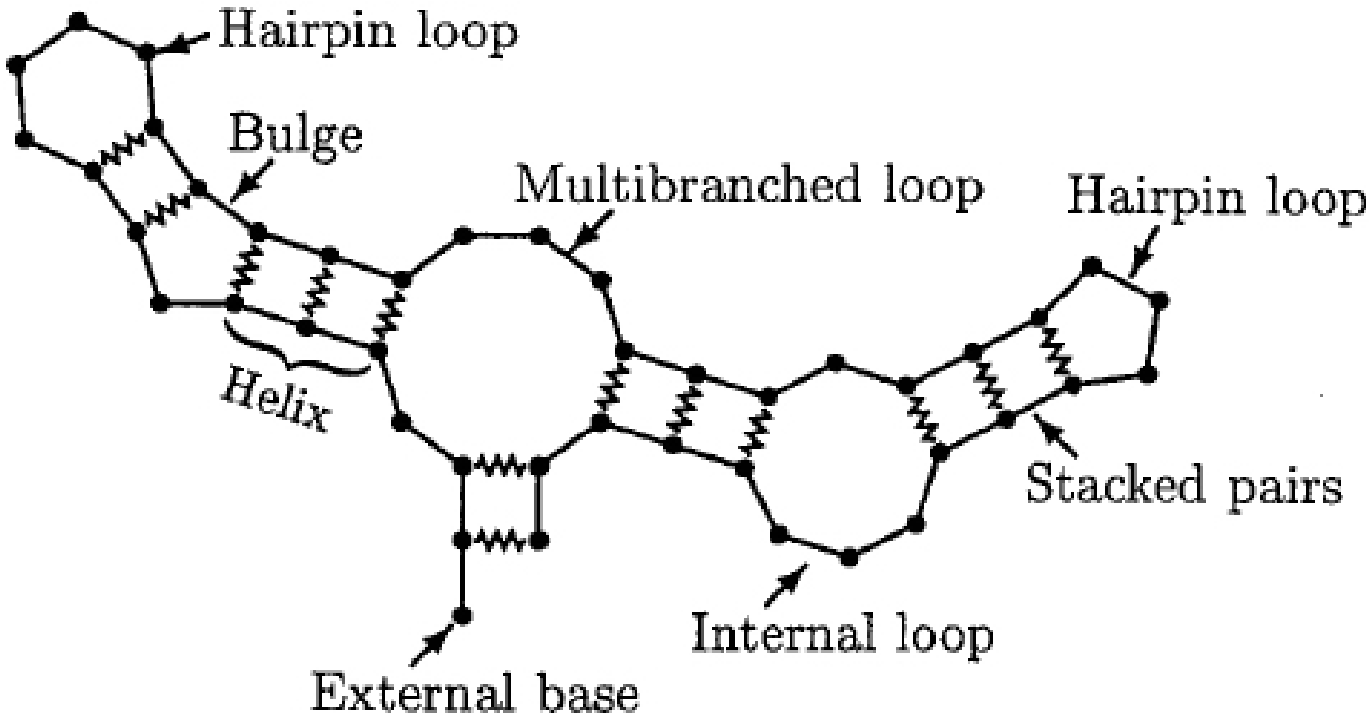
The central dogma



The central dogma



The central dogma



The central dogma

- RNA is a sequence of nucleotides (A,C,G,U) that (often) is just an intermediary between DNA and proteins
- The 3D structure of RNA depends largely on interactions between pairs of nucleotides (base pairing)
- The secondary structure is the set of its base pairings

Mathematically

- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in $\{A, C, G, U\}^*$
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that $(i, j) \in P \rightarrow i < j$

Mathematically

- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in $\{A, C, G, U\}^*$
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that $(i, j) \in P \rightarrow i < j$
  (or, alternatively, such that $(i, j) \in P \leftrightarrow (j, i) \in P$)

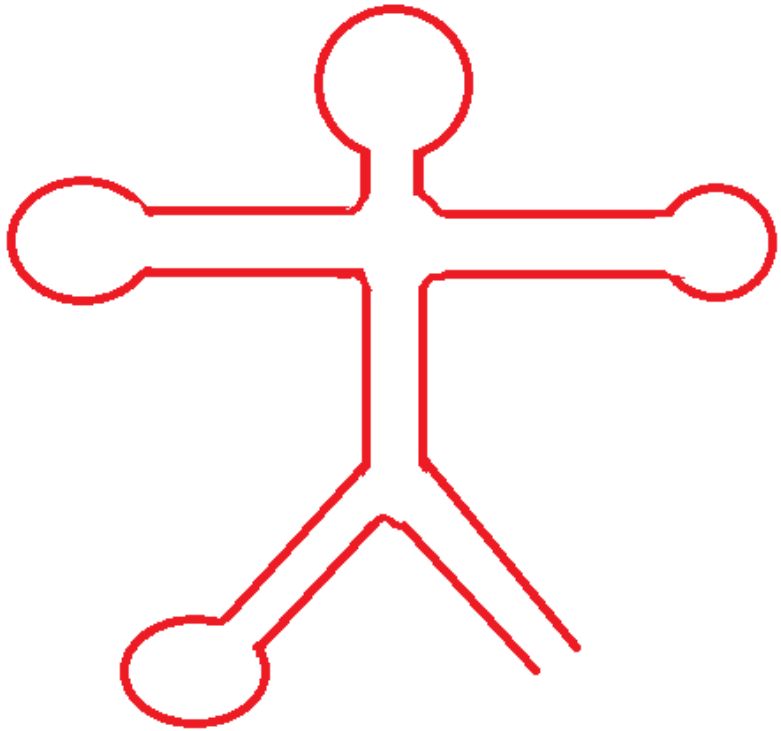
Mathematically

- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in $\{A, C, G, U\}^*$
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that $(i, j) \in P \rightarrow i < j$
  (or, alternatively, such that $(i, j) \in P \leftrightarrow (j, i) \in P$)
- One might also require from the beginning that $(i, j) \in P$ only if $(s_i, s_j) \in \{(A, U), (U, A), (C, G), (G, C), (U, G), (G, U)\}$

Mathematically

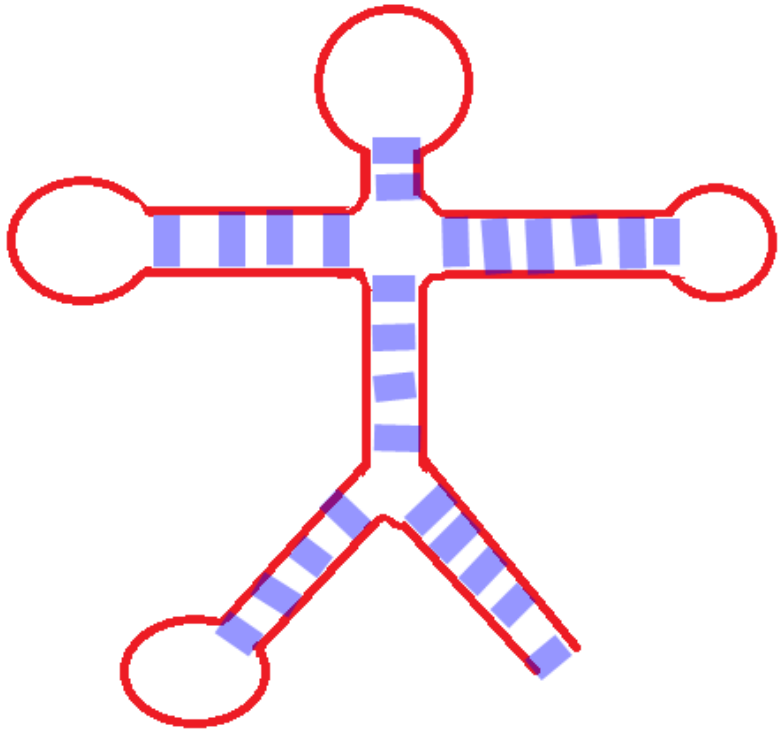
- A RNA sequence $\vec{s} = s_1 s_2 \cdots s_n$ is a string in $\{A, C, G, U\}^*$
- A RNA secondary structure is a (partial) injective function $P \subseteq \{1, \ldots, n\}^2$ such that $(i, j) \in P \rightarrow i < j$
  (or, alternatively, such that $(i, j) \in P \leftrightarrow (j, i) \in P$)
- One might also require from the beginning that $(i, j) \in P$ only if $(s_i, s_j) \in \{(A, U), (U, A), (C, G), (G, C), (U, G), (G, U)\}$
- We are interested in a pairing maximizing the pairings (and/or minimizing a more difficult energy function)

Spatial constraints (pseudo knot)



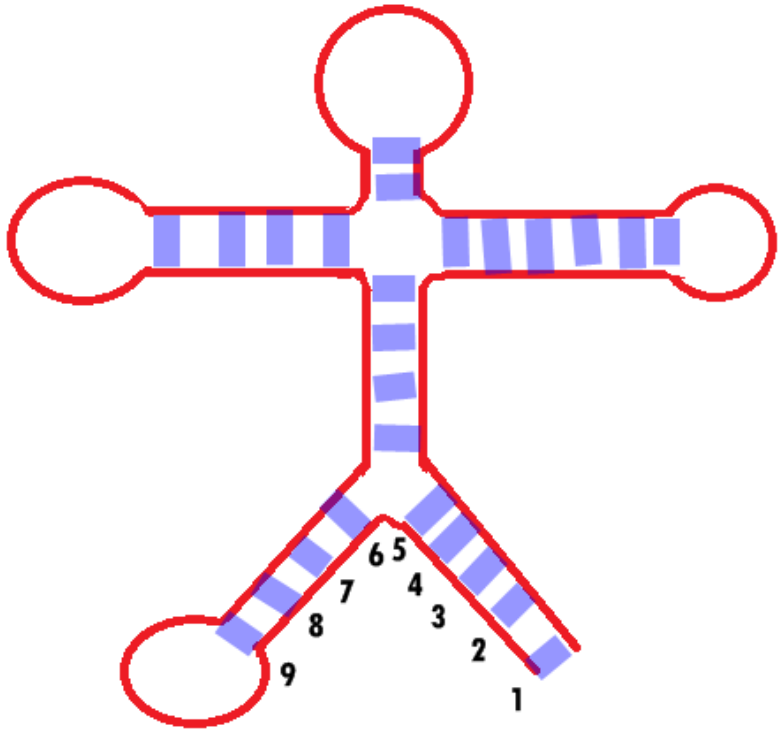
If $i < \ell < j$ and $(i,j) \in P$, and $((\ell,k) \in P$ or $(k,\ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



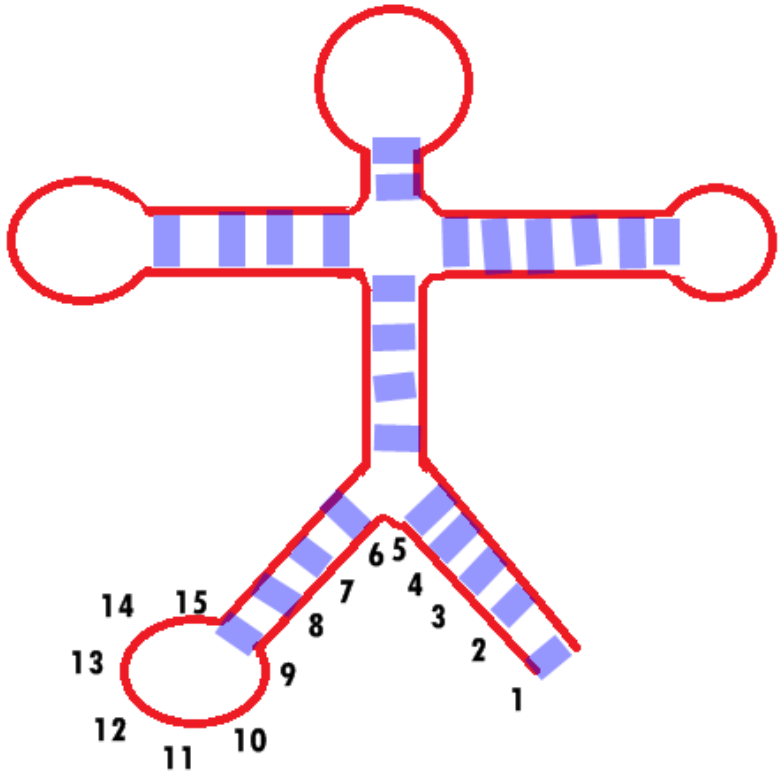
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



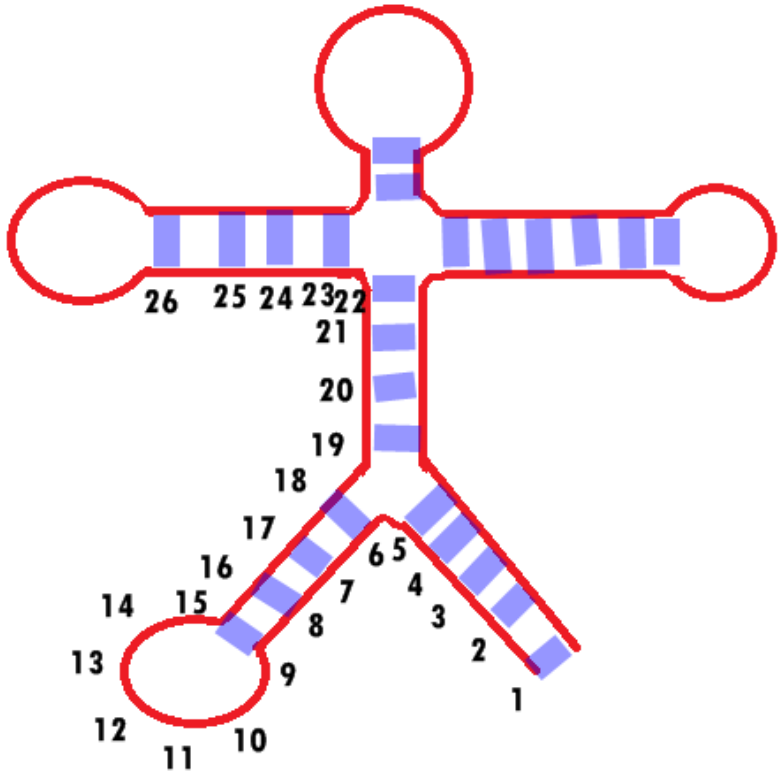
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



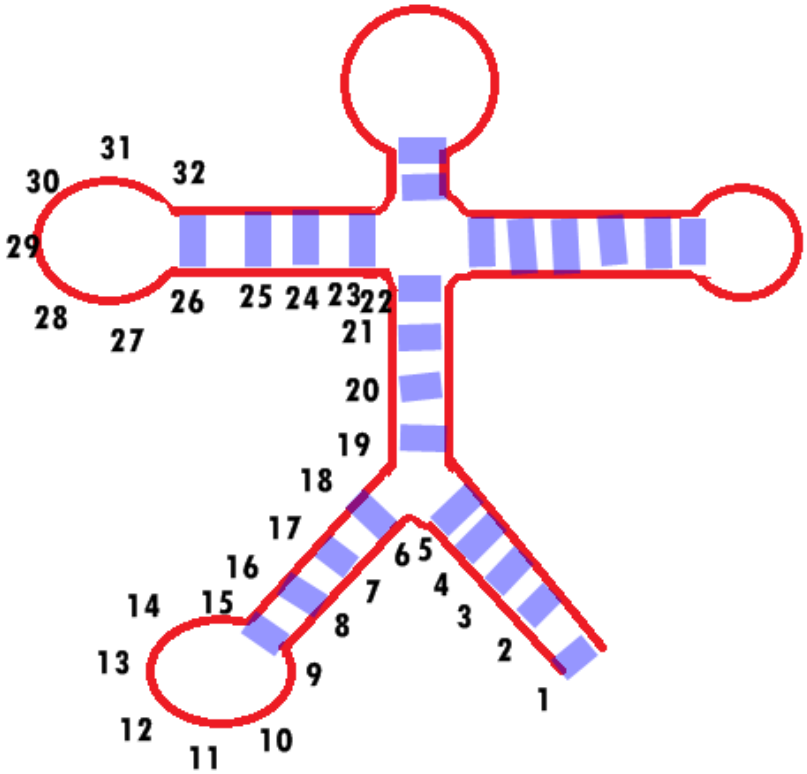
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



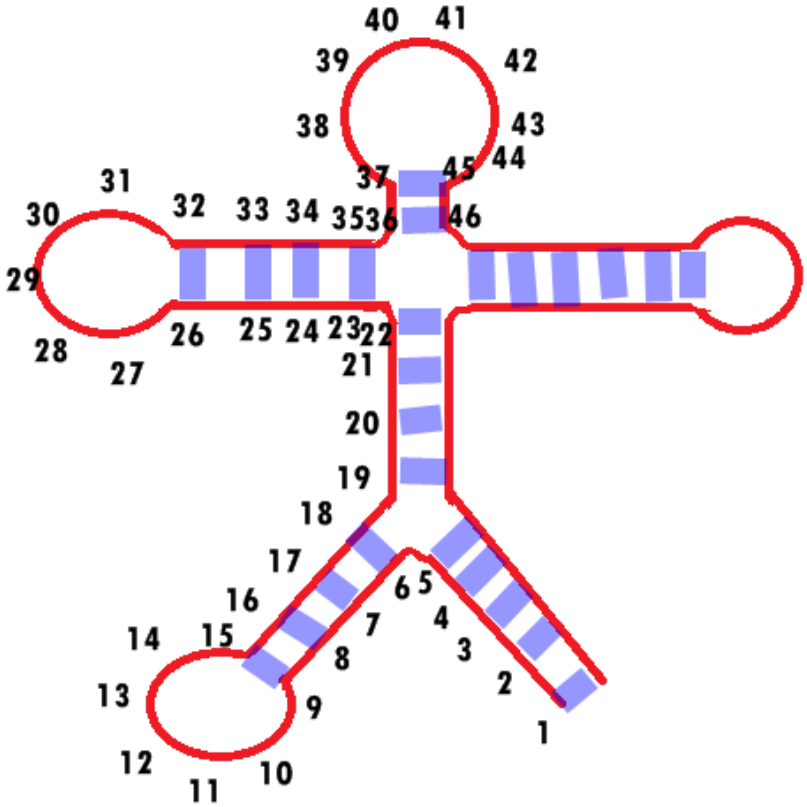
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



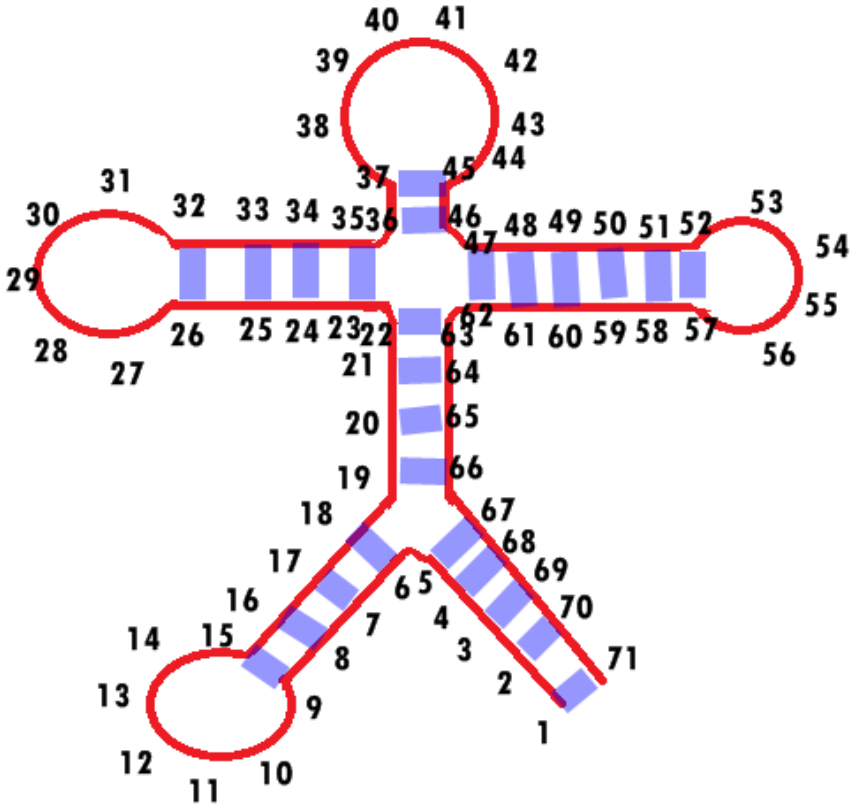
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



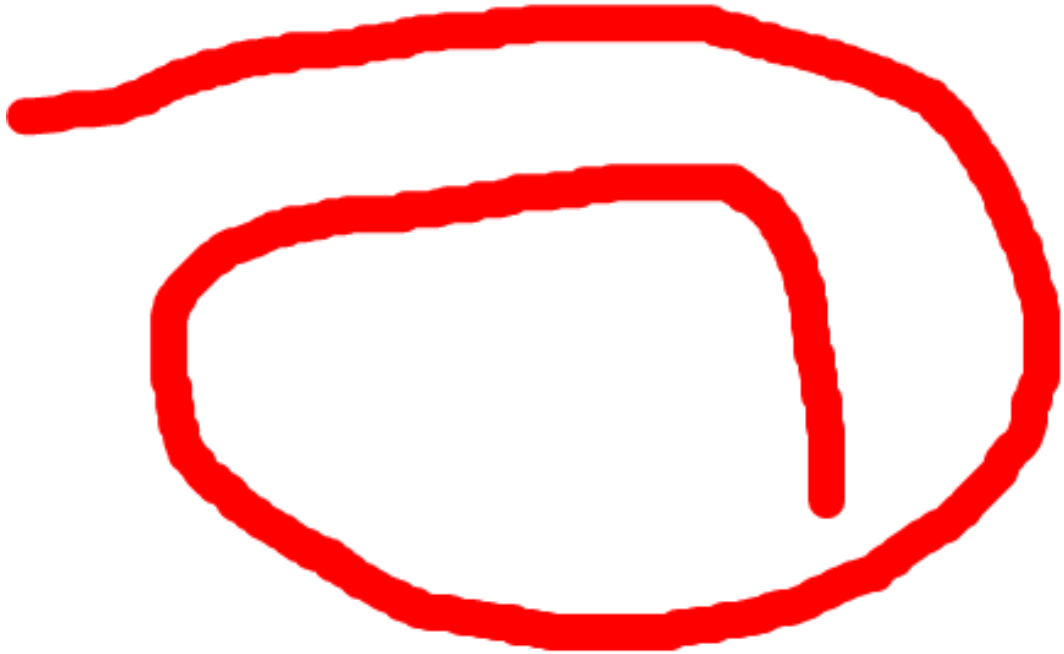
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



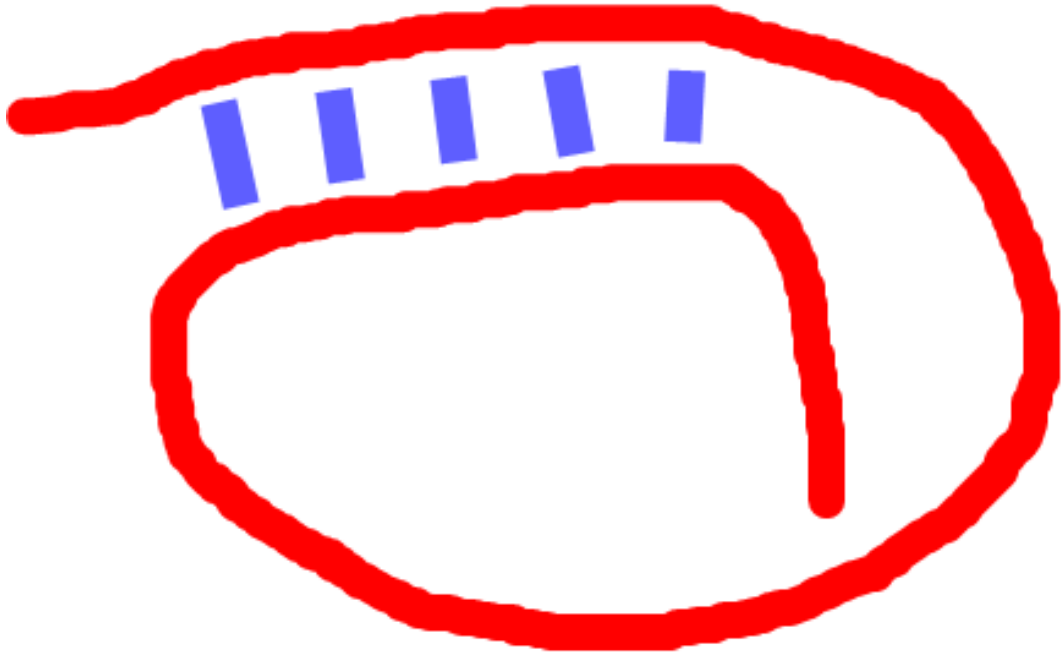
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



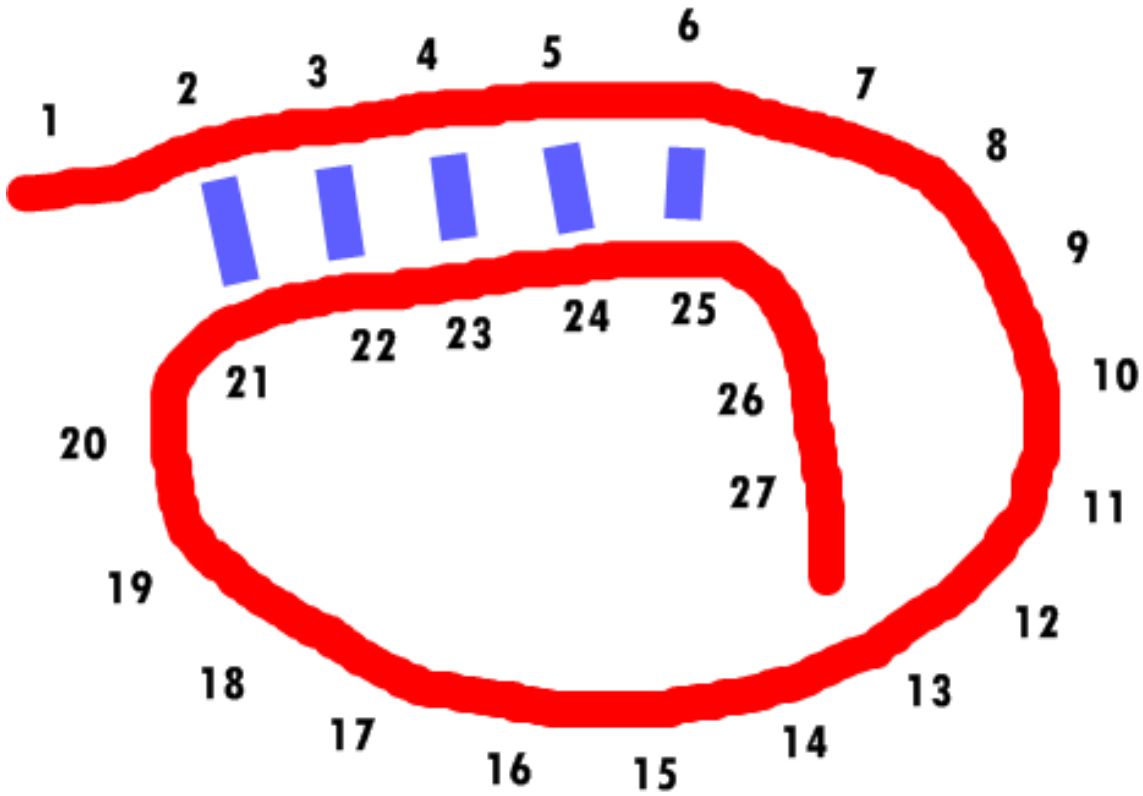
If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.

Spatial constraints (pseudo knot)



If $i < \ell < j$ and $(i, j) \in P$, and $((\ell, k) \in P$ or $(k, \ell) \in P)$, then $i < k < j$.
NO!

Results

- The pseudo-knot constraint is sensible. Adding it there are polynomial-time algorithms (`mfold`: dynamic programming. `http://mfold.rna.albany.edu`)
- Without the pseudo-knot constraint the problem is NP complete.

Results

- The pseudo-knot constraint is sensible. Adding it there are polynomial-time algorithms (`mfold`: dynamic programming. `http://mfold.rna.albany.edu`)
- Without the pseudo-knot constraint the problem is NP complete.
- Actually, what problem?

NP Completeness

Lyngsø and Pedersen, 2000

- Let $\vec{s} = s_1 \cdots s_n$ be a RNA sequence, and P a secondary structure. Then

$$E(\vec{s}, P) = \sum_{(i,j) \in P, i<j} E(\vec{s}, i, j, P)$$

- where $E(\vec{s}, i, j, P)$ depend on s_i and s_j and, moreover, on the s_z such that $(i+1, z) \in P$ or $(j-1, z) \in P$.

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$

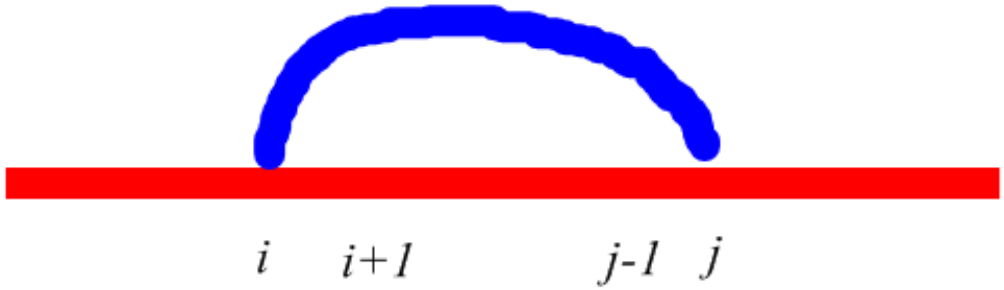
$$E(\vec{s}, i, j, P) = 0$$

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$



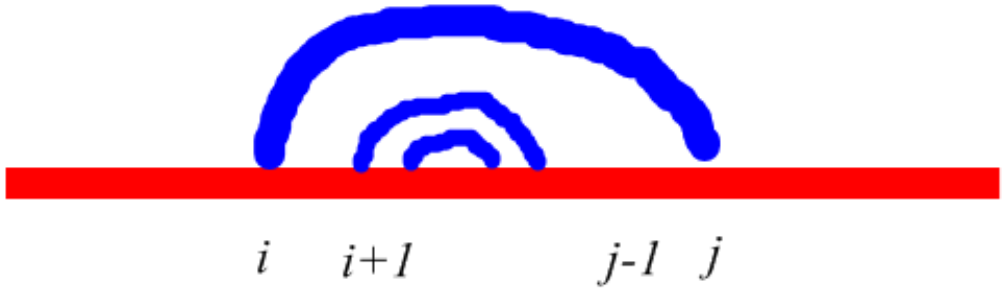
$$E(\vec{s}, i, j, P) = \text{-1}$$

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$



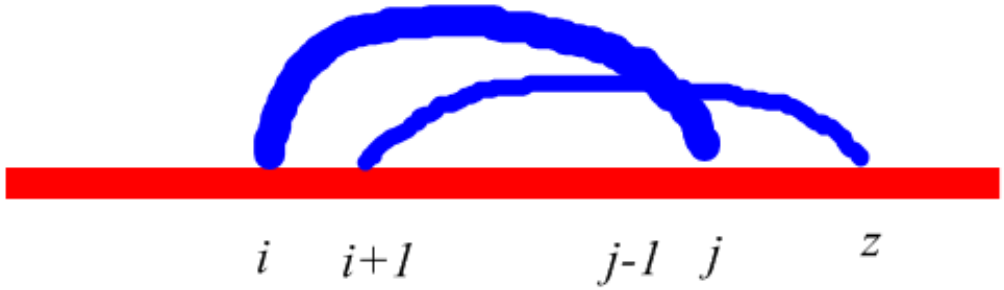
$$E(\vec{s}, i, j, P) = \text{-1}$$

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$



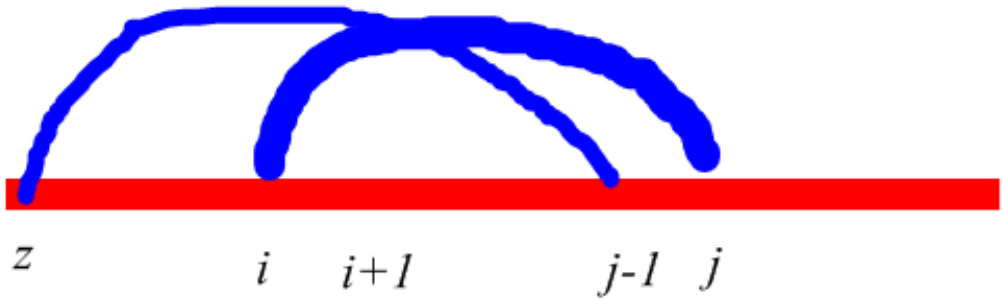
$$E(\vec{s}, i, j, P) = 0$$

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$



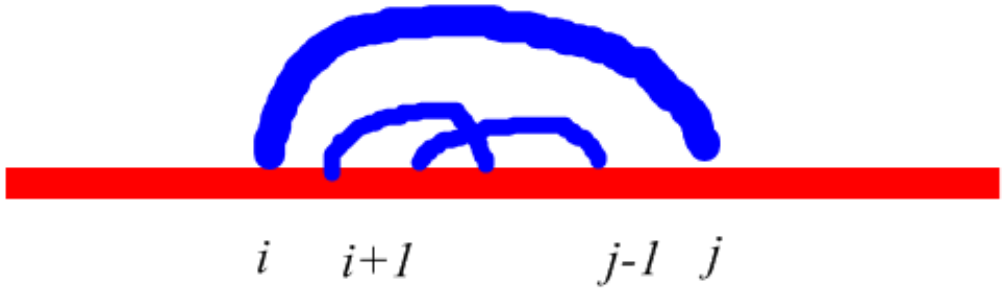
$$E(\vec{s}, i, j, P) = 0$$

NP Completeness

Lyngsø and Pedersen, 2000

- In the NP-completeness proof they first assume to have an infinite set of complementary bases (e.g., $(A_1, U_1), (A_2, U_2), (A_3, U_3), \ldots$) and define E as follows:

$$E(\vec{s}, i, j, P) = \begin{cases} -1 & \text{If } s_i \text{ and } s_j \text{ are complementary symbols and} \\ & (\forall z \in \{1, \ldots, i-1, j+1, \ldots, n\}) \\ & (\{(i+1, z), (z, i+1), (j-1, z), (z, j-1)\} \cap P = \emptyset) \\ 0 & \text{otherwise} \end{cases}$$



$$E(\vec{s}, i, j, P) = -1$$

NP Completeness

Lyngsø and Pedersen, 2000

- To prove NP hardness they start from 3SAT with the further requirement that each literal occurs at most twice (for a variable X, you can have X zero, one or two times and $\neg X$ zero, one, or two times). Prove it is NP complete (exercise)
- For a clause $c_i = \ell_1 \vee \ell_2 \vee \ell_3$ they introduce a gadget:

$$\mathcal{C}_i = c_{i,1}(\ell_1)_{1/2}\overline{c_{i,1}}\,c_{i,2}(\ell_2)_{1/2}c_{i,1}\overline{c_{i,2}}(\ell_3)_{1/2}c_{i,2}$$

  (1/2 according to the leftmost/rightmost occurrence of that literal)
- For a variable X_i that occurs twice positively and twice negatively, introduce a gadget (a substring in case of less occurrences)

$$\mathcal{V}_i = v_i\overline{(X_i)_2}\,\overline{(X_i)_1}\,\overline{v_i}\,v_i\overline{(\neg X_i)_2}\,\overline{(\neg X_i)_1}\,\overline{v_i}$$

- The encoding of a formula $c_1 \wedge \cdots \wedge c_m$ on variables $X_1 \ldots X_n$ is

$$\mathcal{C}_1 \cdots \mathcal{C}_m \mathcal{V}_1 \cdots \mathcal{V}_n$$

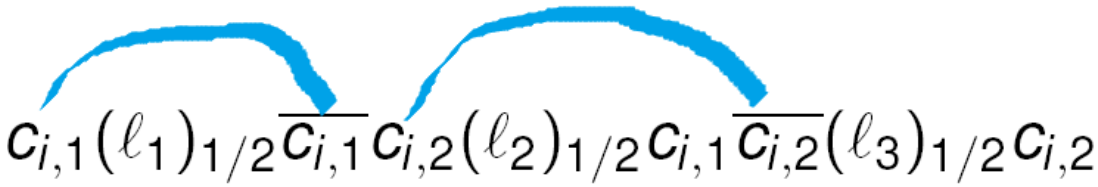
NP Completeness

Main idea

$$c_{i,1}(\ell_1)_{1/2}\overline{c_{i,1}}\,c_{i,2}(\ell_2)_{1/2}c_{i,1}\,\overline{c_{i,2}}(\ell_3)_{1/2}c_{i,2}$$

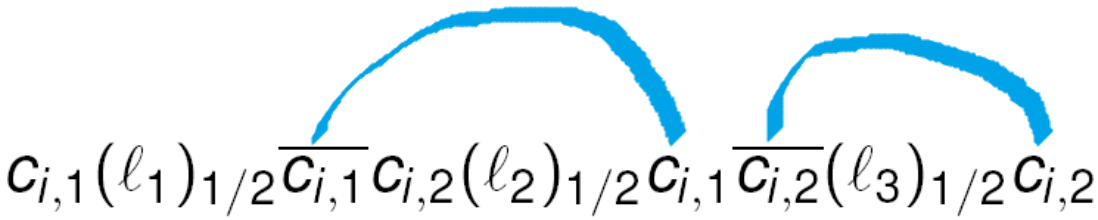
NP Completeness

Main idea



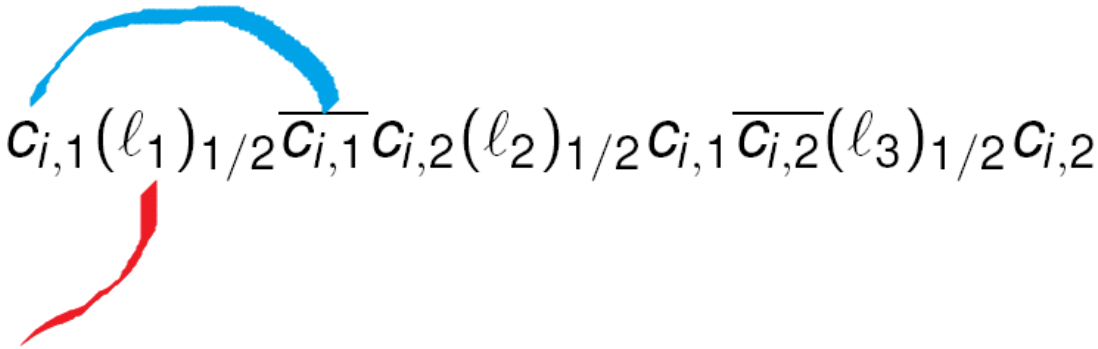
NP Completeness

Main idea



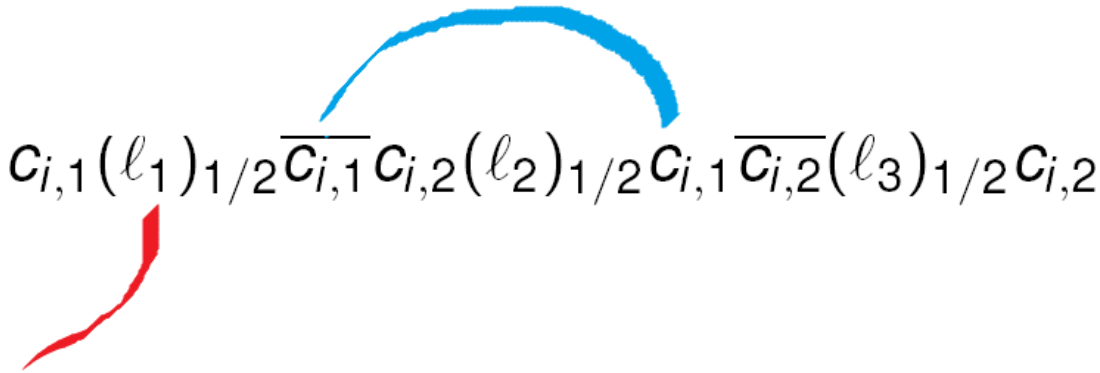
NP Completeness

Main idea



NP Completeness

Main idea



NP Completeness

Lyngsø and Pedersen, 2000

The encoding of a formula $C_1 \wedge \cdots \wedge C_m$ on variables $X_1 \ldots X_n$ is

$$\mathcal{C}_1 \cdots \mathcal{C}_m \mathcal{V}_1 \cdots \mathcal{V}_n$$

NP Completeness

Lyngsø and Pedersen, 2000

The encoding of a formula $C_1 \wedge \cdots \wedge C_m$ on variables $X_1 \ldots X_n$ is

$$\mathcal{C}_1 \cdots \mathcal{C}_m \mathcal{V}_1 \cdots \mathcal{V}_n$$

They prove that φ is satisfiable iff there is a secondary structure with energy -(3m+n) [Nice exercise]

NP Completeness

Lyngsø and Pedersen, 2000

The encoding of a formula $C_1 \wedge \cdots \wedge C_m$ on variables $X_1 \ldots X_n$ is

$$\mathcal{C}_1 \cdots \mathcal{C}_m \mathcal{V}_1 \cdots \mathcal{V}_n$$

They prove that φ is satisfiable iff there is a secondary structure with energy -(3m+n) [Nice exercise]

Then they complete the proof without the hypothesis of infinite alphabet (this is a nice reading—too long for explaining it in this course).

A simple CLP encoding

- Input $s_1, \ldots, s_n$
- Variables $Pairs = [P_1, \ldots, P_n]$.
- Let $S_x = \{i \in \{1, \ldots, n\} \mid s_i = x\}$.
  If $s_i = A$, then $\text{dom}(P_i) = \{0\} \cup S_U$.
  If $s_i = C$, then $\text{dom}(P_i) = \{0\} \cup S_G$.
  If $s_i = G$, then $\text{dom}(P_i) = \{0\} \cup S_C \cup S_U$.
  If $s_i = U$, then $\text{dom}(P_i) = \{0\} \cup S_A \cup S_G$.
- For $i = 1, \ldots, n$, if $P_i > 0$ then $P_{P_i} = I$. If $P_i = 0$ no constraint. It can be stated compactly as:

$$element(P + 1, [I|Pairs], I)$$

- Pseudo-knots: If $P_i > 0$ then $(P_{i+1} \in [i + 3..P_{P_i} - 1]) \vee (P_{i+1} = 0)$

A simple CLP encoding

- As cost function we want either to maximize contacts or

A simple CLP encoding

- As cost function we want either to maximize contacts or (as done by Dahl-Bavarian, WCB05),

A simple CLP encoding

- As cost function we want either to maximize contacts or (as done by Dahl-Bavarian, WCB05),
- a solution close to the statistics, namely 35% for AU, 53% for CG, 12% for GU.
- Let $NC = n - \#\text{contacts}$
- We minimize therefore a weighted sum of the form

$$c_1 \frac{NC}{n} + c_2 \frac{\#(AU) - .35(n - NC)}{n} + c_3 \frac{\#(CG) - .53(n - NC)}{n}$$

(c_1, c_2, c_3 constants that can be changed. The denominator n can be omitted for minimization)

- Let us see some execution of `RNA_alignment.pl`

(Some) References

- M. Zucker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleid Acid Research, 9(1):133–148, 2981.
- R.B. Lyngsø and C.N.S Pedersen. RNA Pseudoknot prediction in Energy-Based Models. J. of Computational Biology 7(3/4), 2000.
- G. Blin, G. Fertin, I. Rusu, and C. Sinoquet. Extending the hardness of RNA secondary structure comparison. LNCS 4614, pp. 140–151, 2007.
- M. Bauer, G.W. Klau, and K. Reinert. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. BMC Bioinformatics, 8, 2007.
- M. Bavarian and V. Dahl. Constraint Based Methods for Biological Sequence Analysis. J. Universal Computer Science 12(11):1500–1520, 2006 (also in WCB 05).
- A. Dal Palù, M. Möhl, S. Will. A Propagator for Maximum Weight String Alignment with Arbitrary Pairwise Dependencies. CP 2010: 167-175 (also in WCB 10)