Parameterized-complexity algorithms for the RNA sequence/structure alignment problem

P. Rinaudo[⋆ † ‡] Y. Ponty[⋆ • ○] D. Barth[†] A. Denise[⋆ ‡]

[⋆] AMIB project, INRIA Saclay, France

[†] LRI, Université Paris-Sud, France

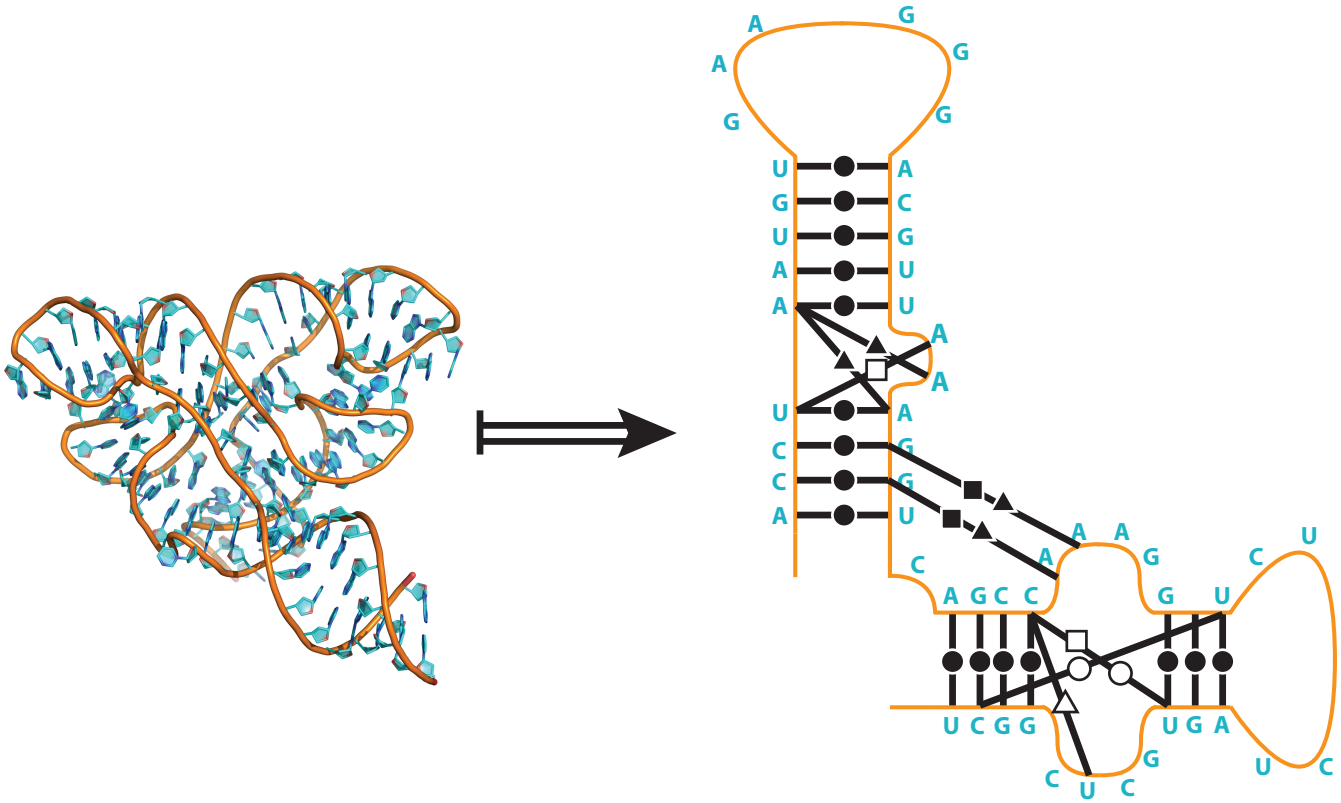
[‡] PRISM, Université Versailles St-Quentin, France

[○] PIMS, Simon Fraser University

[•] LIX/CNRS, Ecole Polytechnique, France

Discrete Maths Seminar@SFU

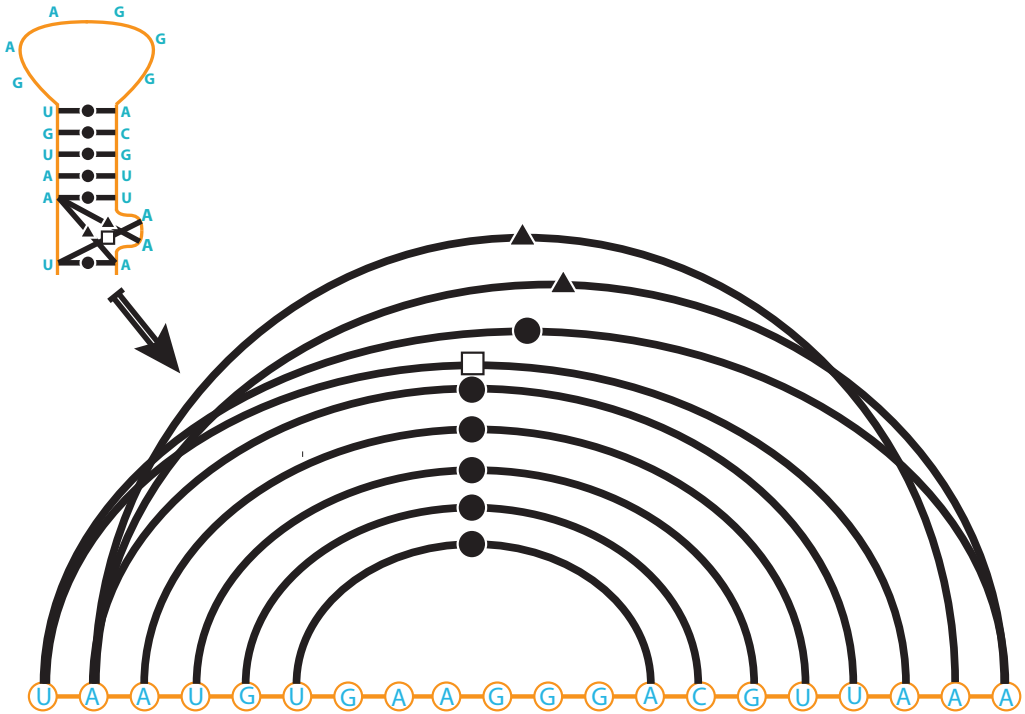
RNA structure: From 3D to 2D



- RNA = Sequence of nucleotides $\{A, C, G, U\}$
- Interactions = Pairs (Canonical/Non-canonical) of nucleotides
- Structure = Set of base-pairs $\approx$ Function (conserved)

Arc-annotated sequences

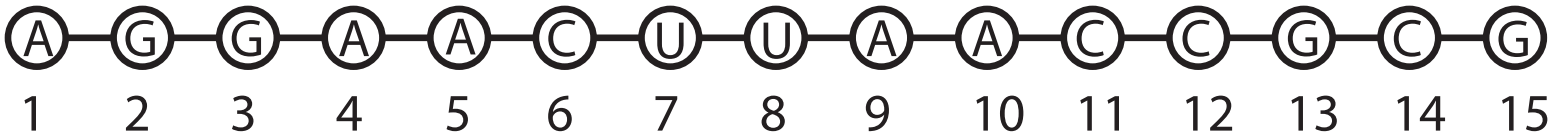
Arc-annotated sequence = Sequence + Interactions



Context: Multiple Structural levels

Primary Structure

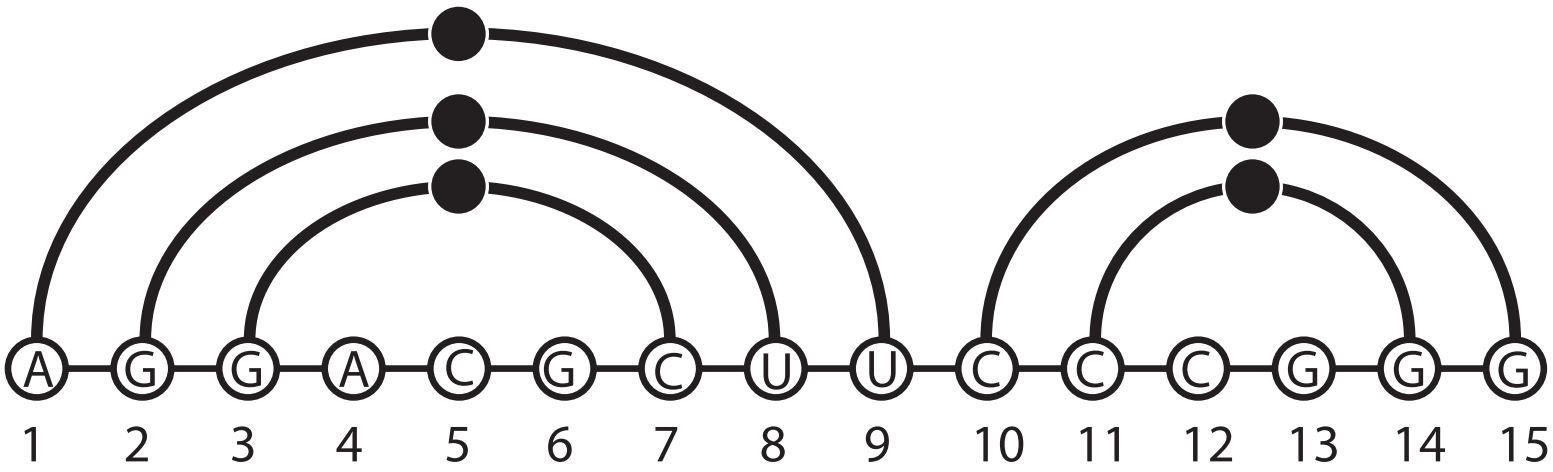
- Represents nucleotides sequence
- No interaction — Boring...



Secondary Structure

- Scaffold/blueprint for 3D
- Only includes non-crossing canonical interactions (WC/WC cis, GC/AU/GU)
- Any nucleotide has ≤ 1 partner

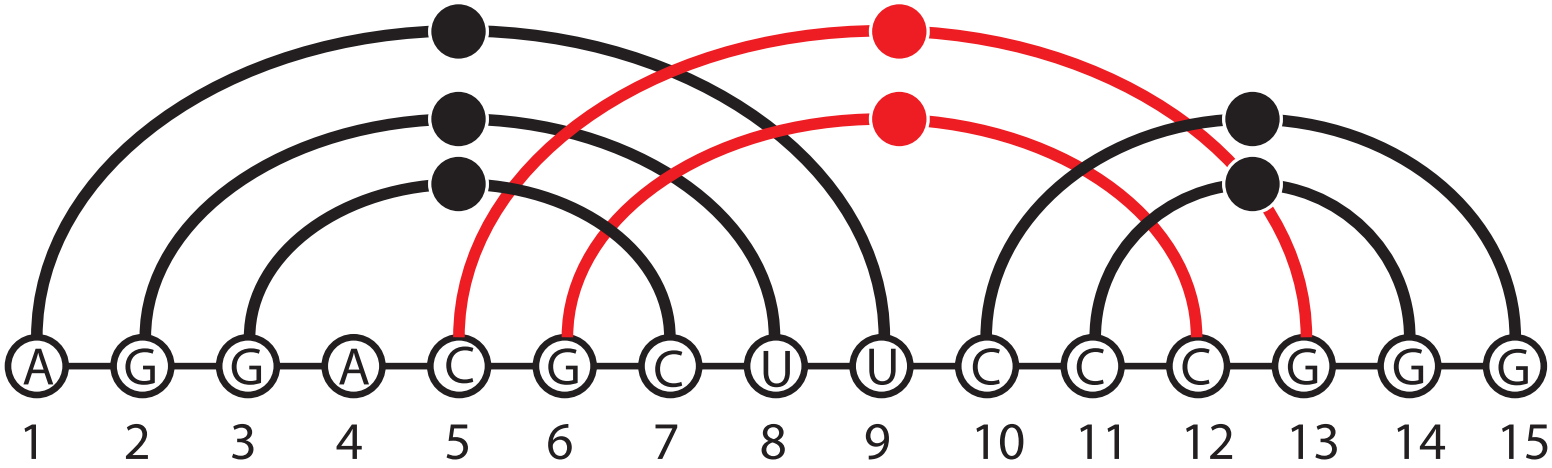
Better...



Secondary Structure with Pseudoknots

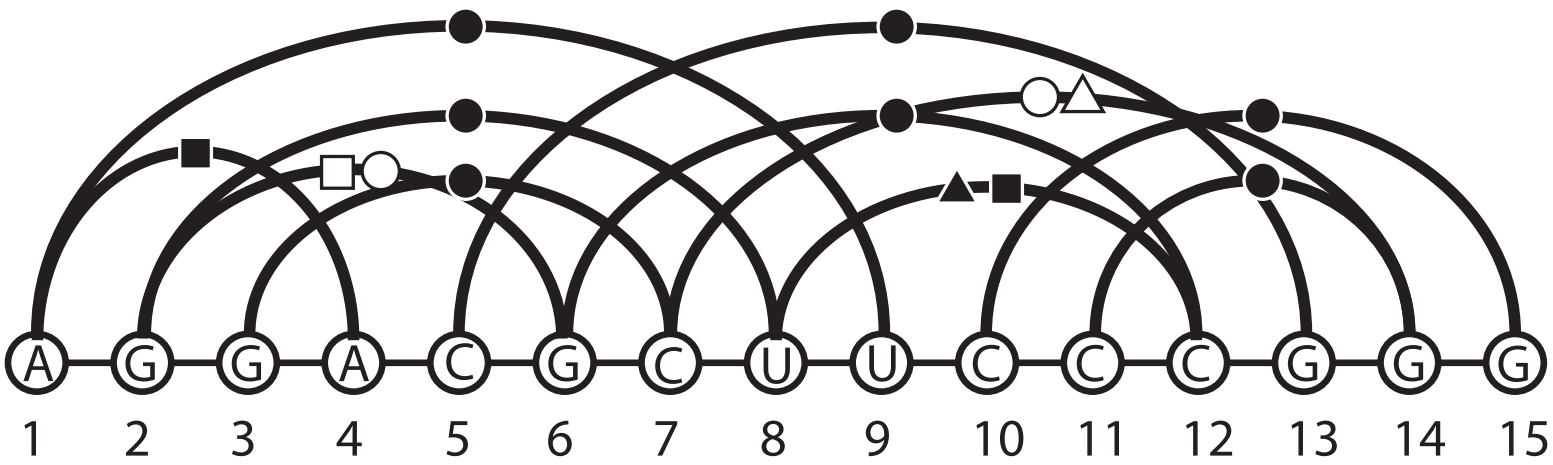
- Includes all canonical crossing interactions
- Any nucleotide has ≤ 1 partner Wow...

Pseudoknots play a major part in the architecture of some RNAs
Yet they are hard to handle algorithmically!

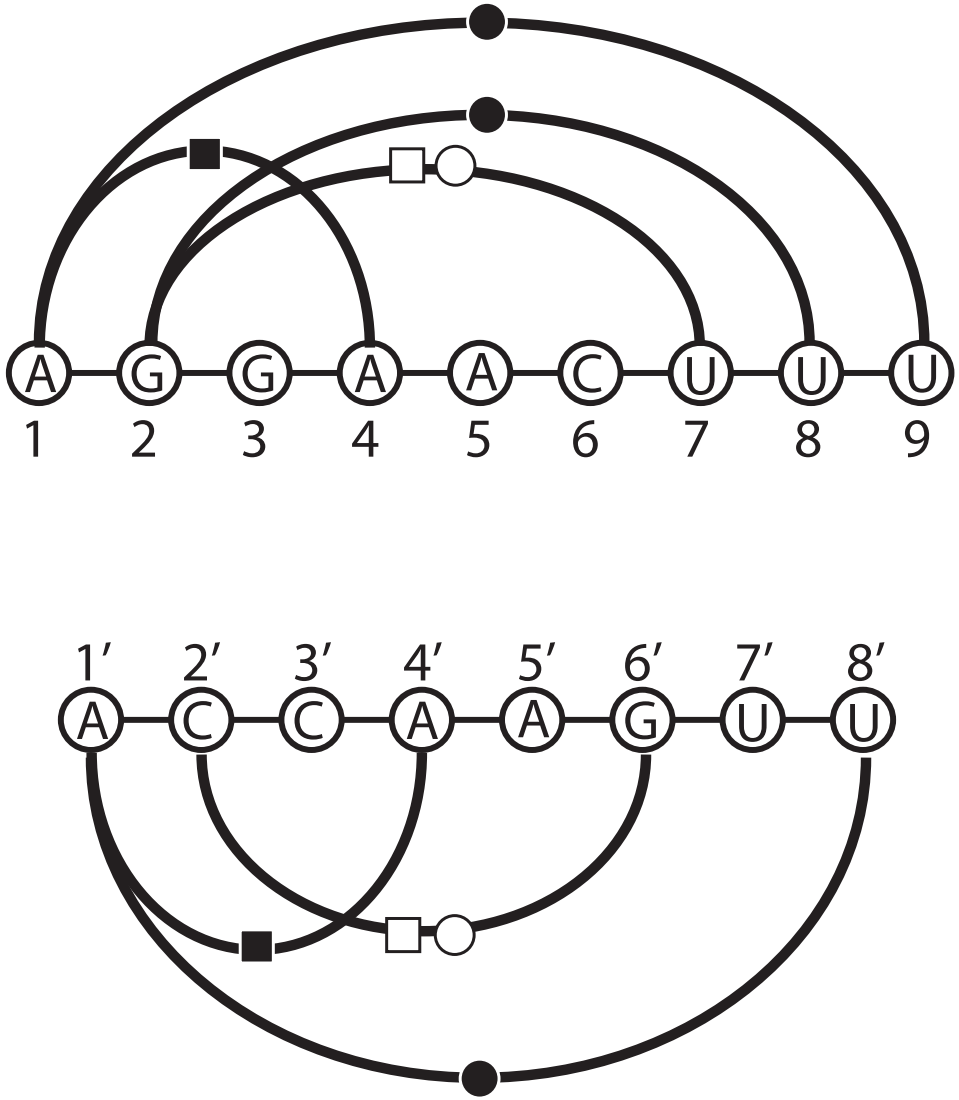


Extended secondary structure

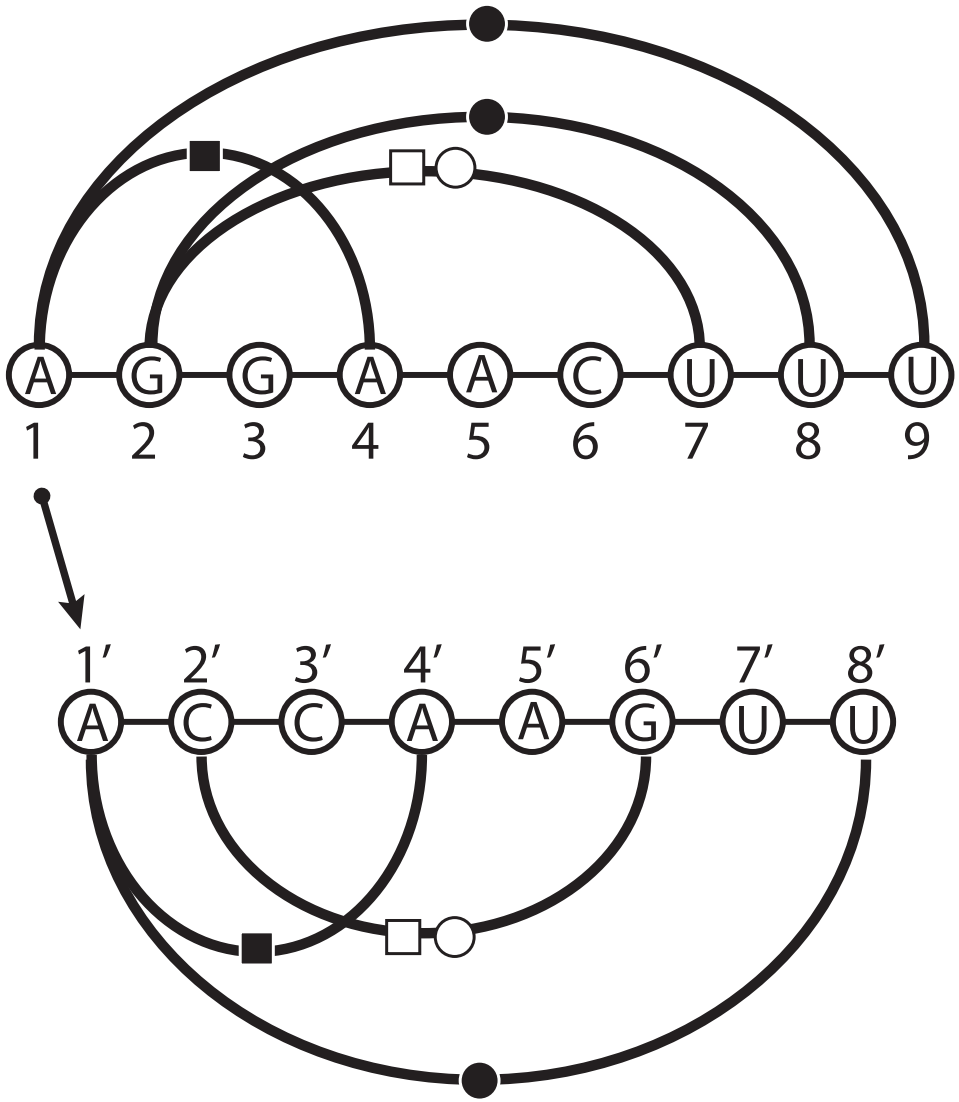
- Captures any interaction (canonical and non-canonical)
- Possibly, multiple partners per position — Now we're talking!



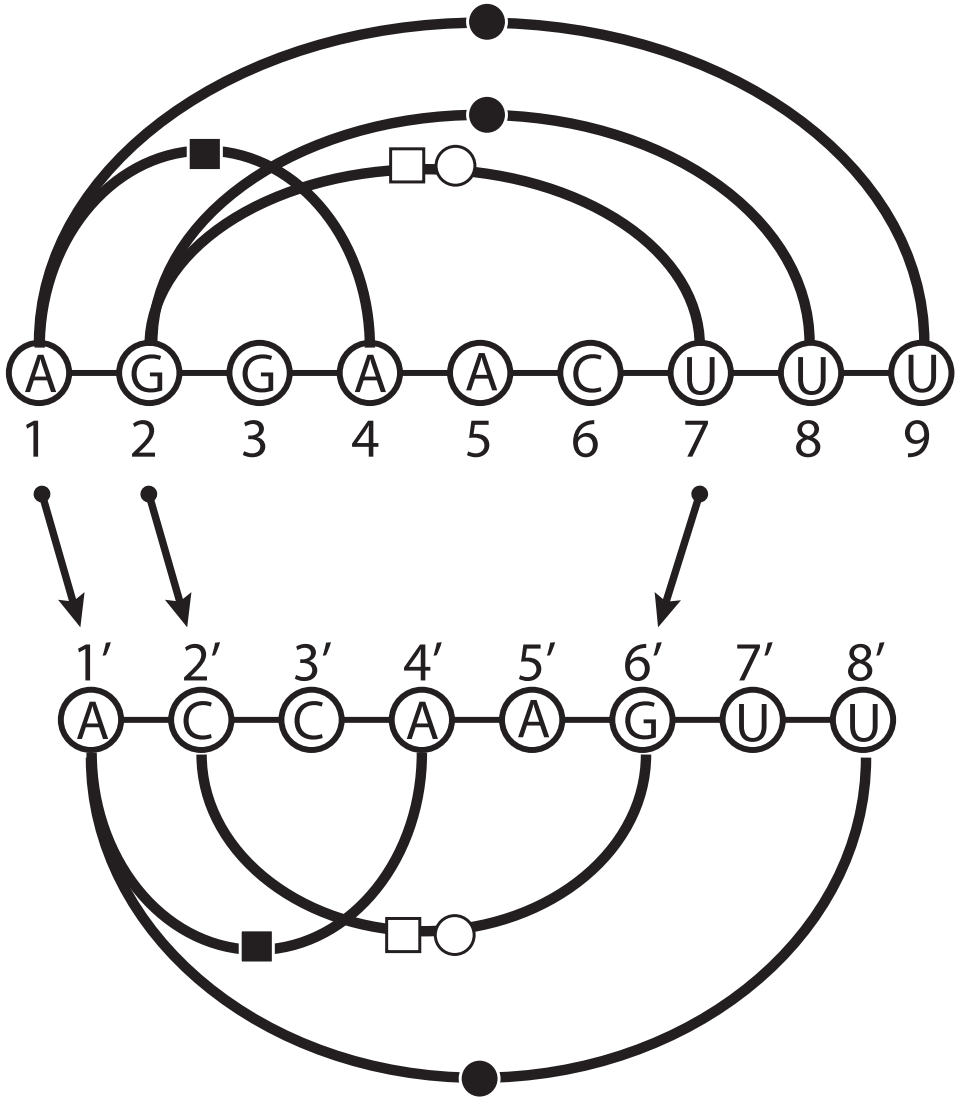
Sequence-structure alignment



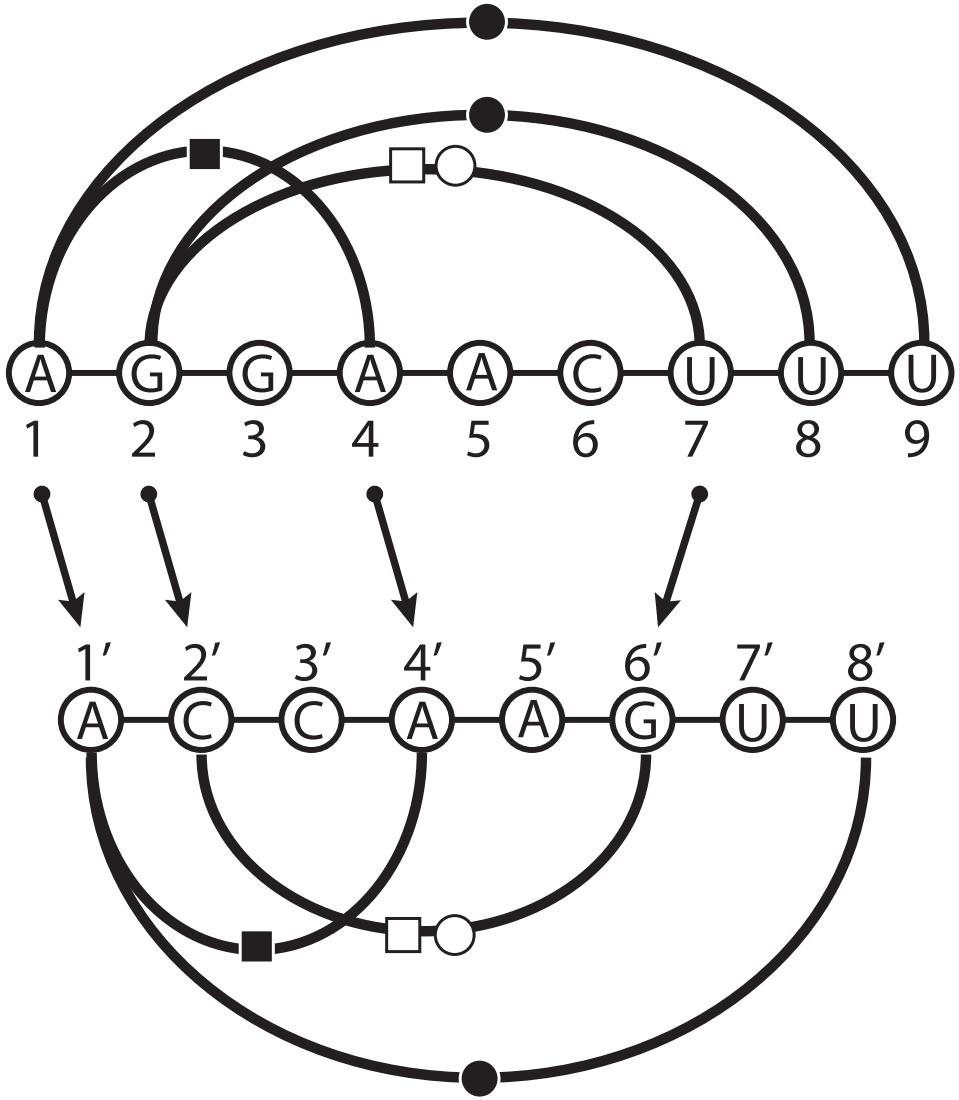
Sequence-structure alignment



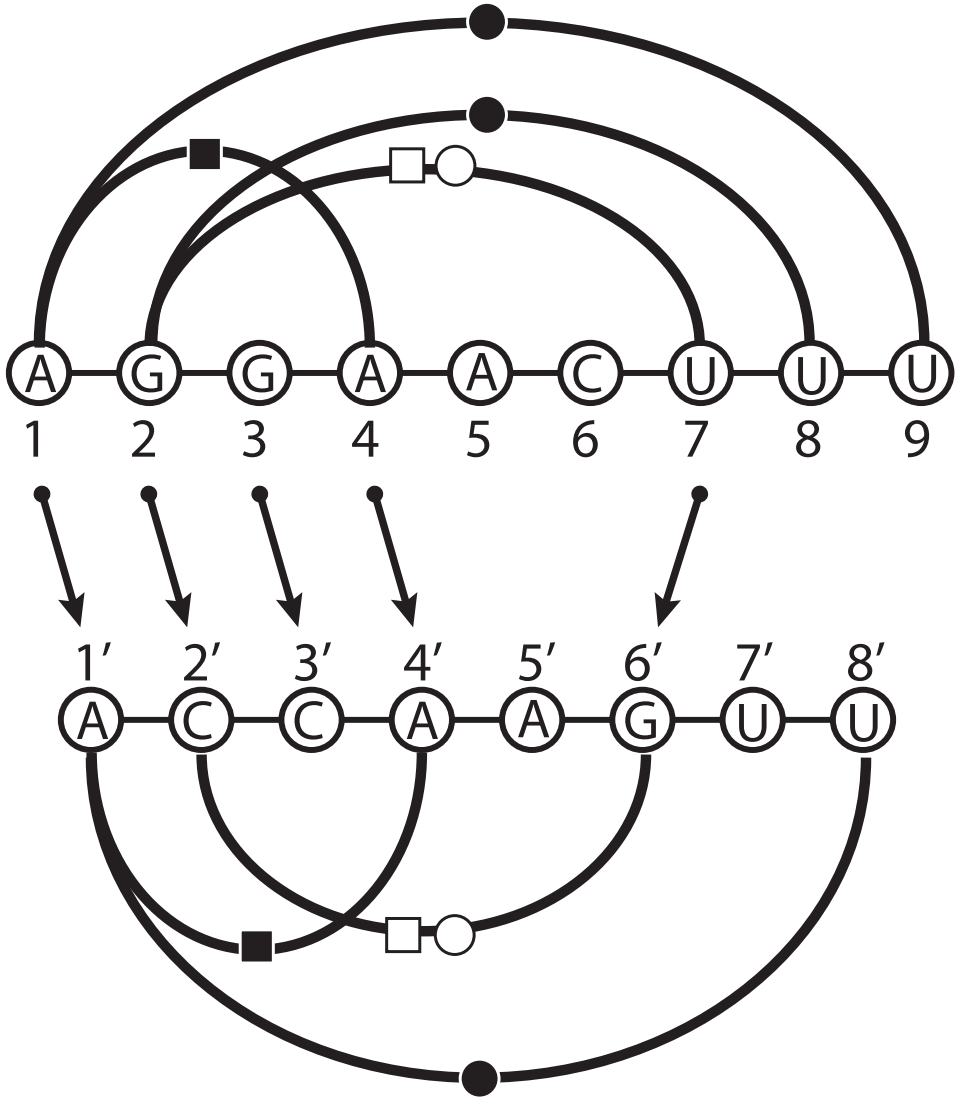
Sequence-structure alignment



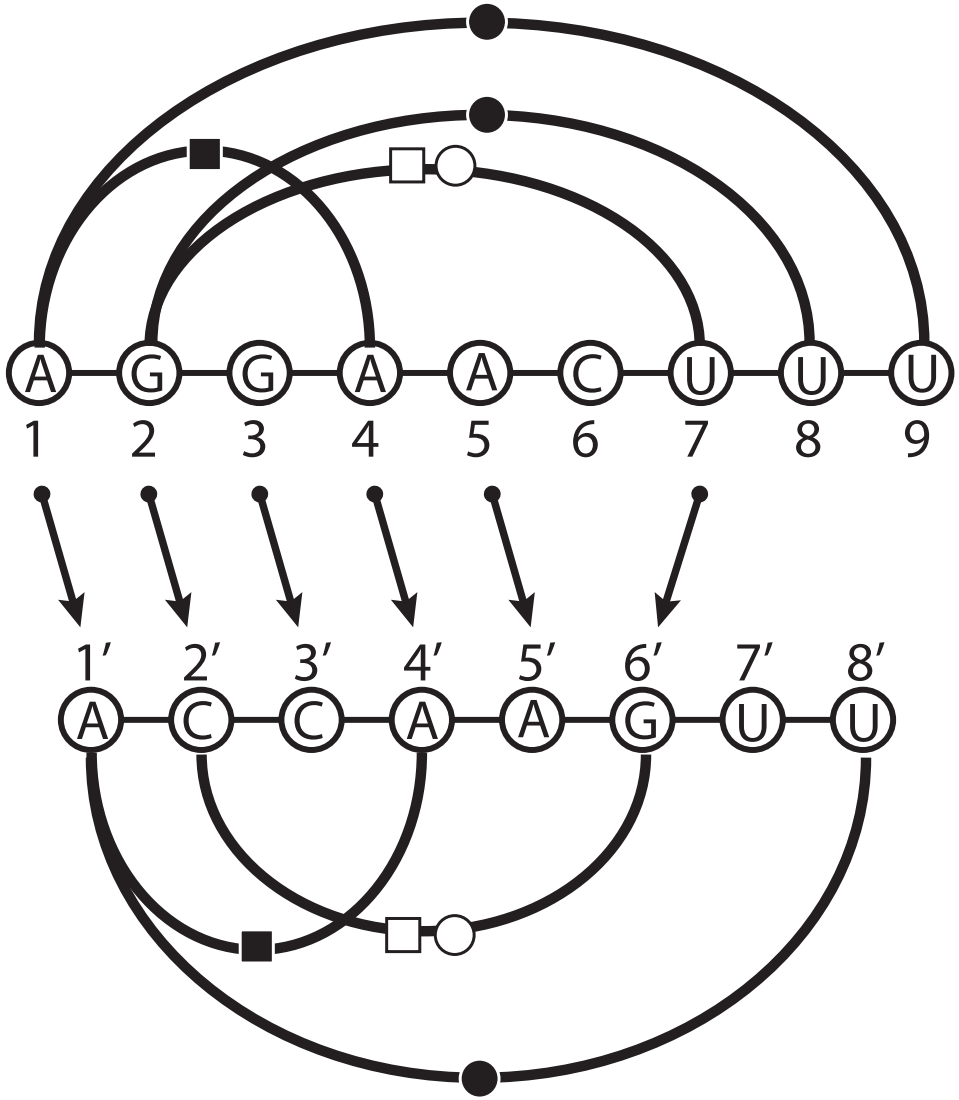
Sequence-structure alignment



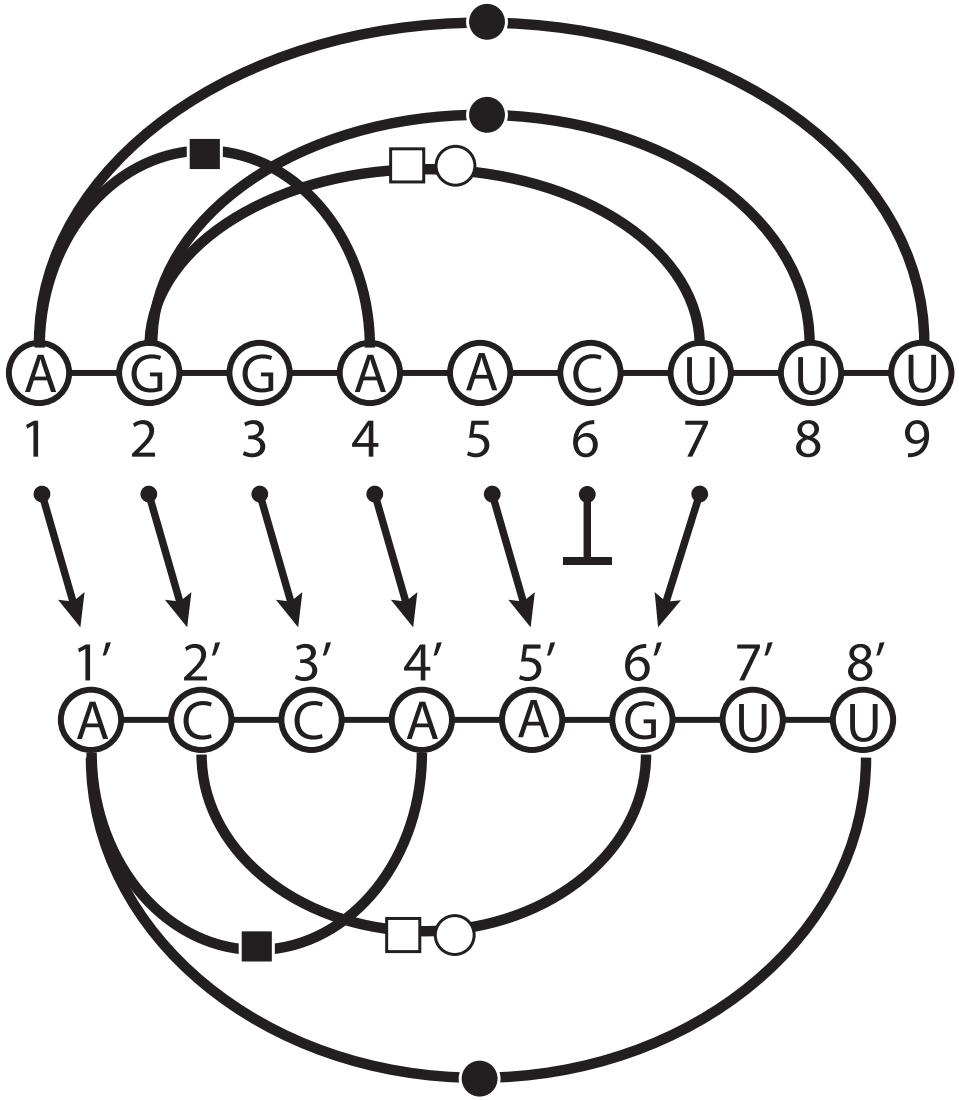
Sequence-structure alignment



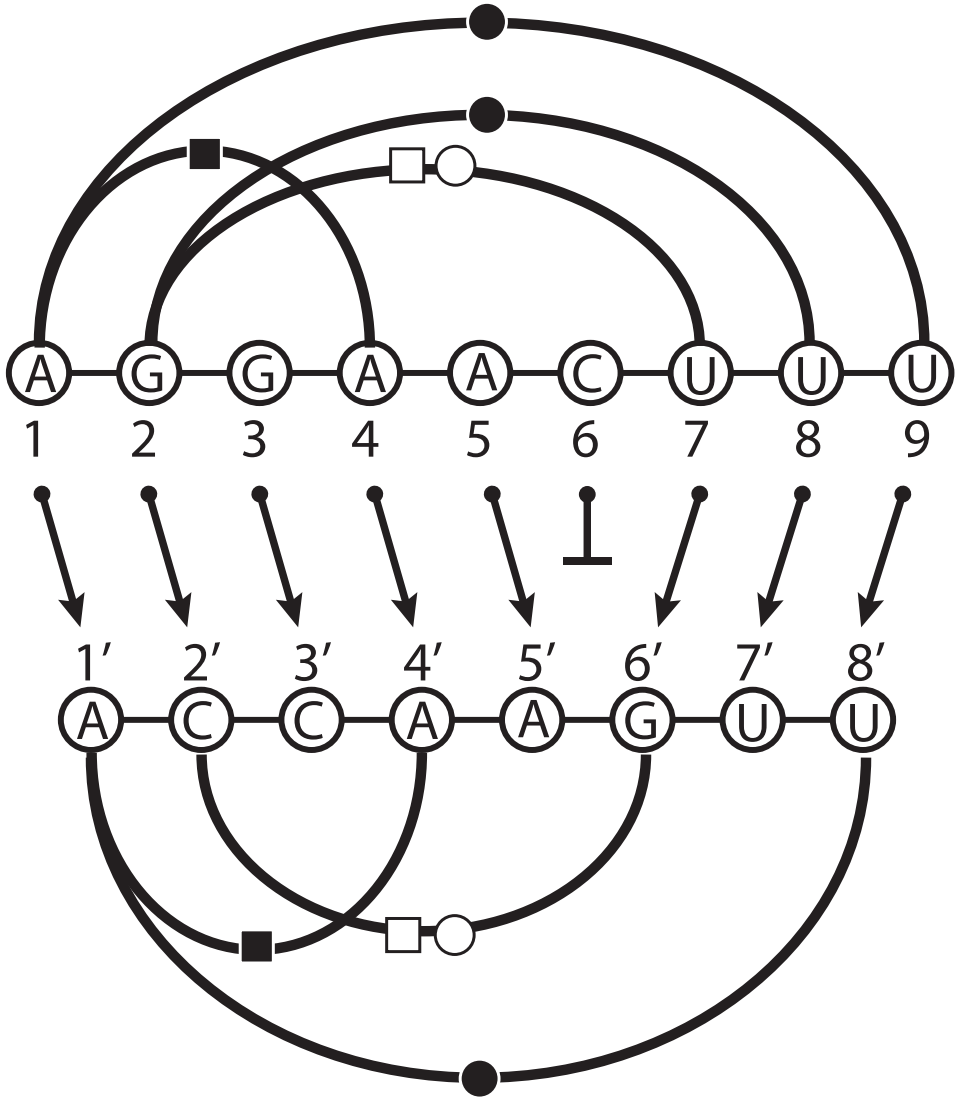
Sequence-structure alignment



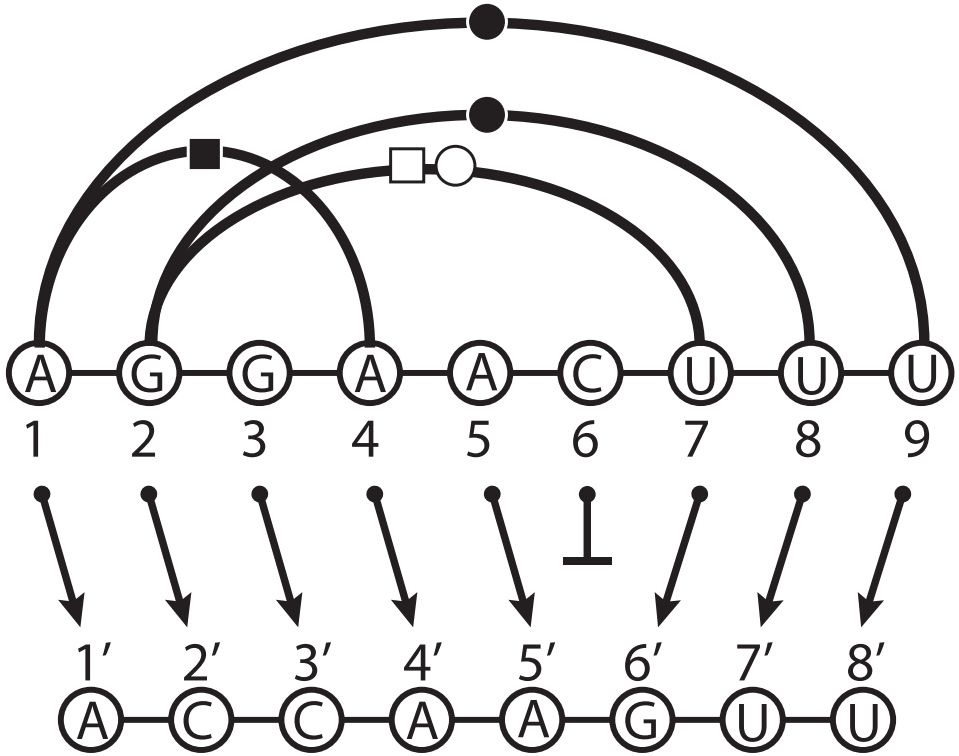
Sequence-structure alignment



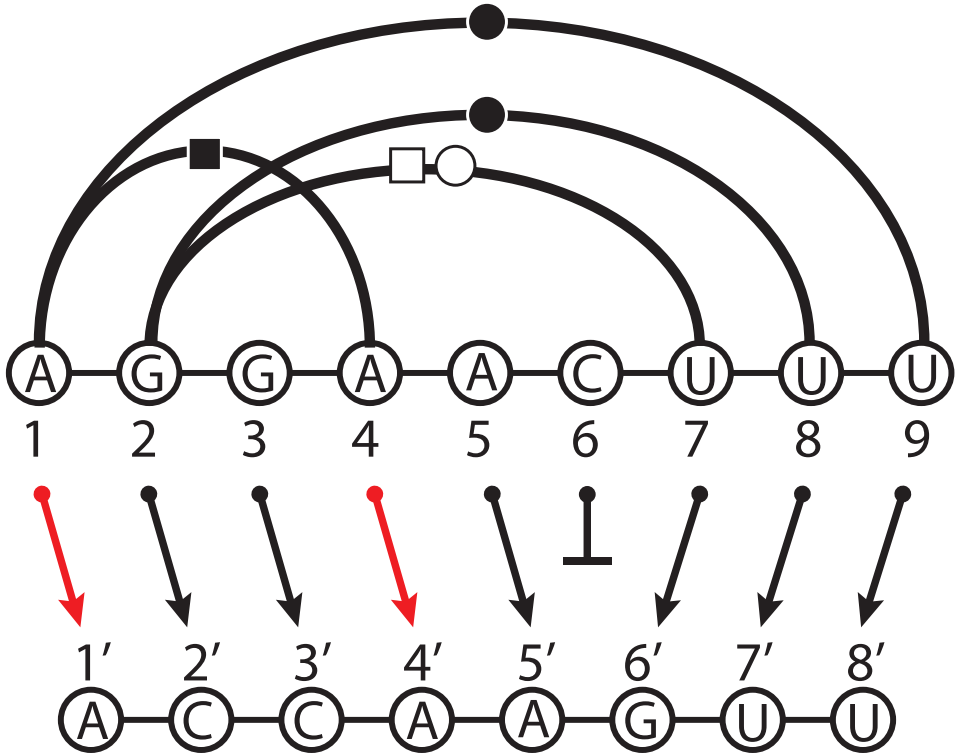
Sequence-structure alignment



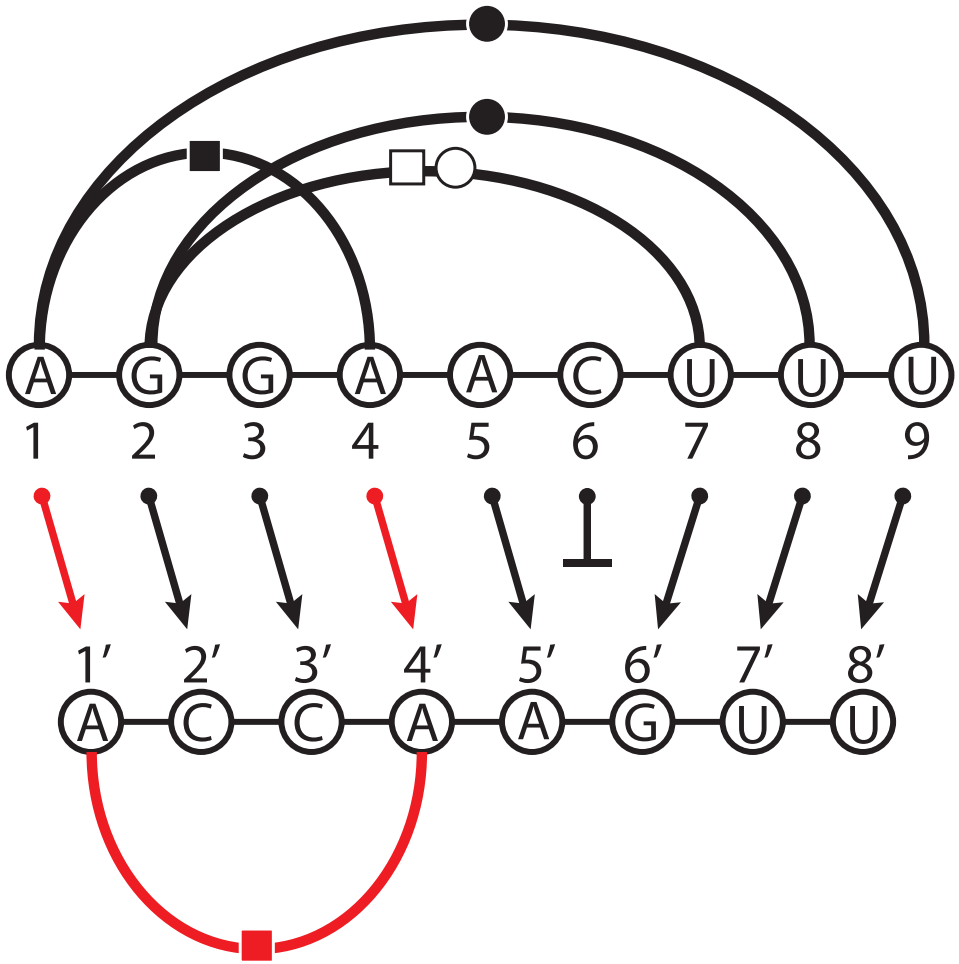
Sequence-structure alignment



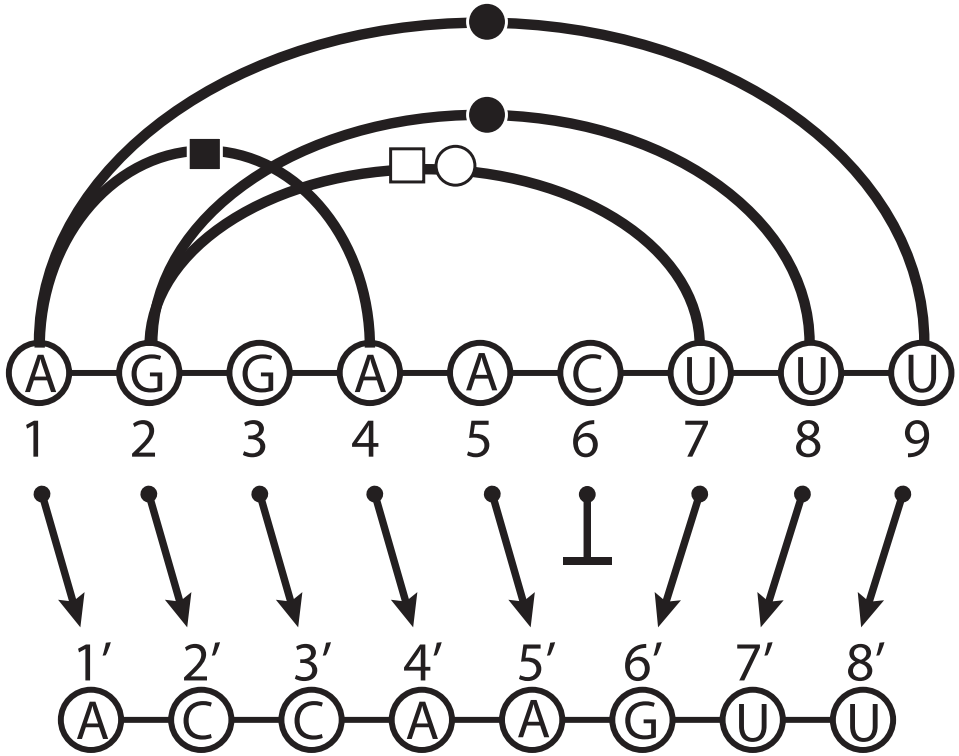
Sequence-structure alignment



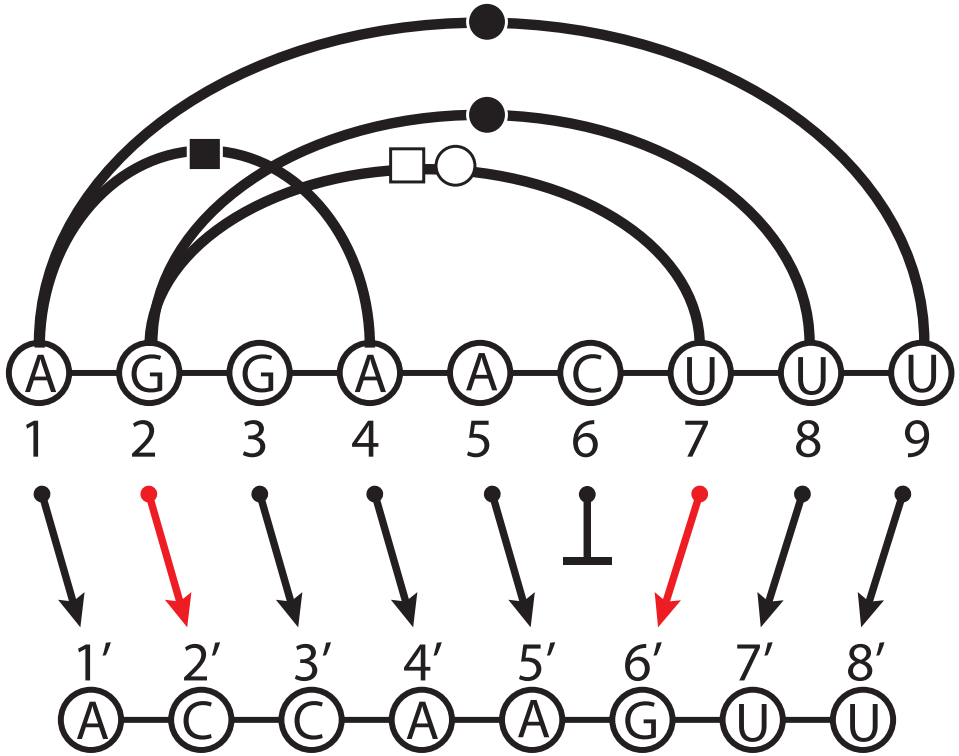
Sequence-structure alignment



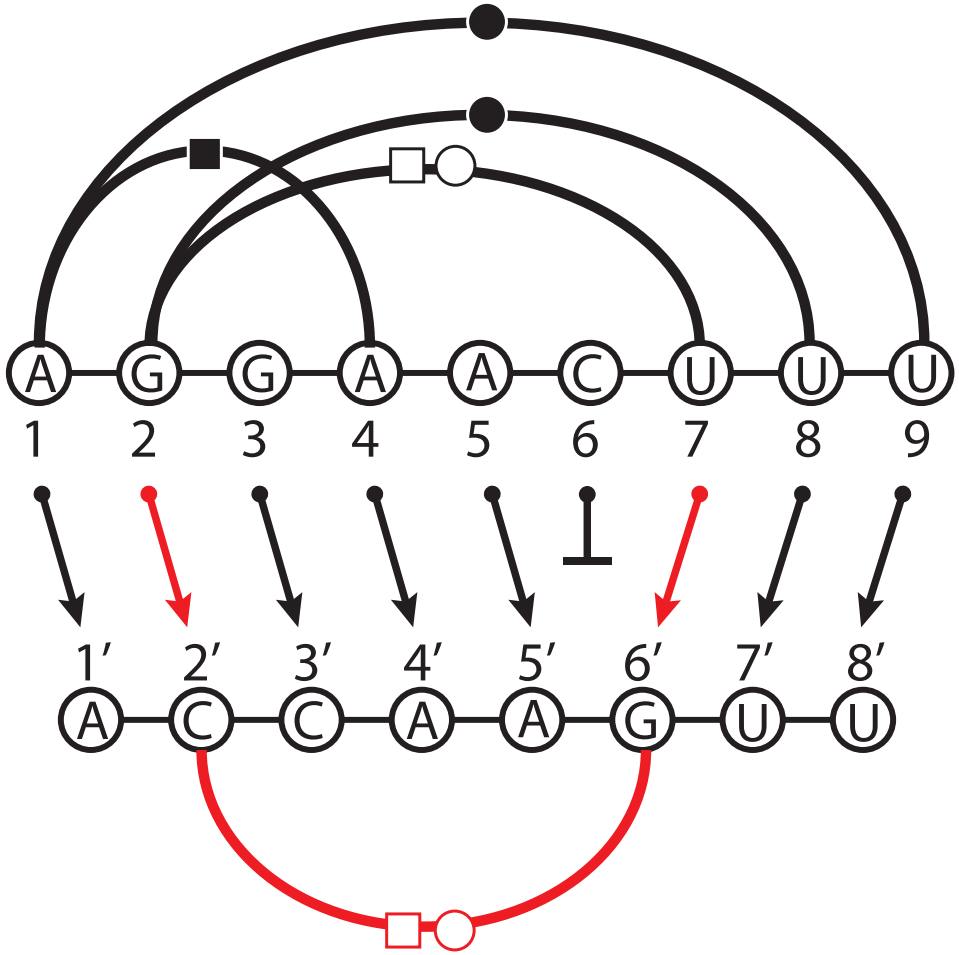
Sequence-structure alignment



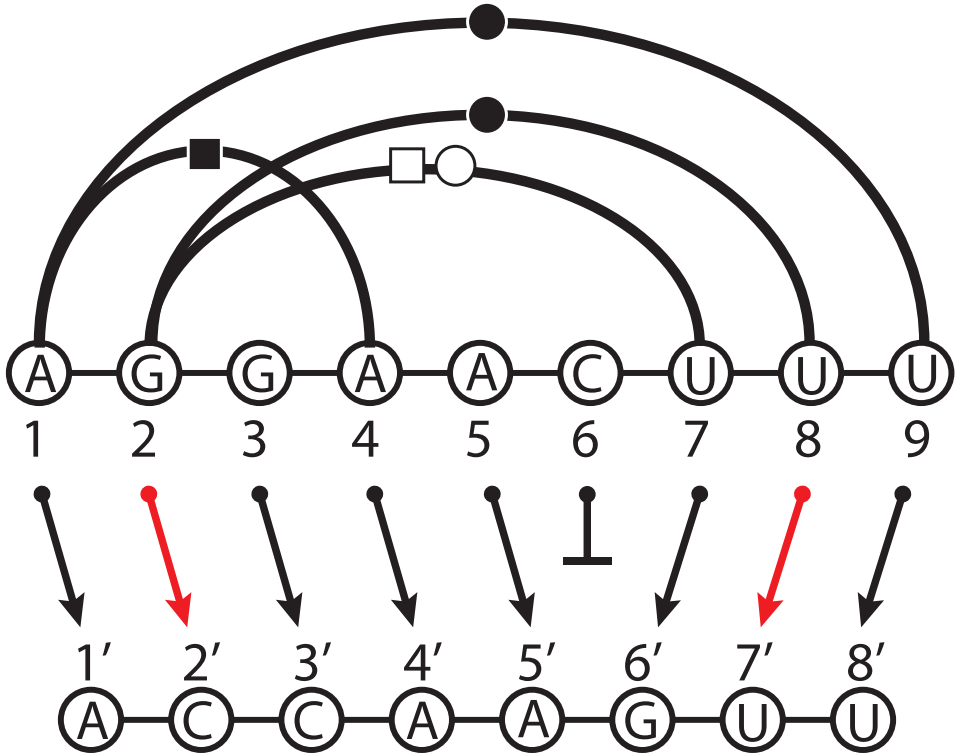
Sequence-structure alignment



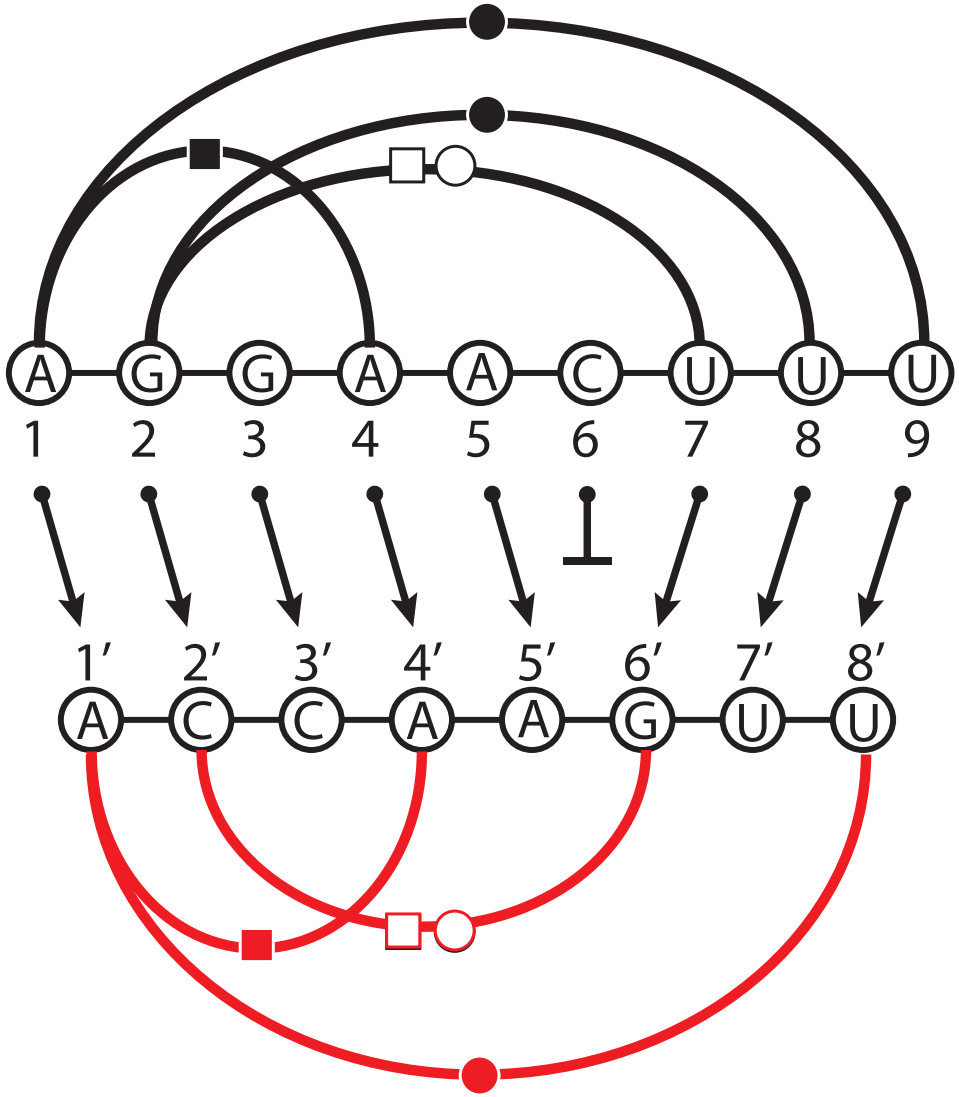
Sequence-structure alignment



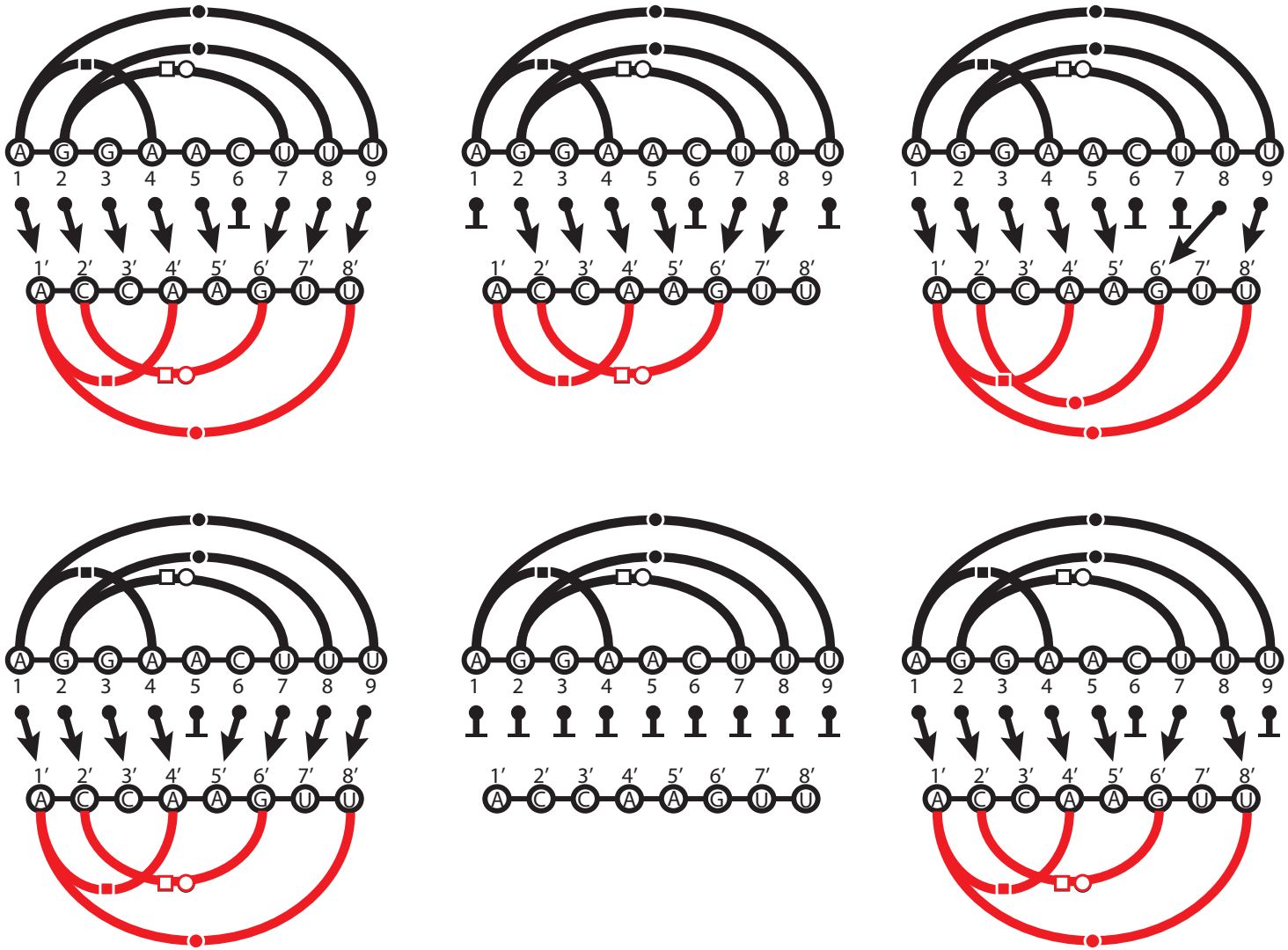
Sequence-structure alignment



Sequence-structure alignment

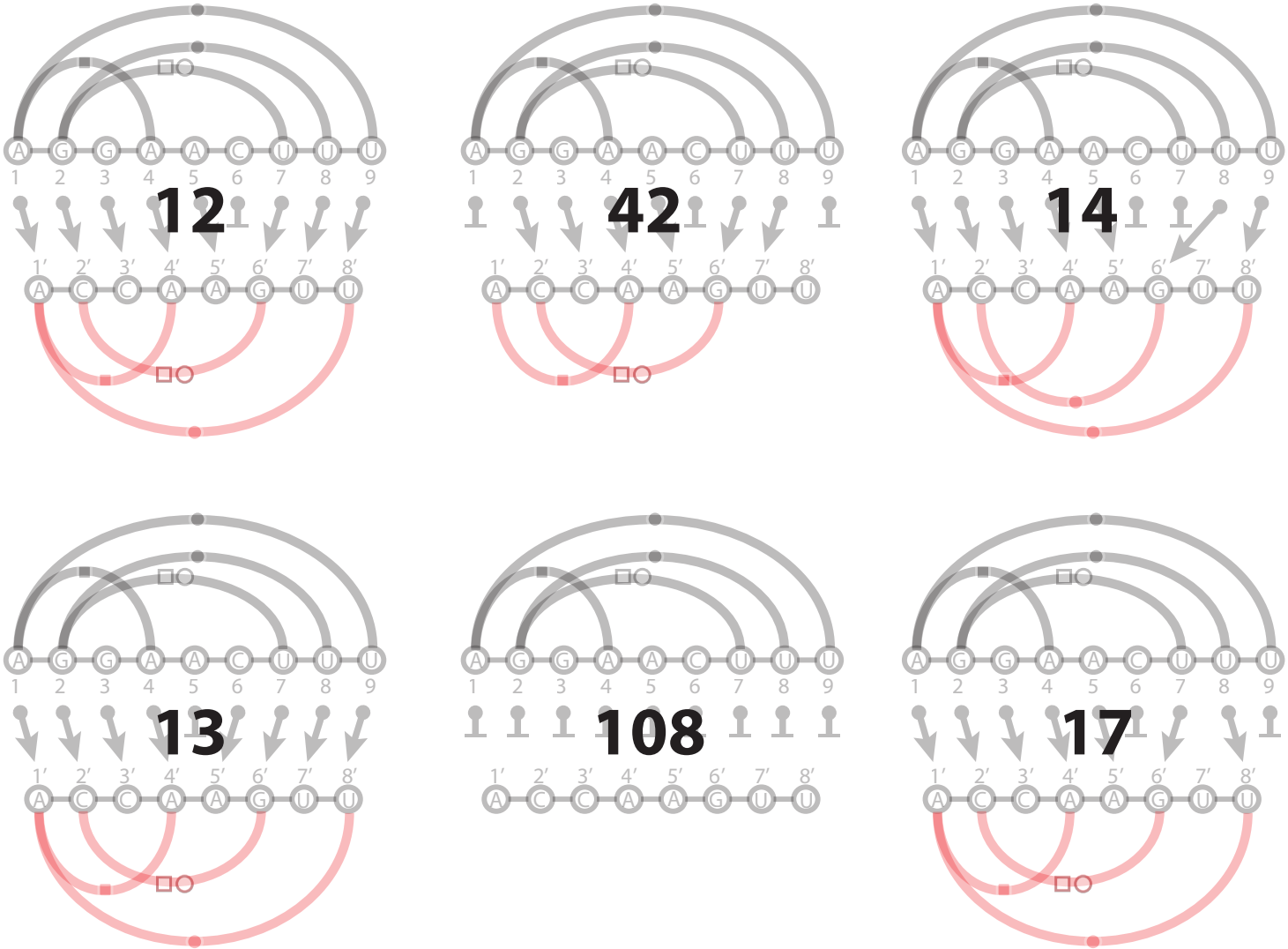


Sequence-structure alignment



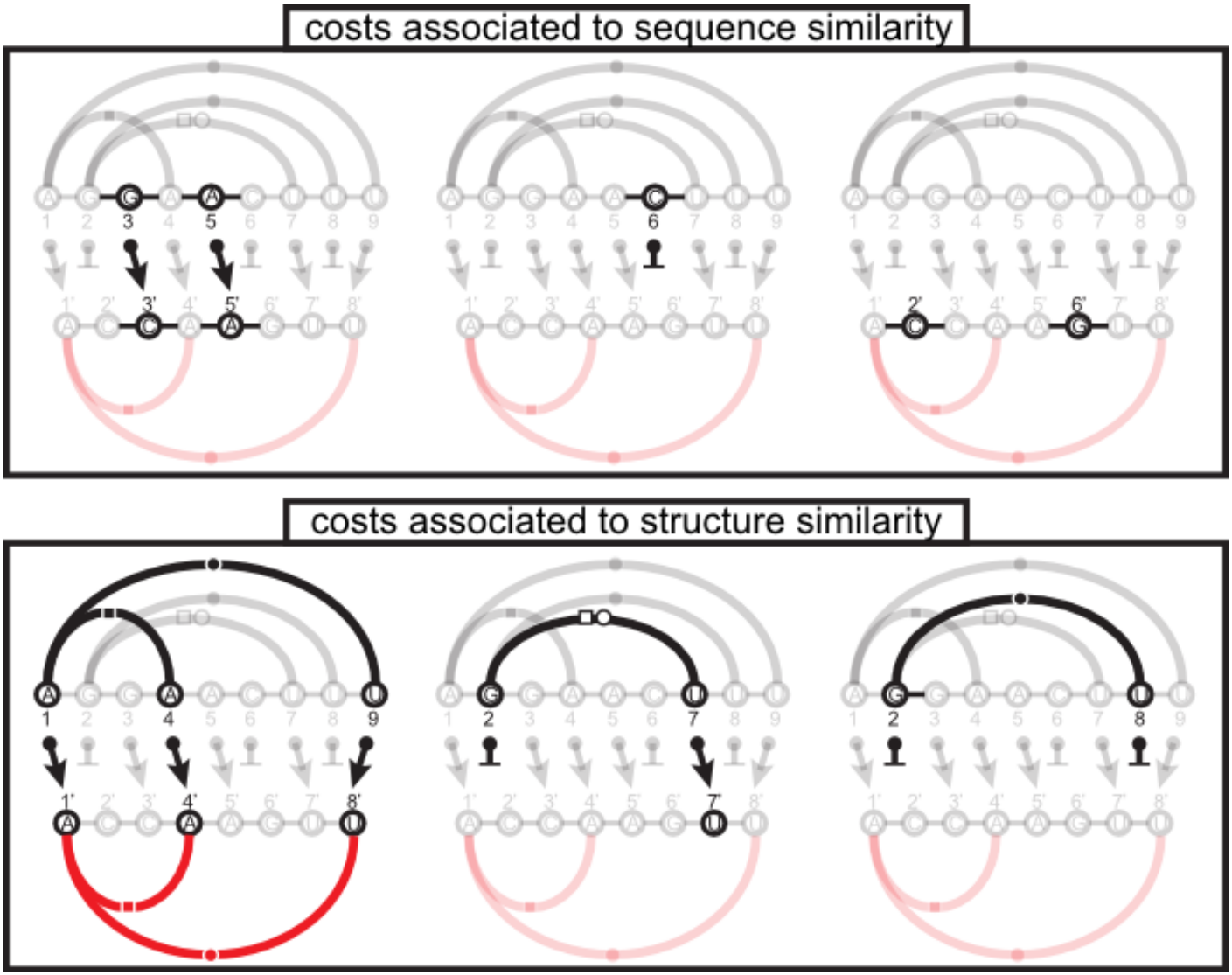
Problem: Find minimal-cost alignment (assignment subject to constraints)

Sequence-structure alignment

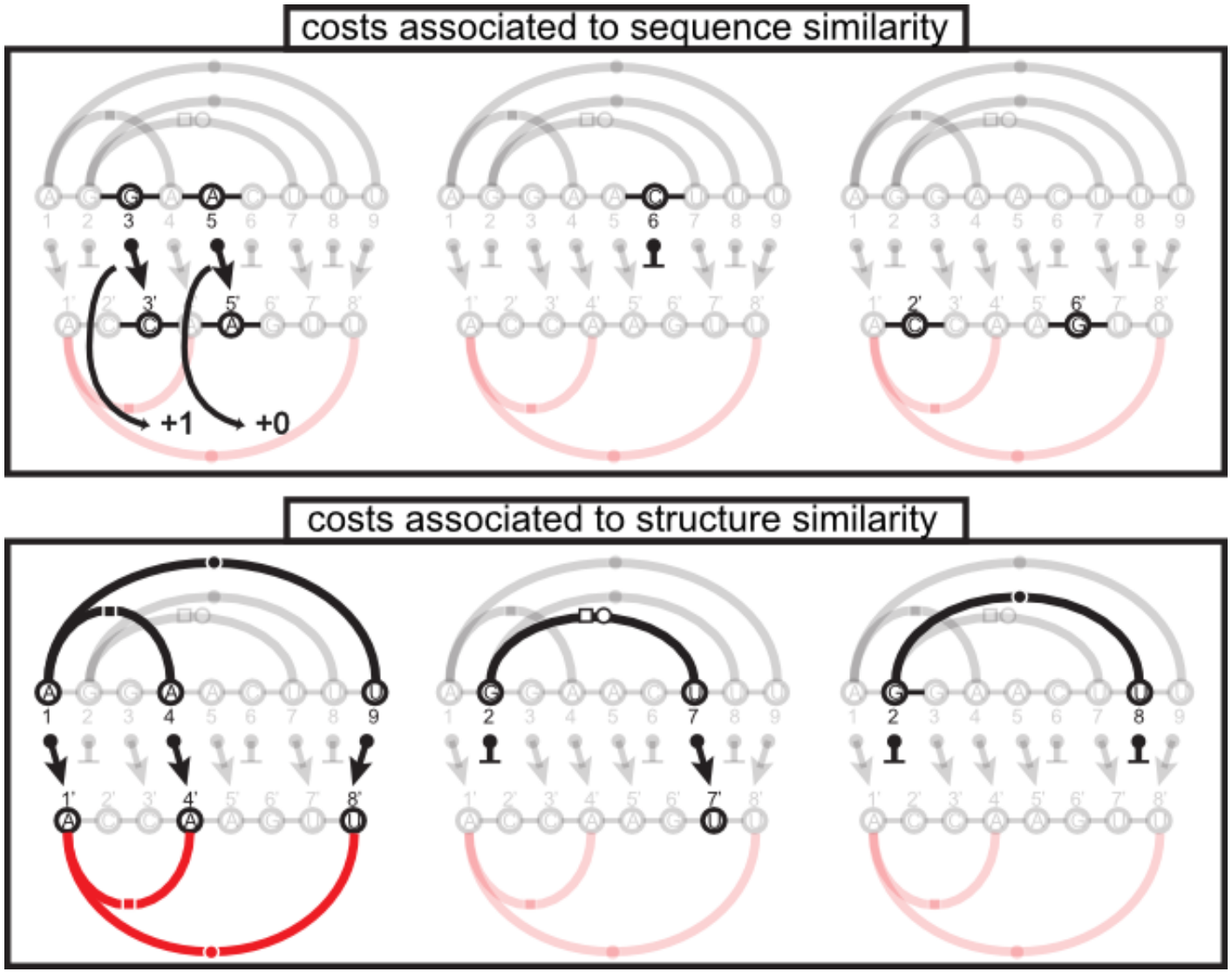


Problem: Find minimal-cost alignment (assignment subject to constraints)

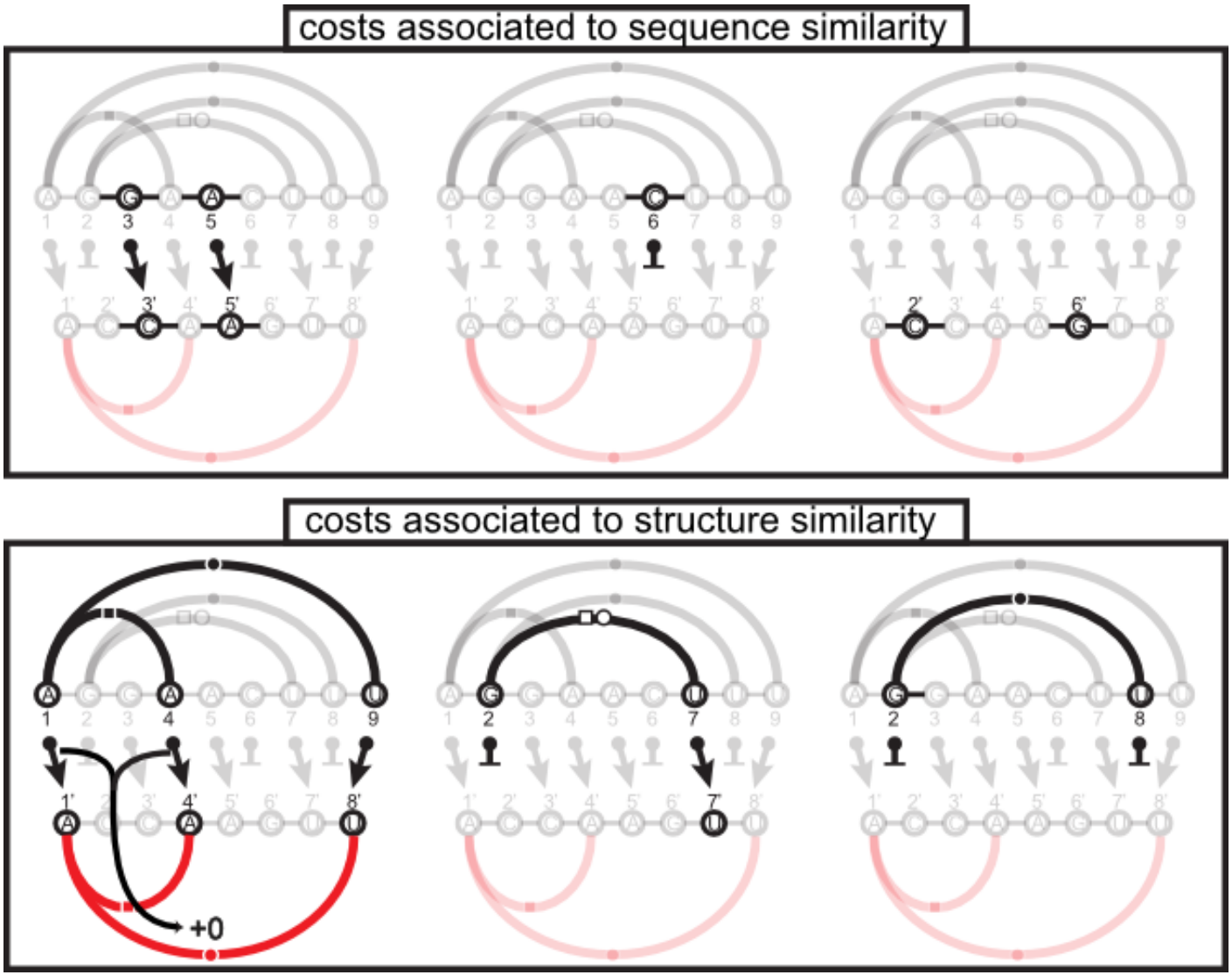
Alignment cost



Alignment cost

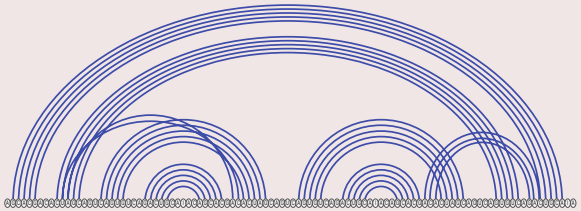


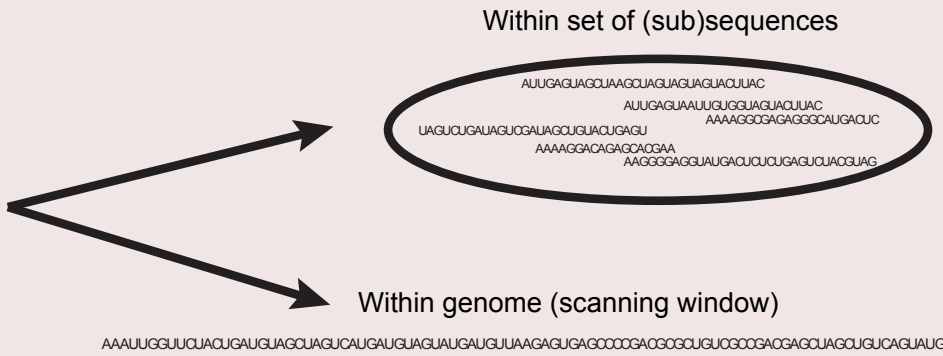
Alignment cost



Sequence structure alignment for ncRNA search and homology-modelling

Search for novel ncRNA instances

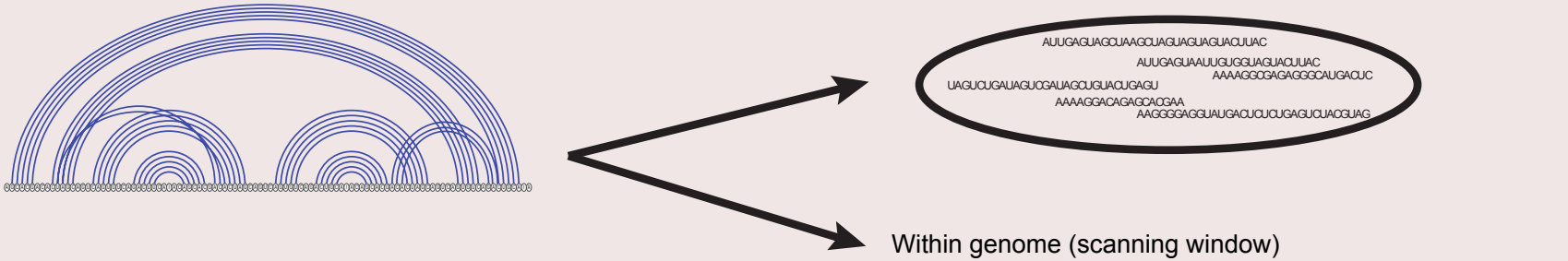




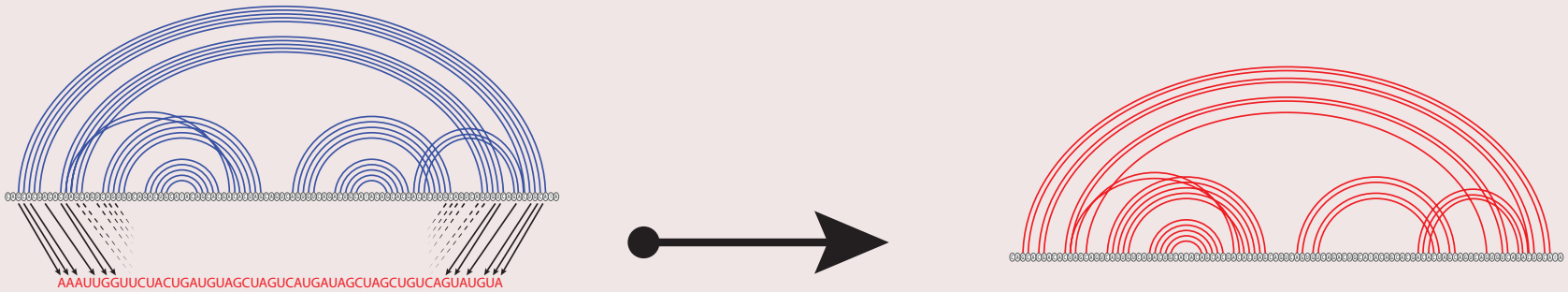
Structure prediction by homology modelling

Sequence structure alignment for ncRNA search and homology-modelling

Search for novel ncRNA instances



Structure prediction by homology modelling



Complexity of structure-sequence alignment

$n =$ Structure Length, $m =$ Sequence Length

Secondary Structure – Sequence	$O(n \cdot m^3)$
Pseudoknots – Sequence	MAX-SNP-Hard
Extended Secondary Structure – Sequence	MAX-SNP-Hard

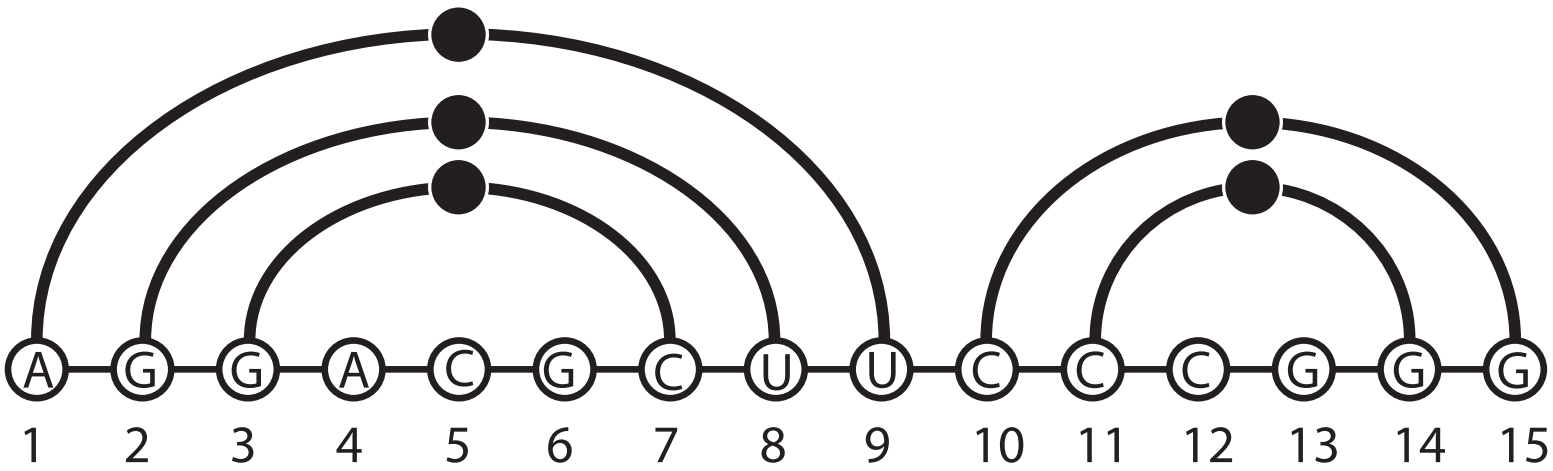
Jiang *et al.* 2001

Complexity of structure-sequence alignment

$n =$ Structure Length, $m =$ Sequence Length

Secondary structure – Sequence	$O(n \cdot m^3)$
Pseudoknots – Sequence	MAX-SNP-Hard
Extended Secondary Structure – Sequence	MAX-SNP-Hard

Jiang *et al.* 2001

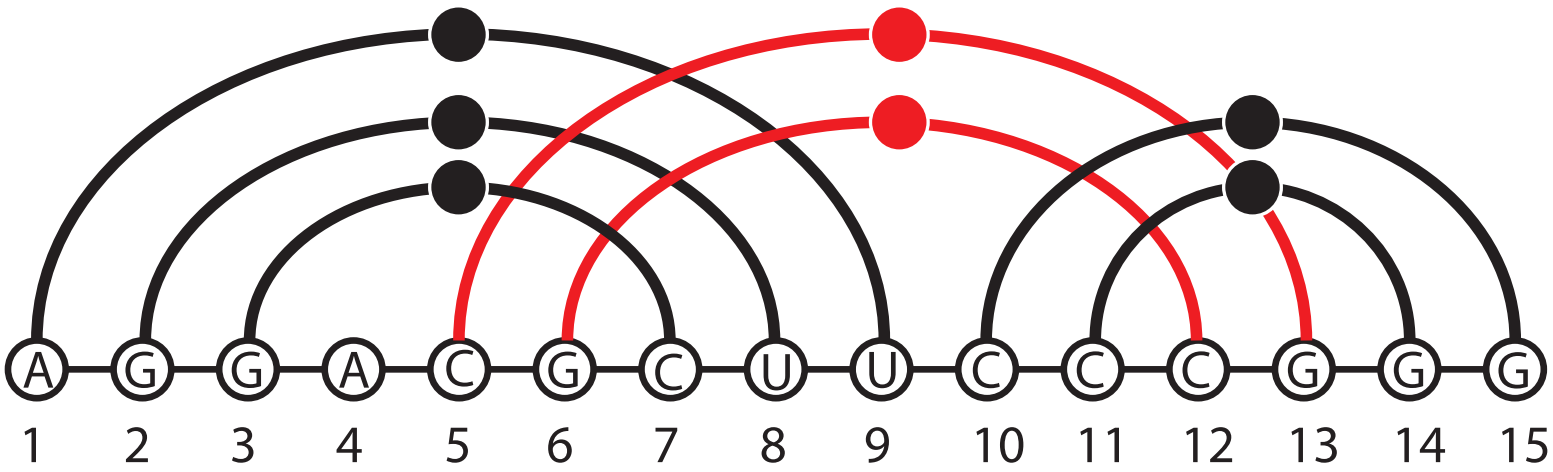


Complexity of structure-sequence alignment

$n =$ Structure Length, $m =$ Sequence Length

Secondary Structure – Sequence	$O(n \cdot m^3)$
Pseudoknots – Sequence	MAX-SNP-Hard
Extended Secondary Structure – Sequence	MAX-SNP-Hard

Jiang *et al.* 2001

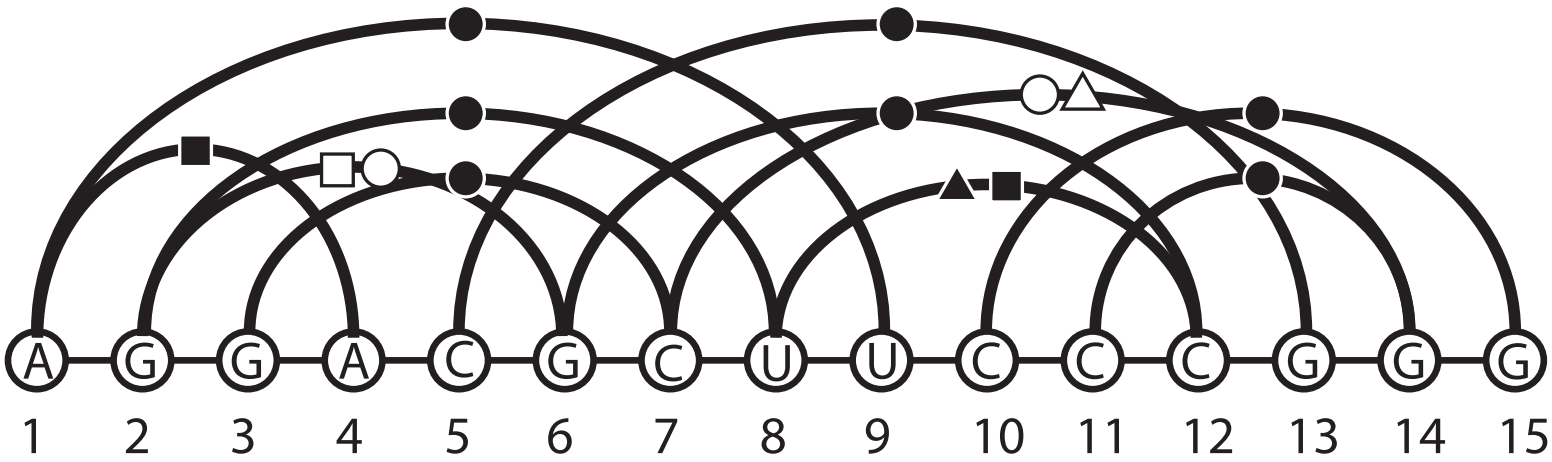


Complexity of structure-sequence alignment

$n =$ Structure Length, $m =$ Sequence Length

Secondary Structure – Sequence	$O(n \cdot m^3)$
Pseudoknots – Sequence	MAX-SNP-Hard
Extended Secondary Structure – Sequence	MAX-SNP-Hard

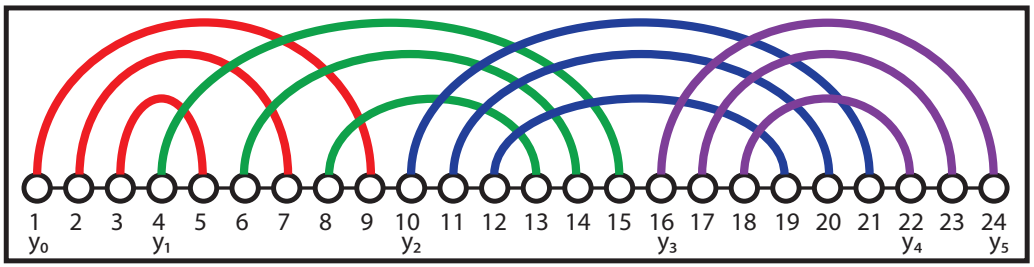
Jiang *et al.* 2001

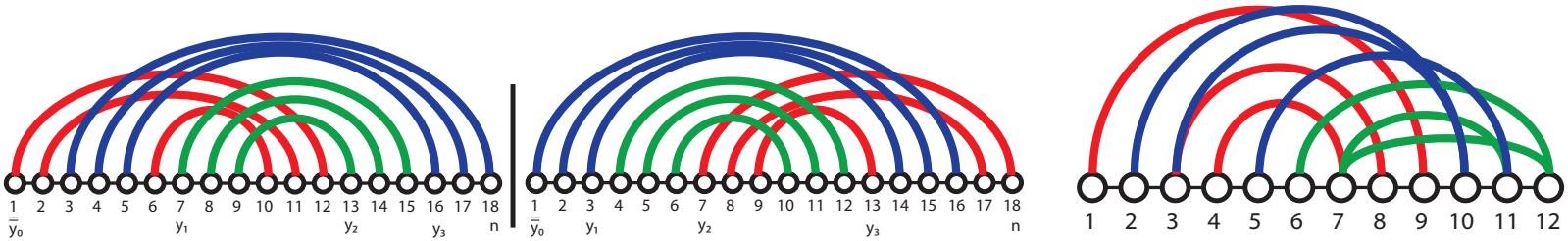


Complexity of struct.-seq. alignment: Polynomial classes

$n =$ Structure Length, $m =$ Sequence Length, $b =$ #Bands

Standard Pseudoknots	$O(n \cdot m^b)$
Standard Embedded Pseudoknots	$O(n \cdot m^{b+1})$
Simple Non-standard Pseudoknots	$O(n \cdot m^{b+1})$
Standard Triple Helices	$O(n \cdot m^3)$



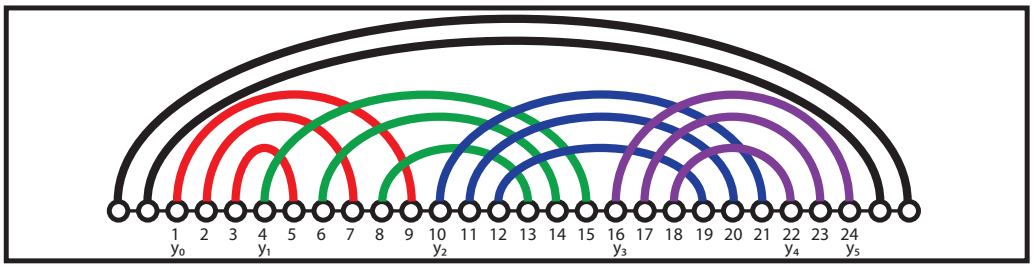


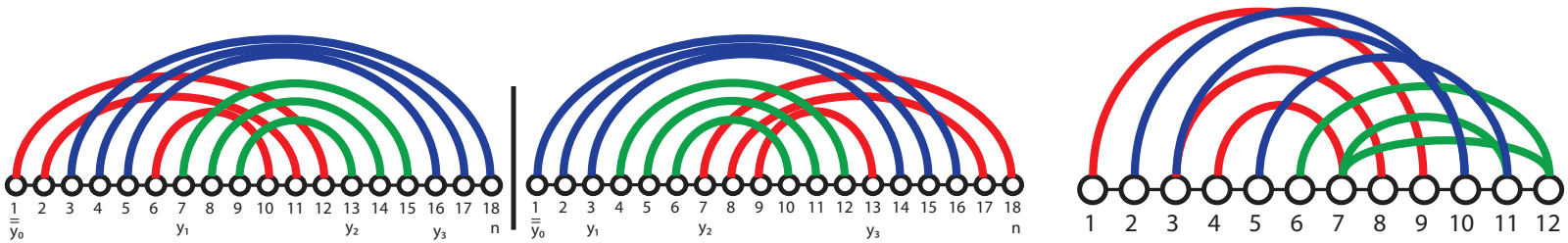
Han *et al.* 2008

Complexity of struct.-seq. alignment: Polynomial classes

$n =$ Structure Length, $m =$ Sequence Length, $b =$ #Bands

Standard Pseudoknots	$O(n \cdot m^b)$
Standard Embedded Pseudoknots	$O(n \cdot m^{b+1})$
Simple Non-standard Pseudoknots	$O(n \cdot m^{b+1})$
Standard Triple Helices	$O(n \cdot m^3)$



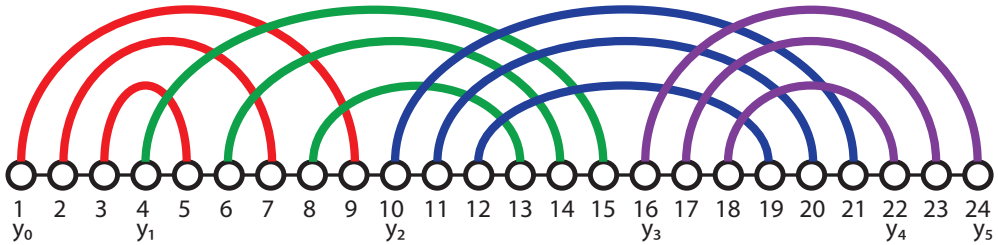


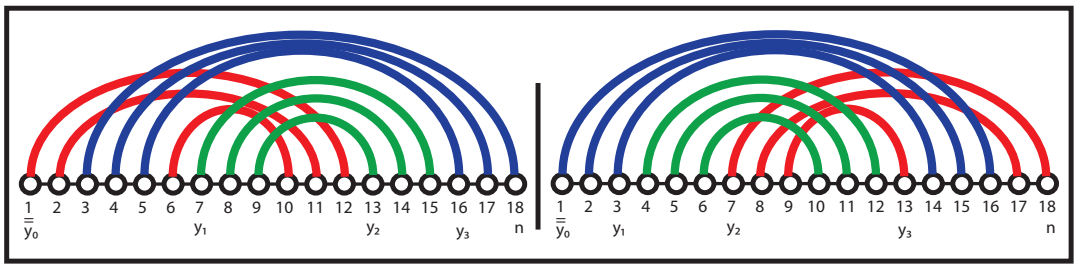
Han *et al.* 2008

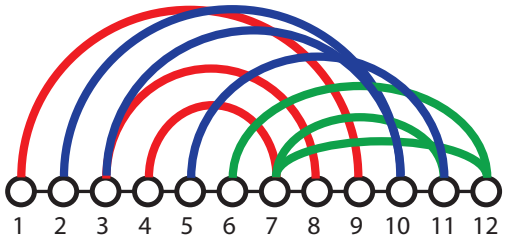
Complexity of struct.-seq. alignment: Polynomial classes

n = Structure Length, m = Sequence Length, b = #Bands

Standard Pseudoknots	$O(n \cdot m^b)$
Standard Embedded Pseudoknots	$O(n \cdot m^{b+1})$
Simple Non-standard Pseudoknots	$O(n \cdot m^{b+1})$
Standard Triple Helices	$O(n \cdot m^3)$





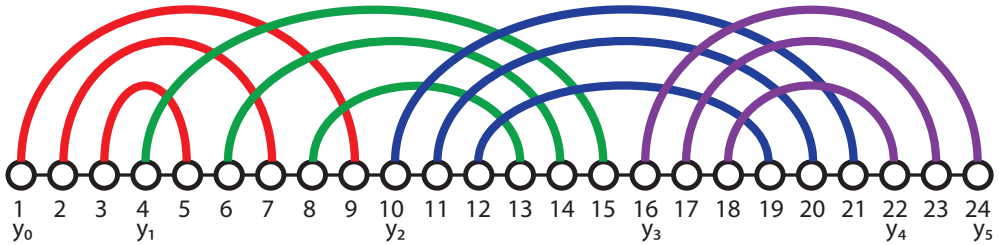


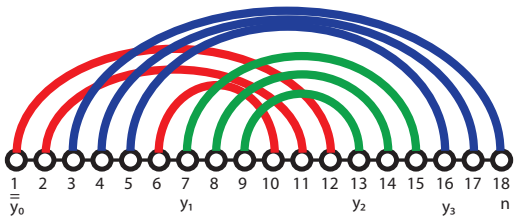
Wong *et al.* 2011

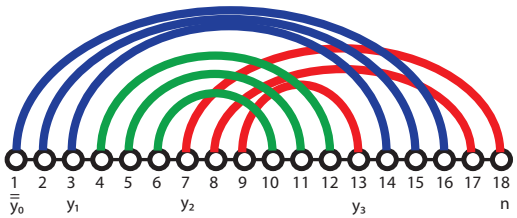
Complexity of struct.-seq. alignment: Polynomial classes

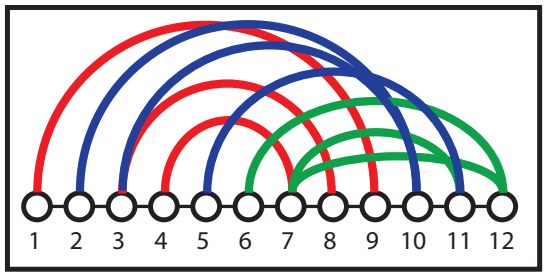
$n =$ Structure Length, $m =$ Sequence Length, $b =$ #Bands

Standard Pseudoknots	$O(n \cdot m^b)$
Standard Embedded Pseudoknots	$O(n \cdot m^{b+1})$
Simple Non-standard Pseudoknots	$O(n \cdot m^{b+1})$
Standard Triple Helices	$O(n \cdot m^3)$







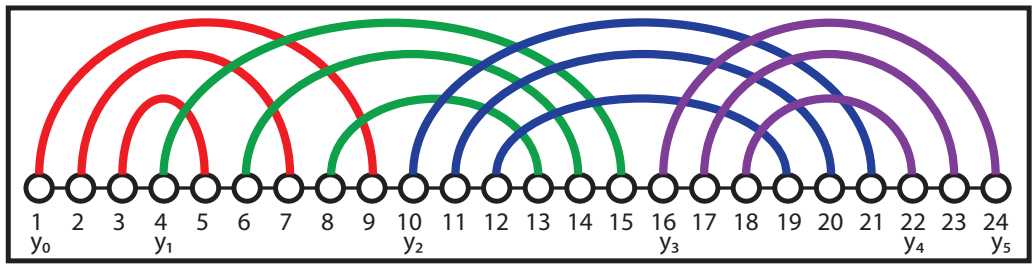


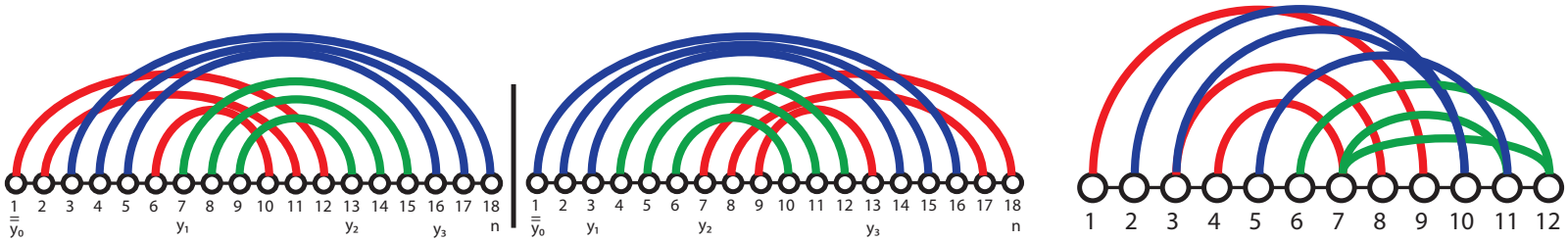
Wong *et al.* 2012

Complexity of struct.-seq. alignment: Polynomial classes

n = Structure Length, m = Sequence Length, b = #Bands

Standard Pseudoknots	$O(n \cdot m^b)$
Standard Embedded Pseudoknots	$O(n \cdot m^{b+1})$
Simple Non-standard Pseudoknots	$O(n \cdot m^{b+1})$
Standard Triple Helices	$O(n \cdot m^3)$





+ Other $O(n.m^4)/O(n.m^6)$ classes based on folding DP schemes

[Möhl/Will/Backofen 2009]

Complexity of struct.-seq. alignment: Polynomial classes

Message#1

No such a thing as free cupcakes

Just ask Marni about it. . .

Message#2

One class $\Rightarrow$ One algorithm

Is that really necessary. . . ?

What will you do with this RNA?

Message#3

Despite huge time/memory consumptions, existing algorithms disregard most non-canonical motifs/modules.

Complexity of struct.-seq. alignment: Polynomial classes

Message#1

No such a thing as free cupcakes

Just ask Marni about it. . .

Message#2

One class $\Rightarrow$ One algorithm

Is that really necessary. . . ?

What will you do with this RNA?

Message#3

Despite huge time/memory consumptions, existing algorithms disregard most non-canonical motifs/modules.

Complexity of struct.-seq. alignment: Polynomial classes

Message#1

No such a thing as free cupcakes

Just ask Marni about it. . .

Message#2

One class $\Rightarrow$ One algorithm

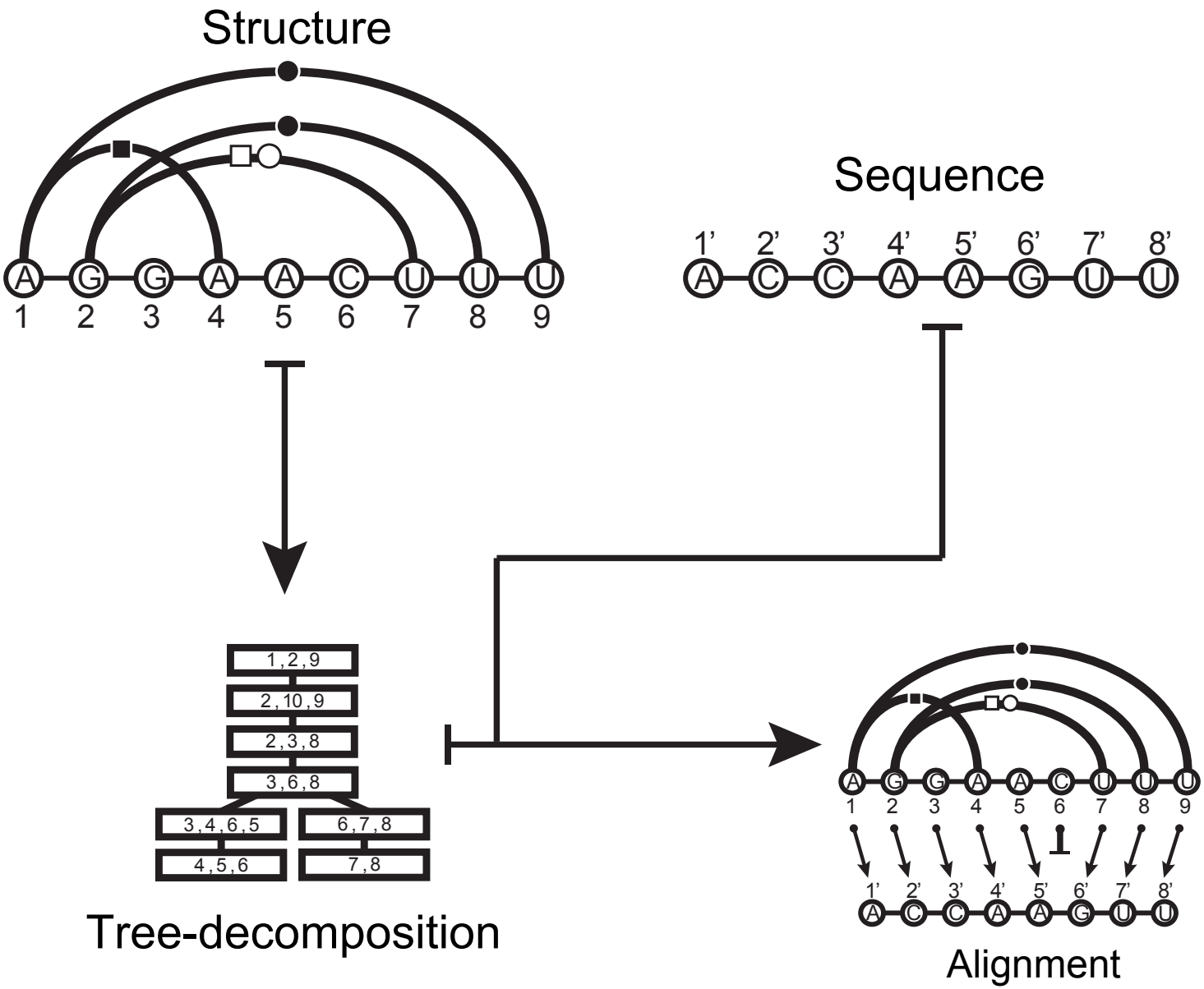
Is that really necessary. . . ?

What will you do with this RNA?

Message#3

Despite huge time/memory consumptions, existing algorithms disregard most non-canonical motifs/modules.

Outline of general parameterized approach



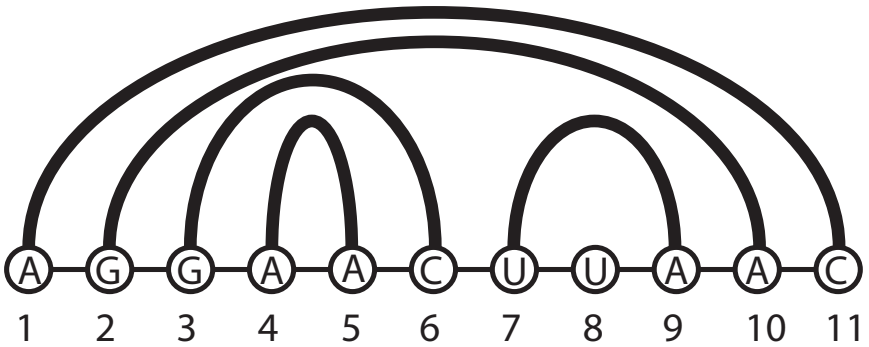
Tree decomposition of RNA structure

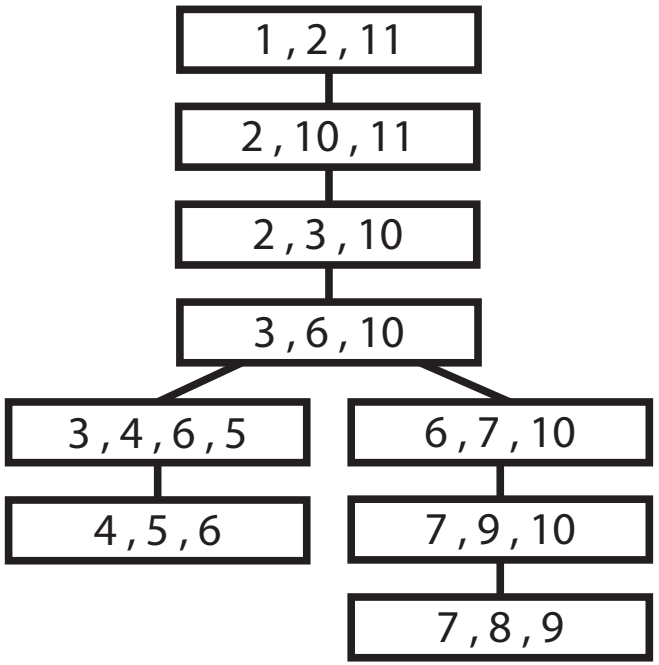
Structure-centric alignment ⇒ Constraints

- Adjacent positions in structure → Precedence
- Paired positions → Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





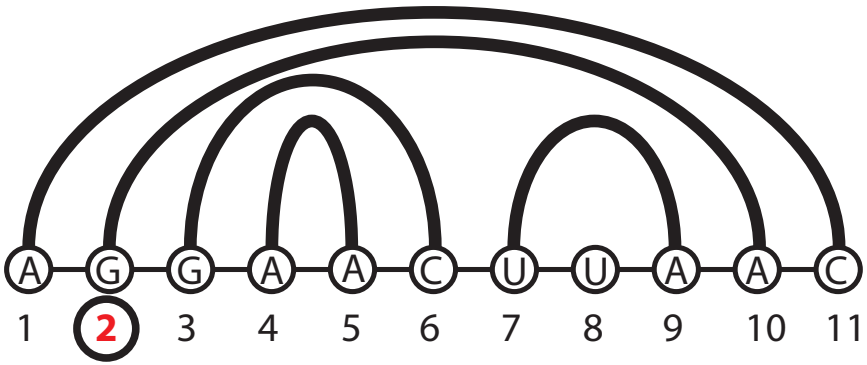
Tree decomposition of RNA structure

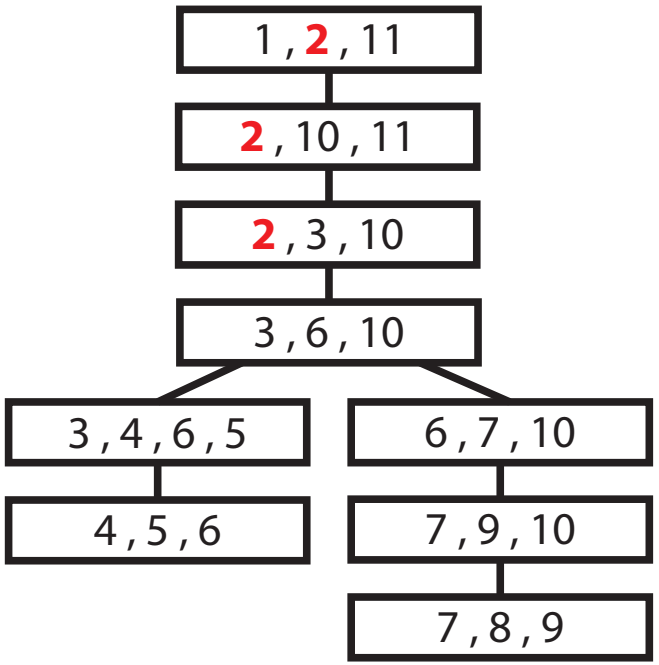
Structure-centric alignment ⇒ Constraints

- Adjacent positions in structure → Precedence
- Paired positions → Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





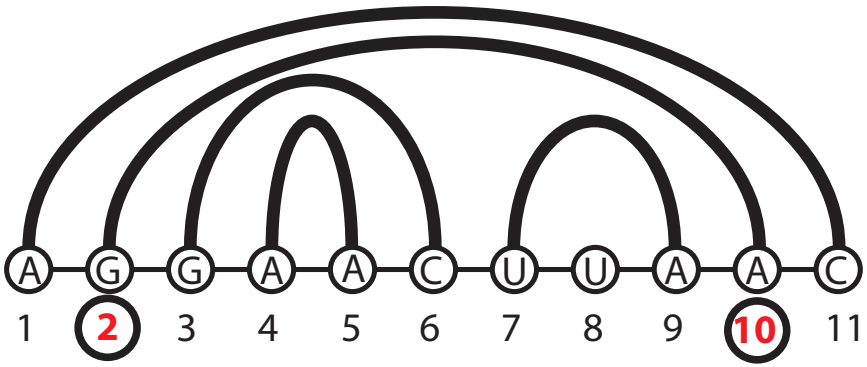
Tree decomposition of RNA structure

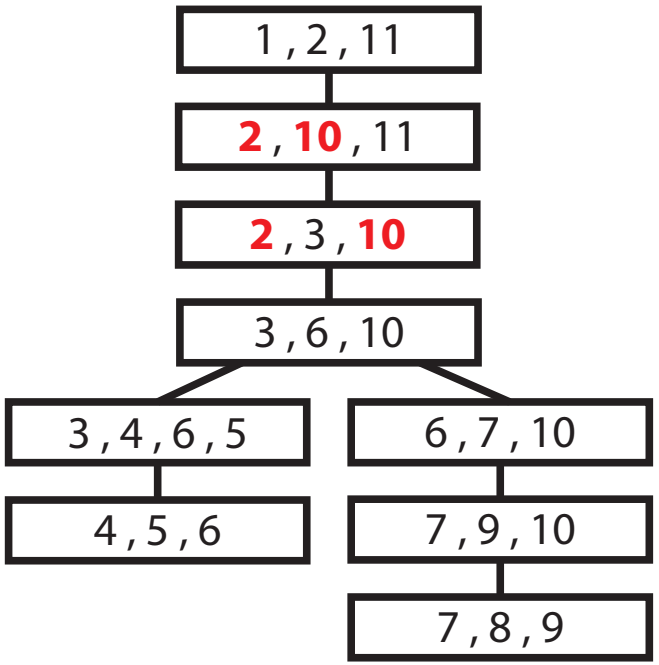
Structure-centric alignment ⇒ Constraints

- Adjacent positions in structure → Precedence
- Paired positions → Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





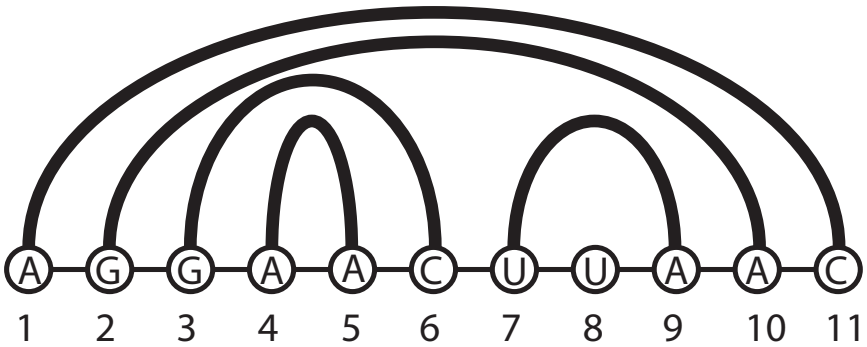
Tree decomposition of RNA structure

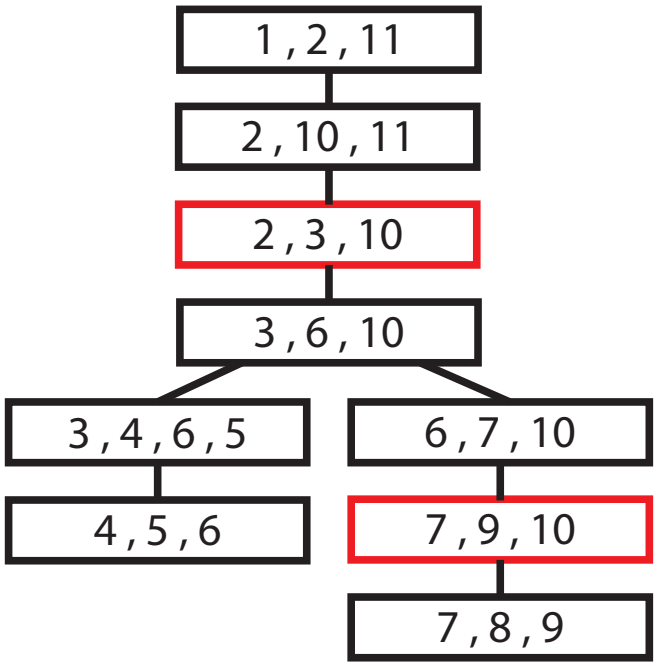
Structure-centric alignment $\Rightarrow$ Constraints

- Adjacent positions in structure $\rightarrow$ Precedence
- Paired positions $\rightarrow$ Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





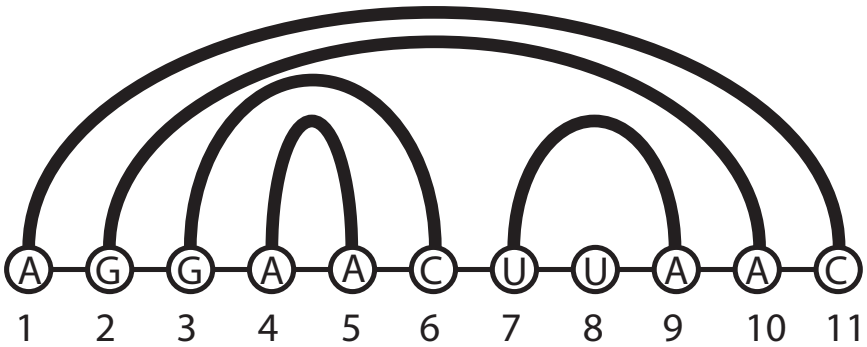
Tree decomposition of RNA structure

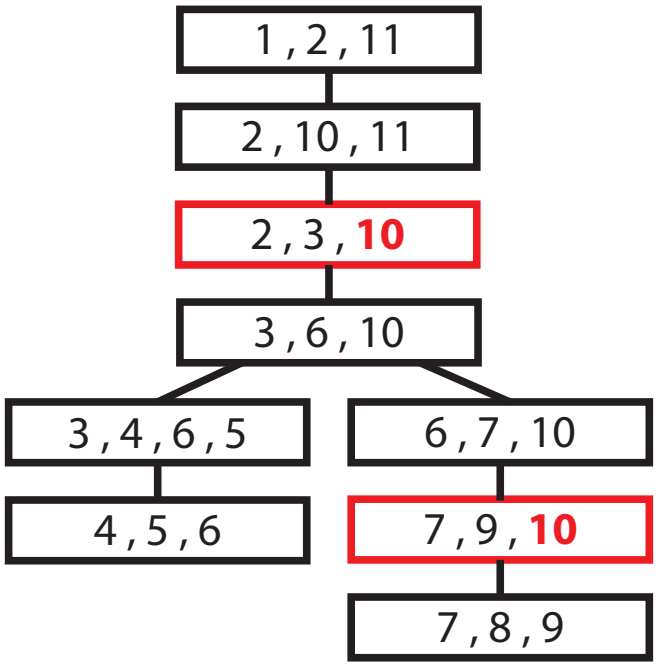
Structure-centric alignment $\Rightarrow$ Constraints

- Adjacent positions in structure $\rightarrow$ Precedence
- Paired positions $\rightarrow$ Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





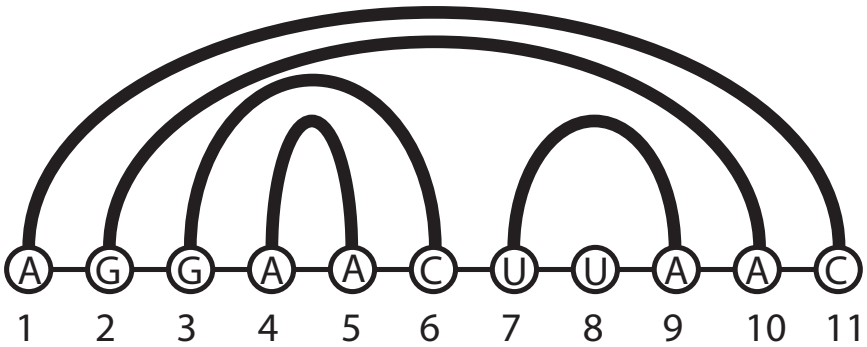
Tree decomposition of RNA structure

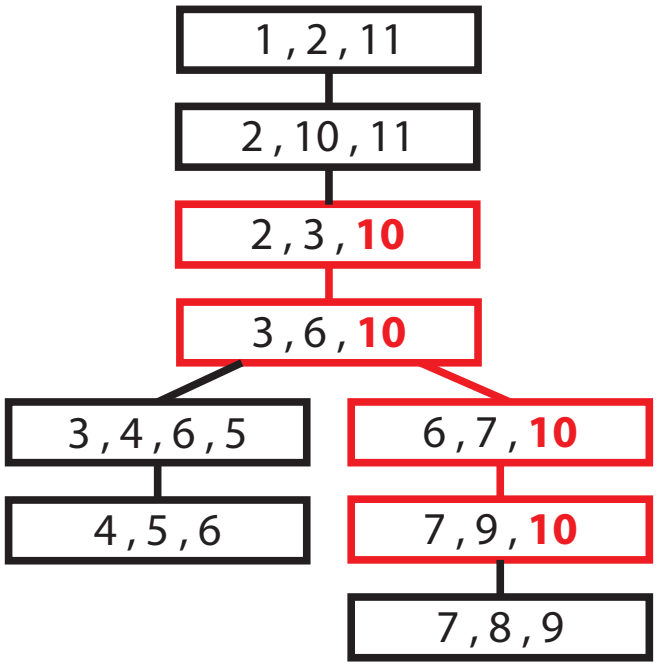
Structure-centric alignment $\Rightarrow$ Constraints

- Adjacent positions in structure $\rightarrow$ Precedence
- Paired positions $\rightarrow$ Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$





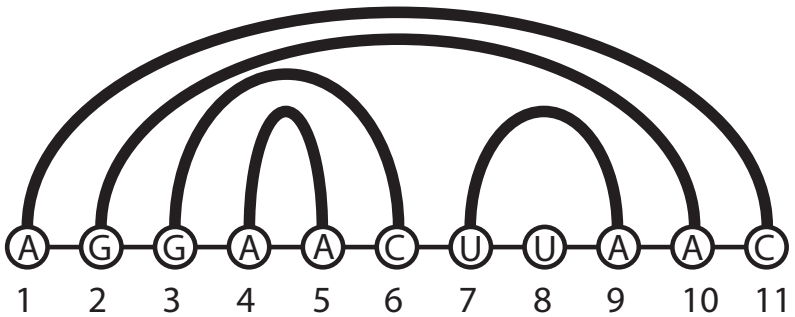
Tree decomposition of RNA structure

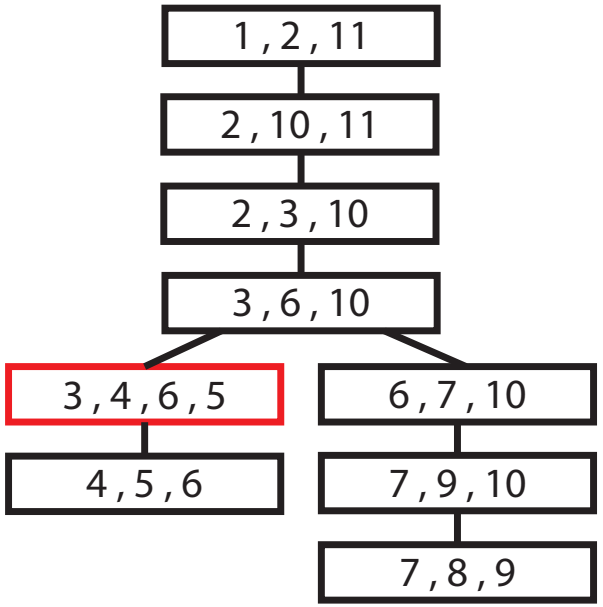
Structure-centric alignment ⇒ Constraints

- Adjacent positions in structure → Precedence
- Paired positions → Both partners needed to assign score

Sets of structure-side positions (bags $\{\mathcal{B}_i\}$), in a tree such that:

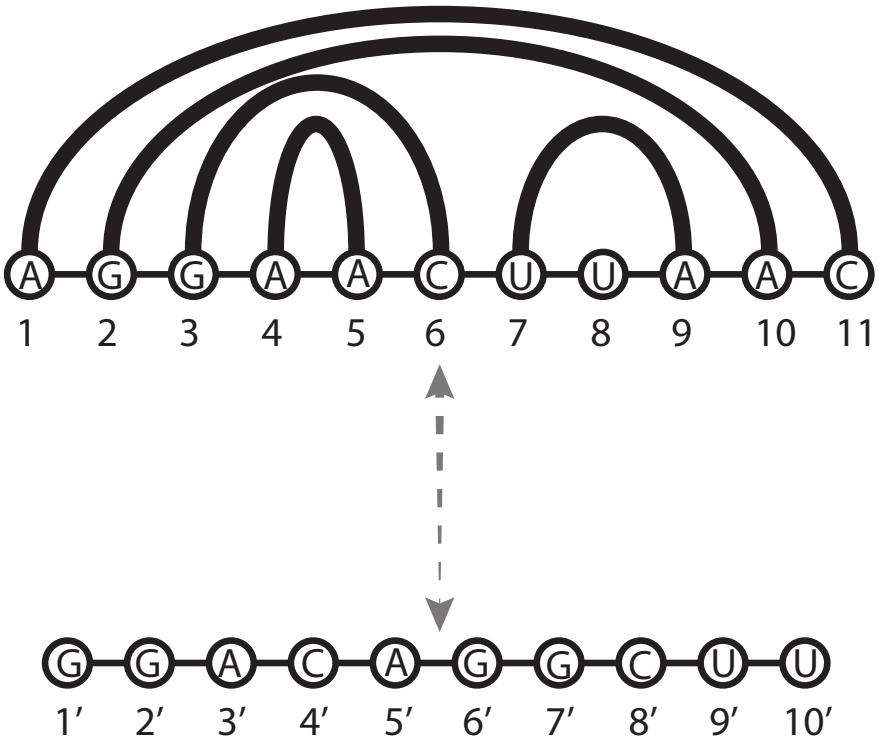
- Every position in the structure appears at least once
- Each interacting pair of positions simultaneously appear in ≥ 1 bag
- If $x \in \mathcal{B} \cap \mathcal{B}'$, than x is in every bag $\mathcal{B}''$ on the path from $\mathcal{B}$ to $\mathcal{B}'$

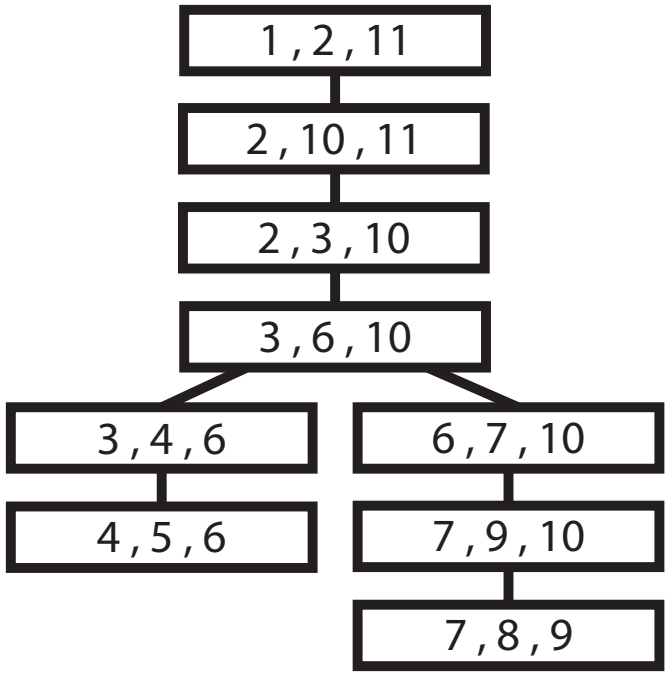




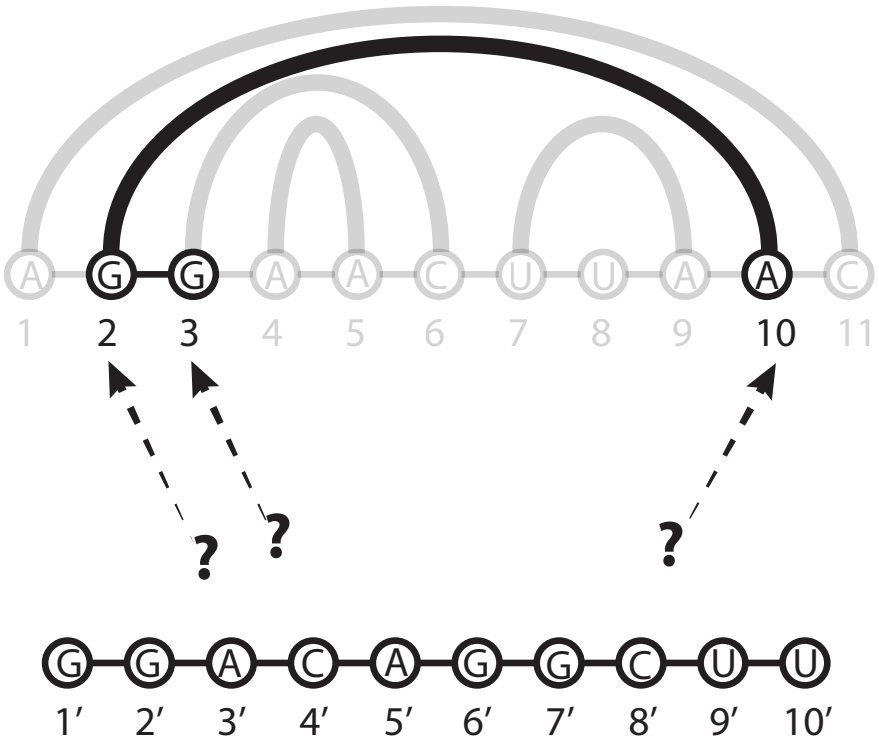
Width k = Size of biggest bag minus one.

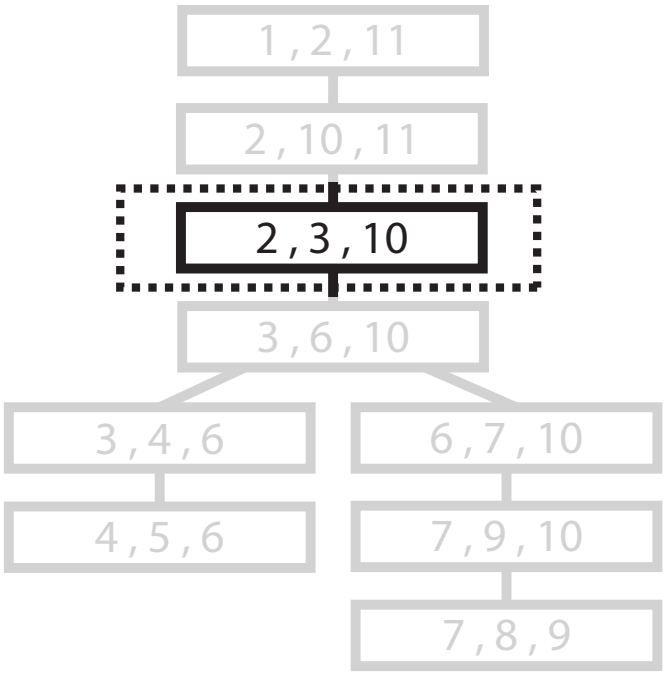
Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]



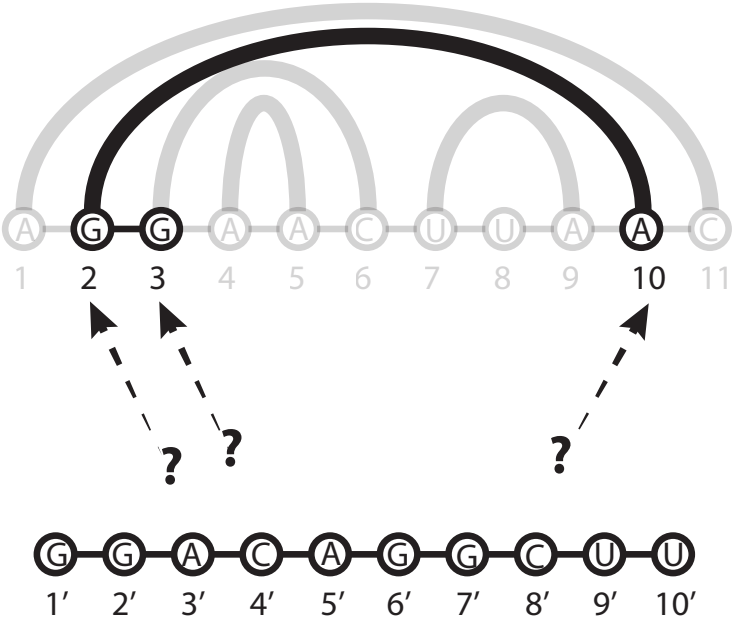


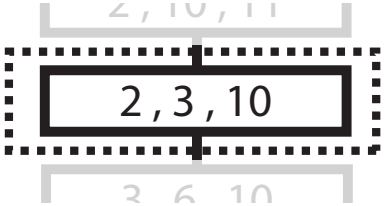
Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]





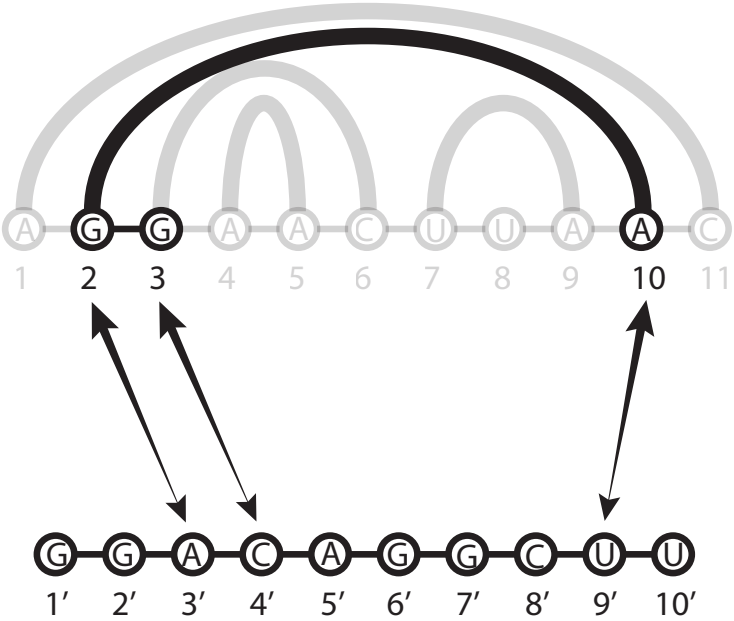
Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]

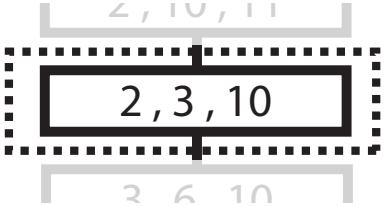




2	3	10	costs
1′	2′	3′	
1′	2′	4′	
3′	4′	9′	
7′	9′	10′	
8′	9′	10′	

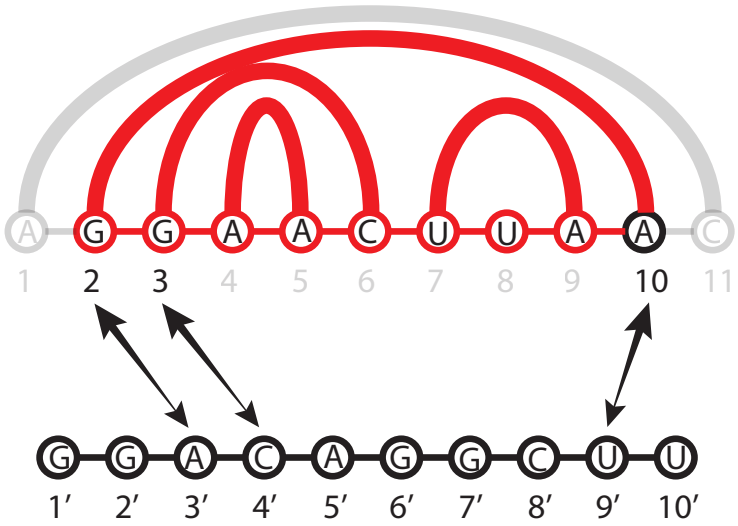
Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]

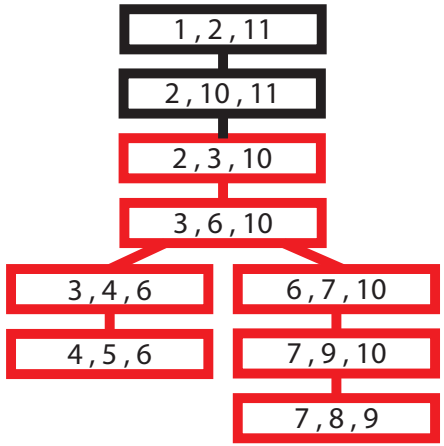


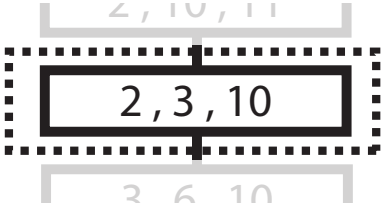


2	3	10	costs
1′	2′	3′	
1′	2′	4′	
3′	**4′**	**9′**	
7′	9′	10′	
8′	9′	10′	

Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]

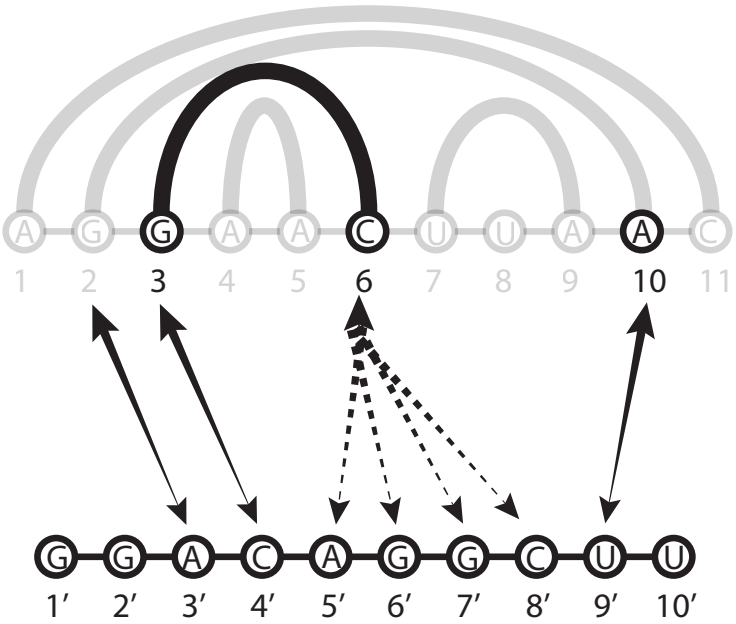


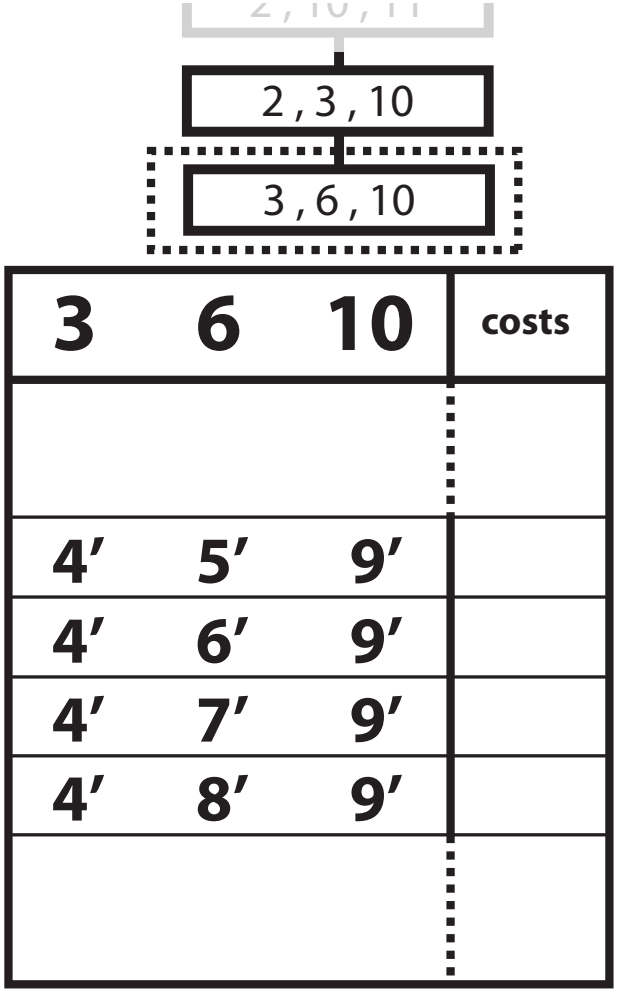


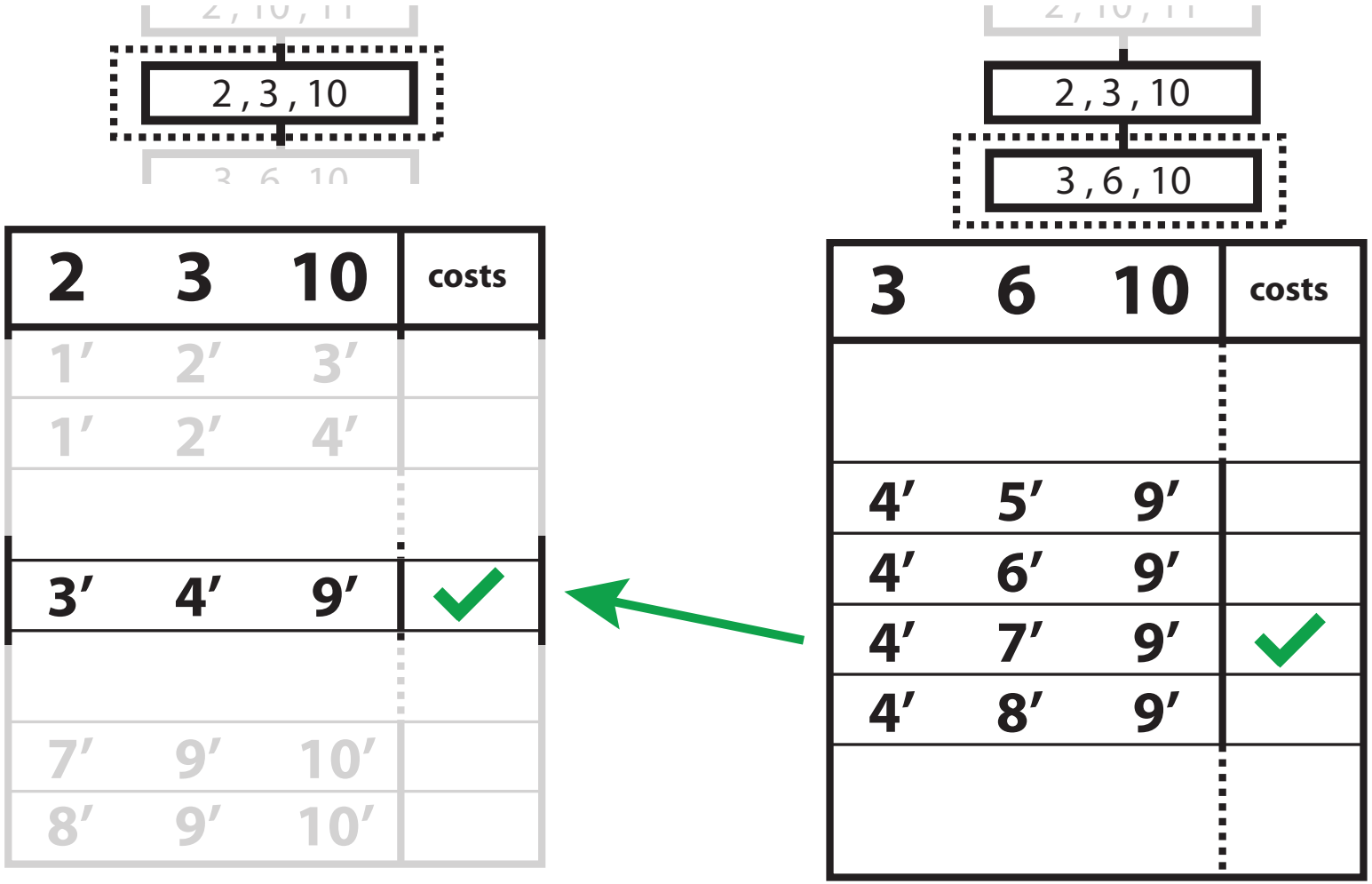


2	3	10	costs
1′	2′	3′	
1′	2′	4′	
3′	**4′**	**9′**	✗
7′	9′	10′	
8′	9′	10′	

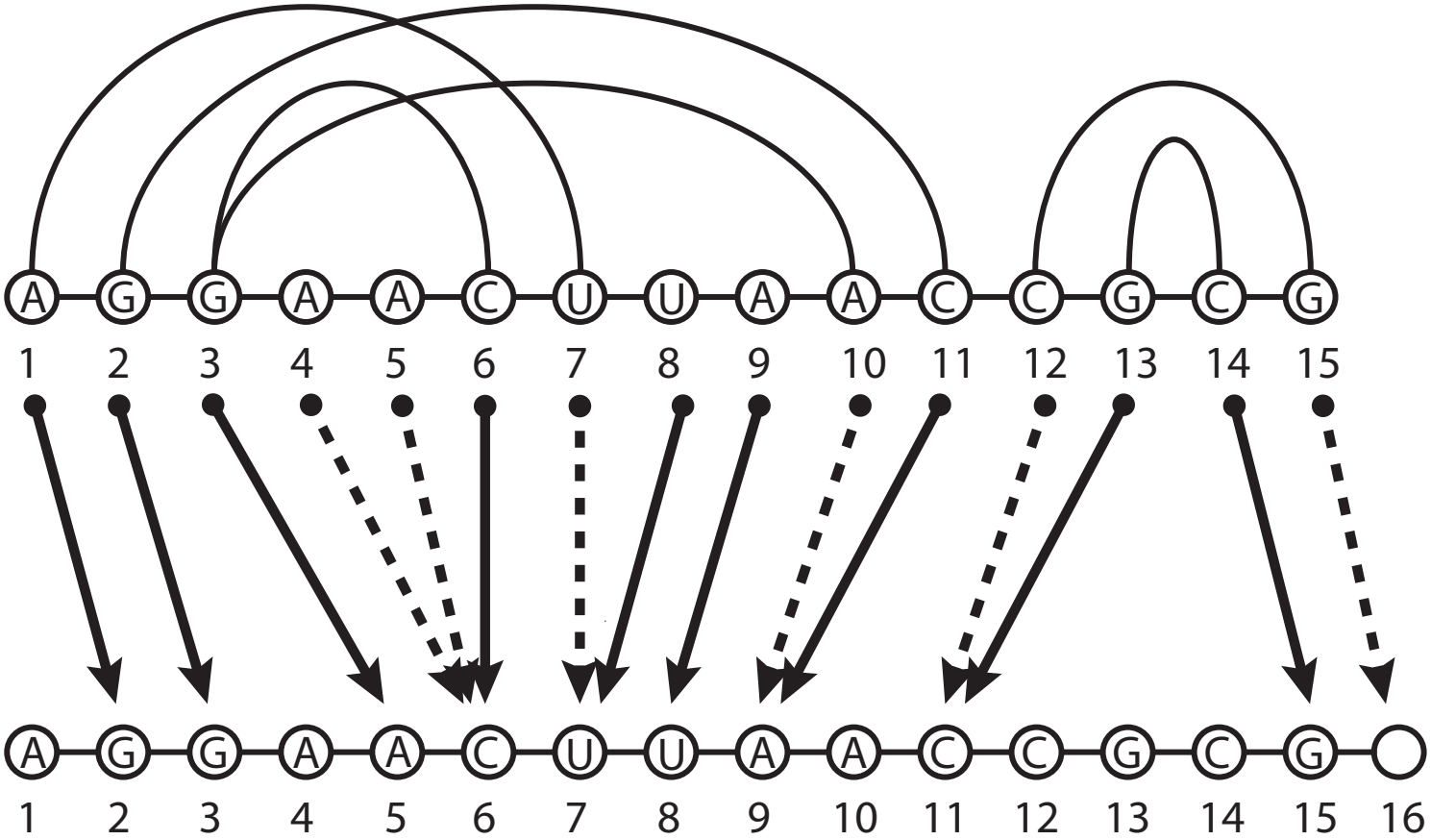
Tree-Decomposition-based Alignment [*Rinaudo et al., WABI 2012*]






2 , 3 , 10
3 , 6 , 10
2 3 10 costs
1′ 2′ 3′
1′ 2′ 4′
3′ 4′ 9′
7′ 9′ 10′
8′ 9′ 10′
3 6 10 costs
4′ 5′ 9′
4′ 6′ 9′
4′ 7′ 9′
4′ 8′ 9′

Encoding structure-sequence alignments



Theorem

- Structure of length n
- Sequence of length m
- Tree decomposition of structure, having width k

$\Rightarrow$ Best structure-sequence alignment can be computed in $\mathcal{O}\left(n.m^{k+1}\right)$ time and $\mathcal{O}\left(n.m^{k}\right)$ space

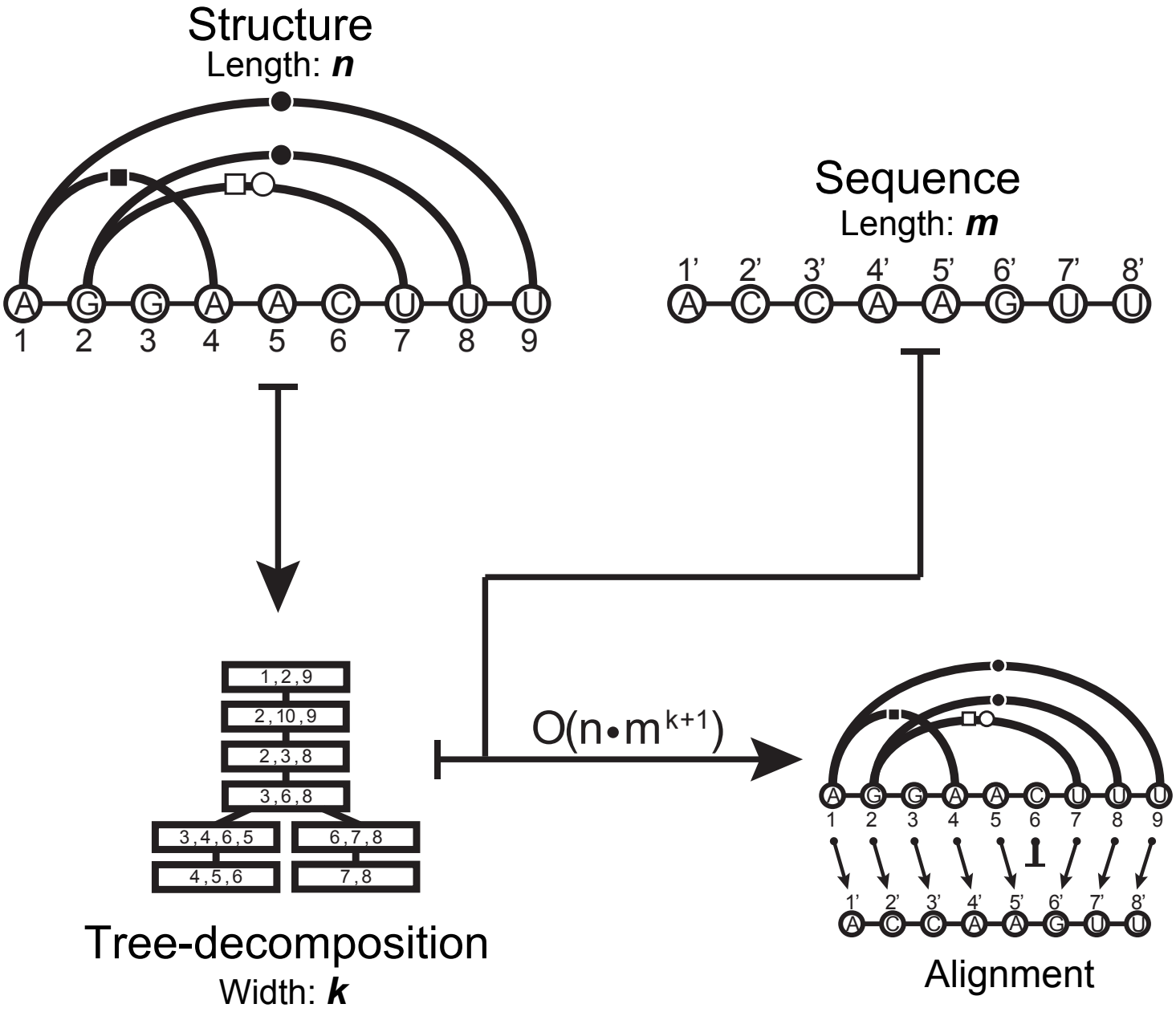
Dynamic programming equation:

$$\text{Cost}(I, f) = \min_{\substack{f'=(\mu',\delta')\in\mathcal{F}|_{X_I} \\ f' \text{ compatible with } f}} \left\{ \phi(X_I, f') + \sum_{s \text{ child of } I} \text{Cost}(s, f'|_{X_{s,I}}) \right\},$$

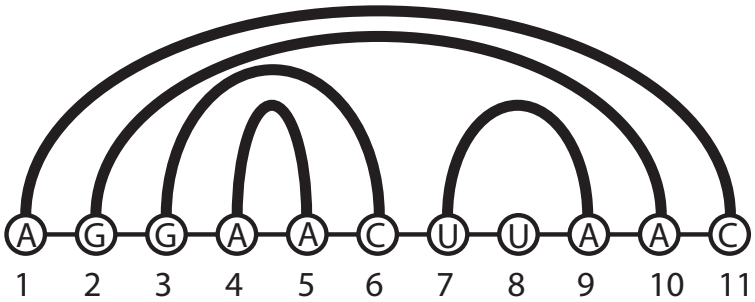
where $\phi(X_I, f')$: local cost contribution of alignment f' to a bag X_I

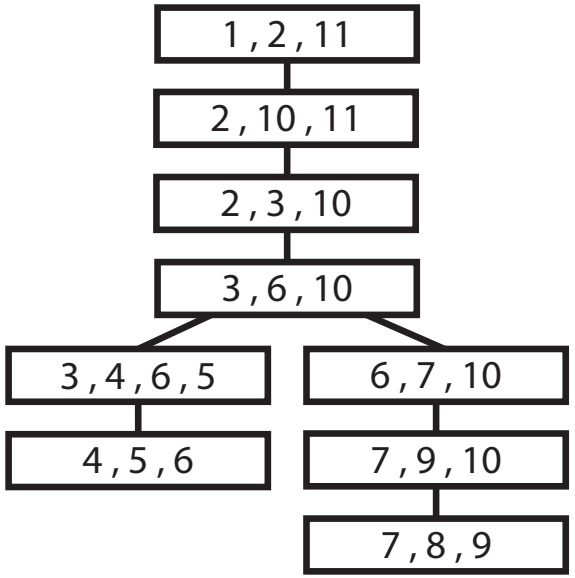
Algorithm: In depth-first order, Compute/Memorize Cost (+Best assignment)

Fixed-parameter tractable algorithm

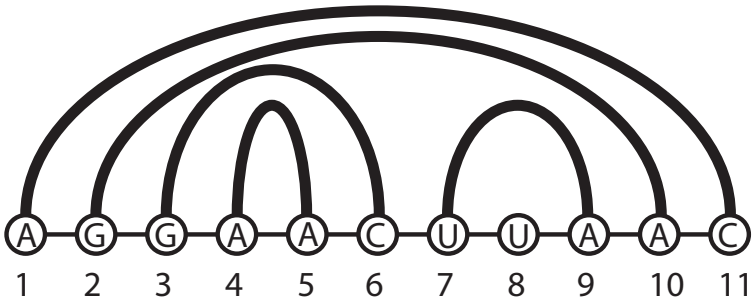


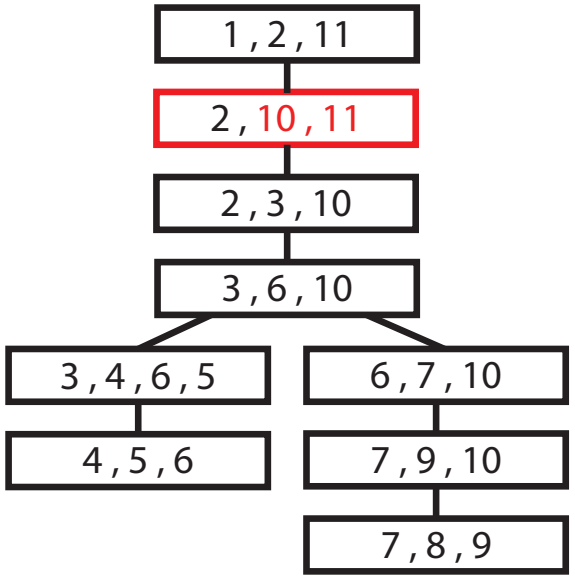
Smooth tree-decomposition



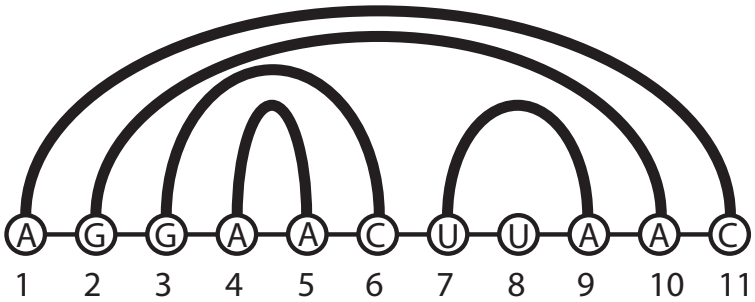


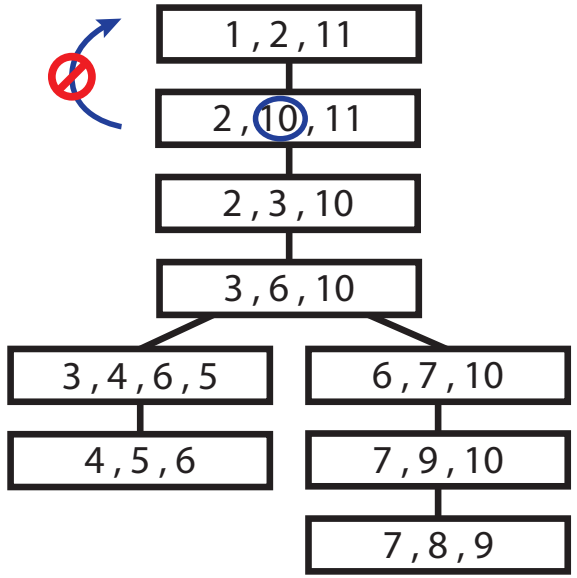
Smooth tree-decomposition



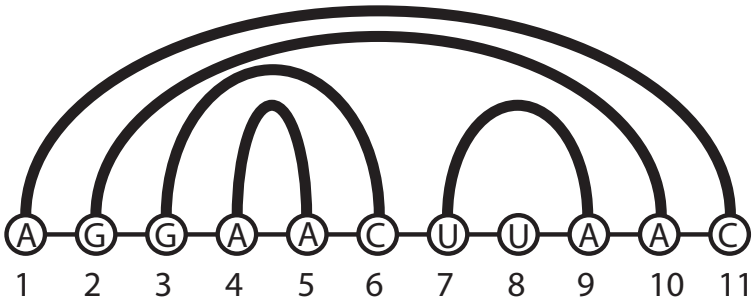


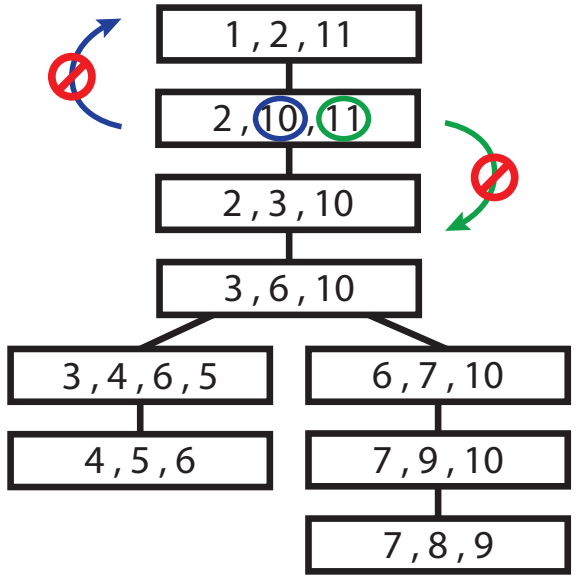
Smooth tree-decomposition



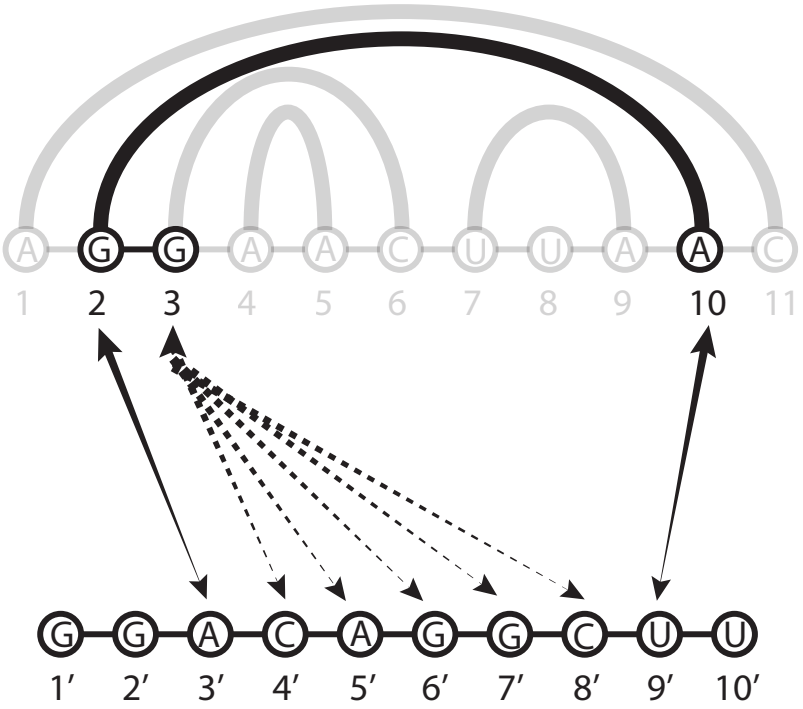


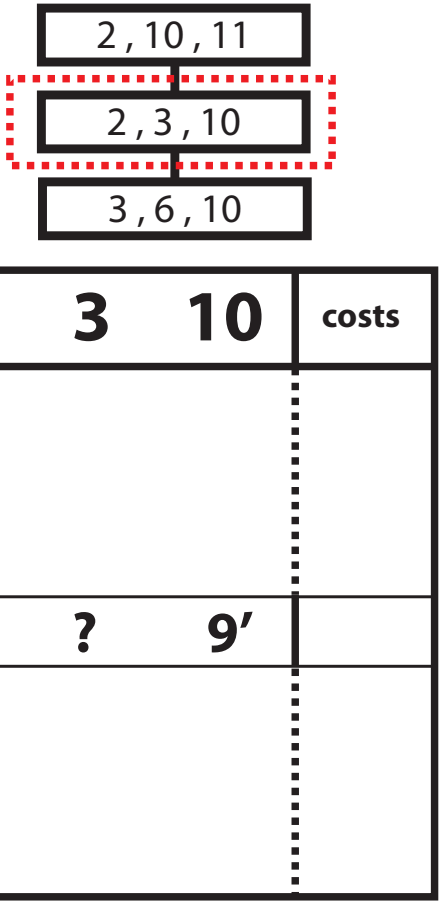
Smooth tree-decomposition



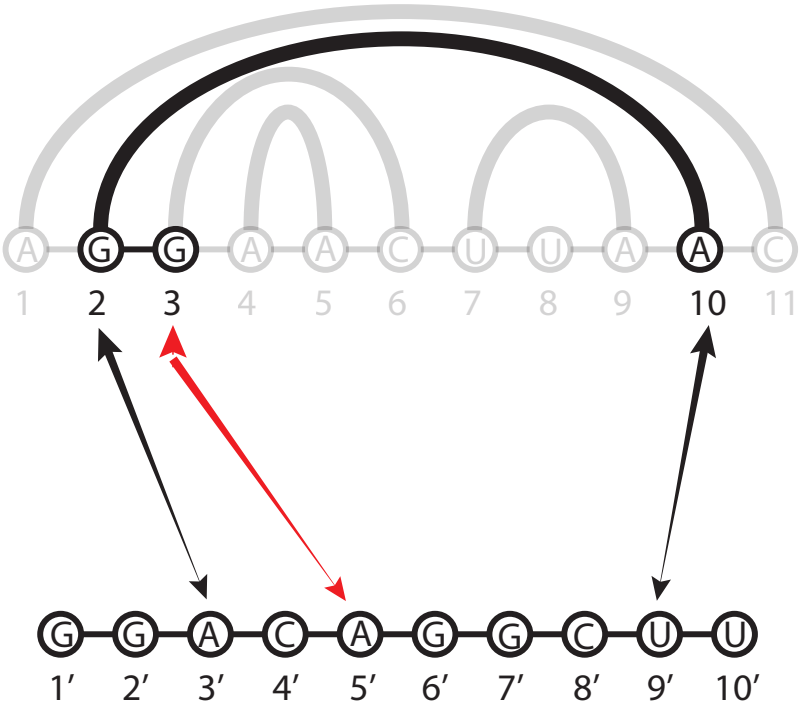


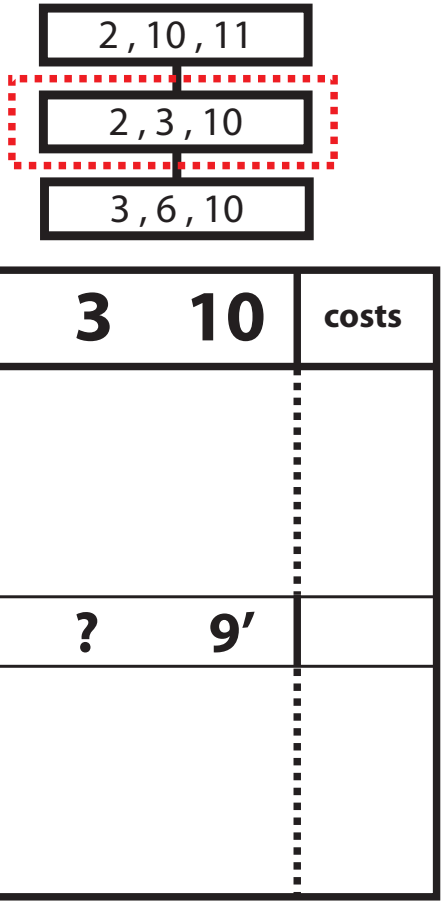
Aligning smooth tree-decomposition



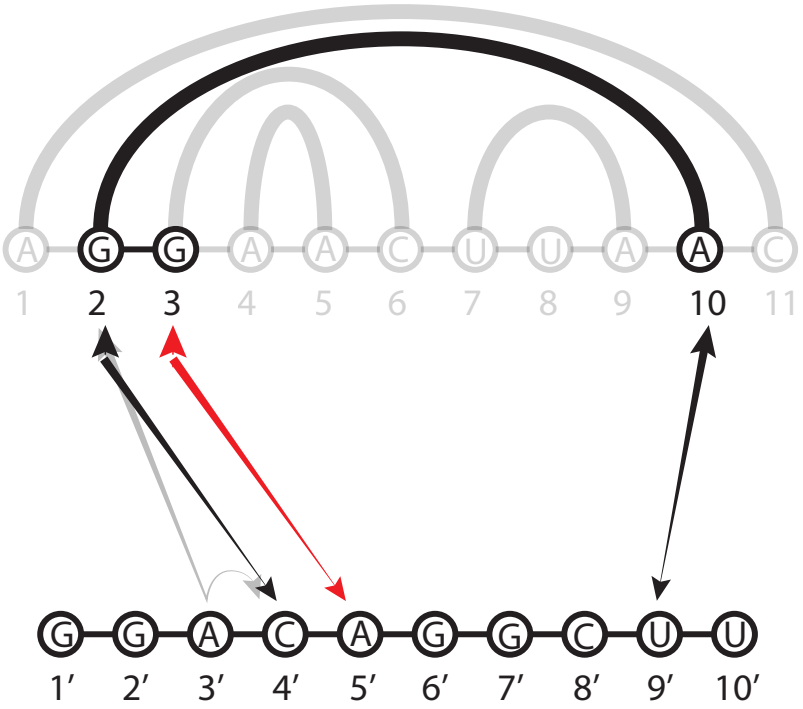


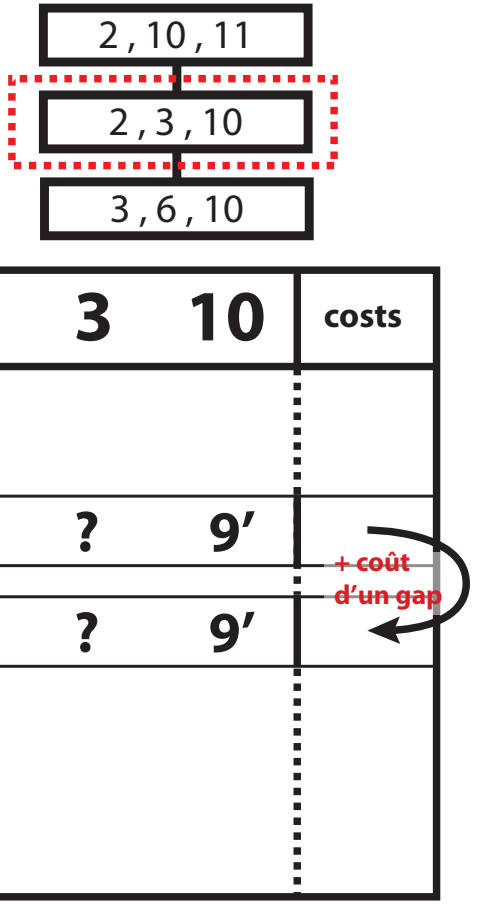
Aligning smooth tree-decomposition



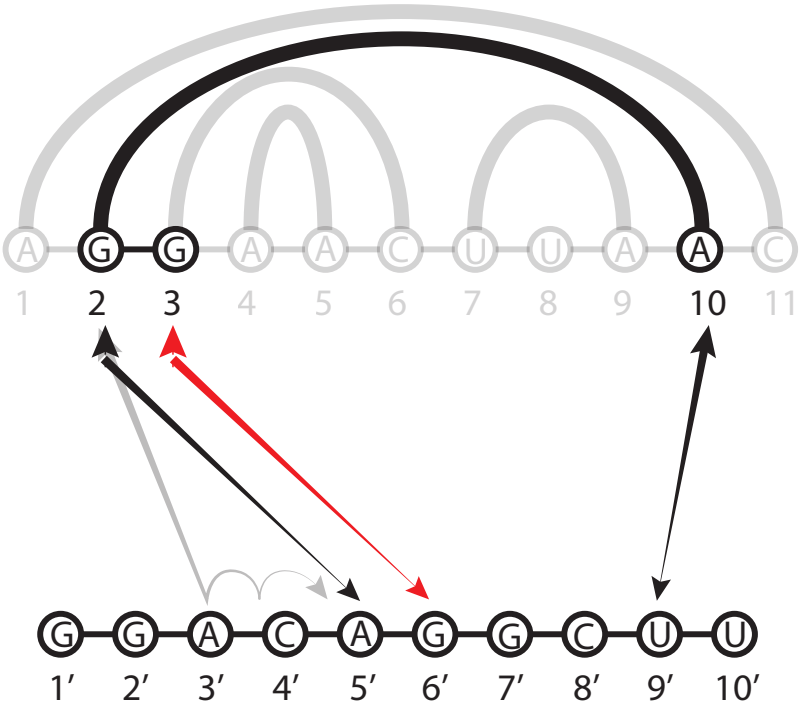


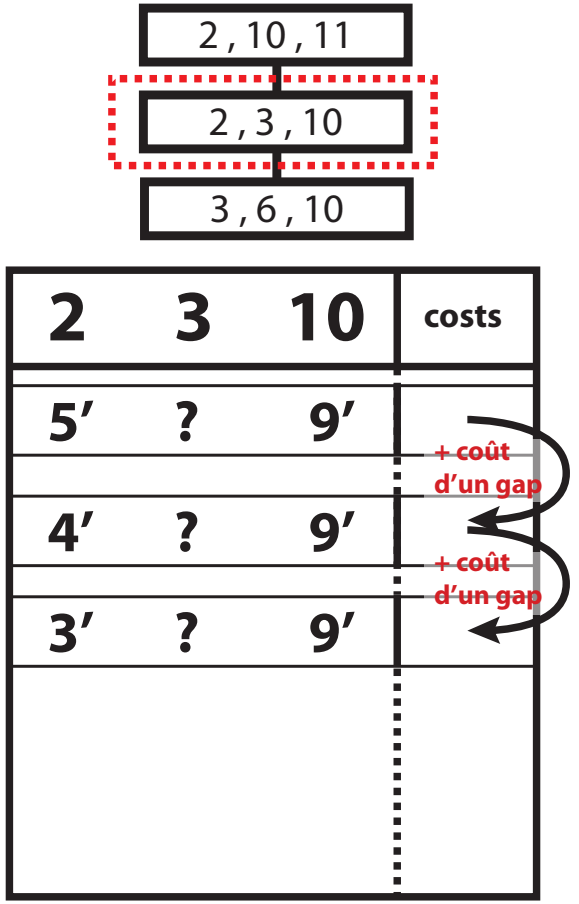
Aligning smooth tree-decomposition





Aligning smooth tree-decomposition





Theorem

- Structure of length n
- Sequence of length m
- **Smooth** tree decomposition of width k (+affine costs)

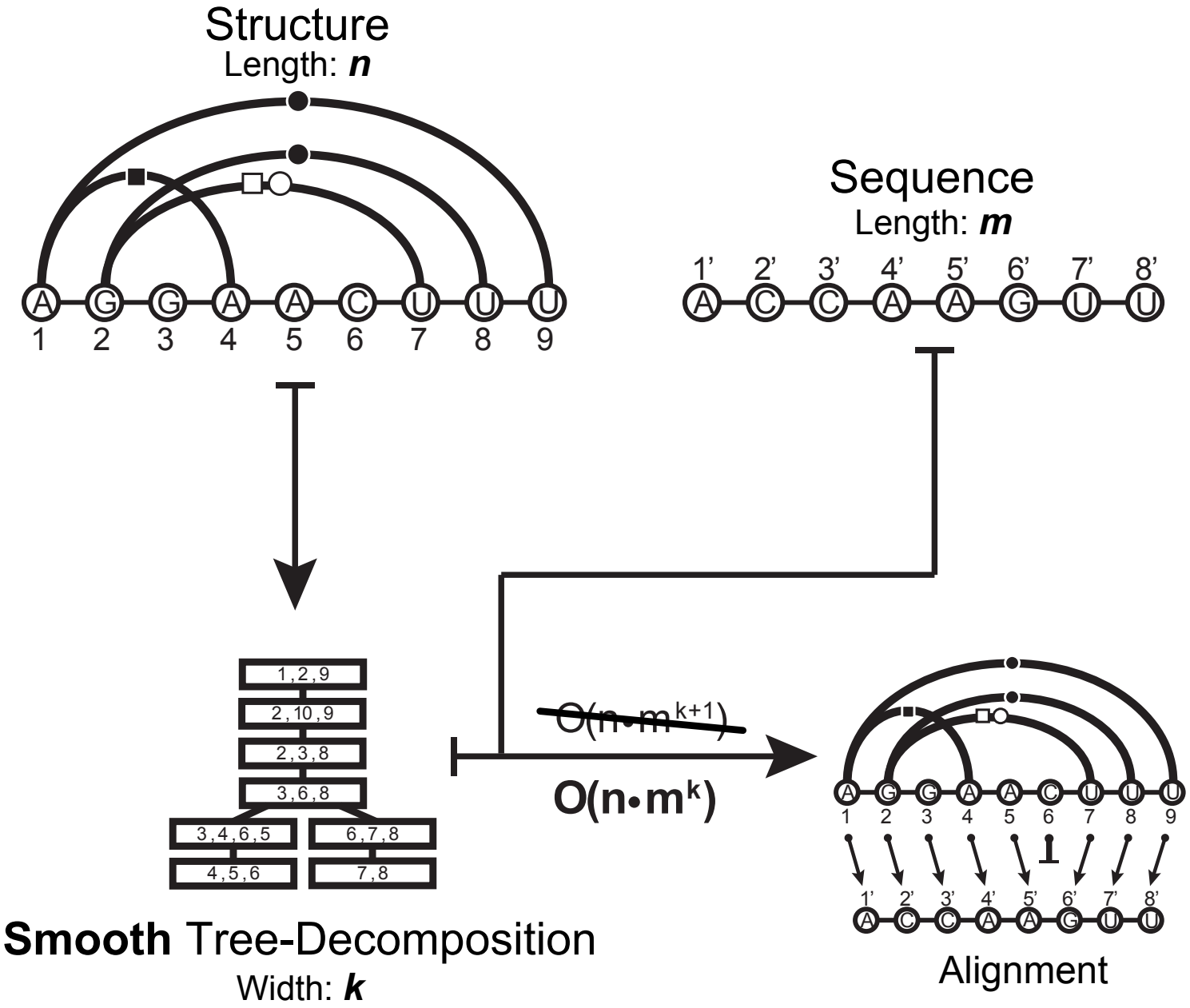
$\Rightarrow$ Best structure-sequence alignment computed in $\mathcal{O}\left(n.m^k\right)$ time/space

$$C_f^l = \min \left\{ \begin{array}{c} \min \quad \Delta_l(f') \\ f'=(\mu',\delta') \in \mathcal{F}|_{X_l} \\ \text{s.t. } f'=f \text{ on } X_{l,r} \\ \alpha_B + \beta_B + D^l_{f''} \end{array} \right. \qquad D_f^l = \min \left\{ \begin{array}{c} \min \quad \Delta_l(f') \\ f'=(\mu',\delta') \in \mathcal{F}|_{X_l} \\ \text{s.t. } f'=f \text{ on } X_{l,r} \\ \alpha_B + D^l_{f''} \end{array} \right.$$

$$C_f^l = \min \left\{ \begin{array}{c} \min \quad \Delta_l(f') \\ f'=(\mu',\delta') \in \mathcal{F}|_{X_l} \\ \text{s.t. } f'=f \text{ on } X_{l,r} \\ \alpha_B + \beta_B + D^l_{f''} \end{array} \right. \qquad D_f^l = min \left\{ \begin{array}{c} \min \quad \Delta_l(f') \\ f'=(\mu',\delta') \in \mathcal{F}|_{X_l} \\ \text{s.t. } f'=f \text{ on } X_{l,r} \\ \alpha_B + D^l_{f''} \end{array} \right.$$

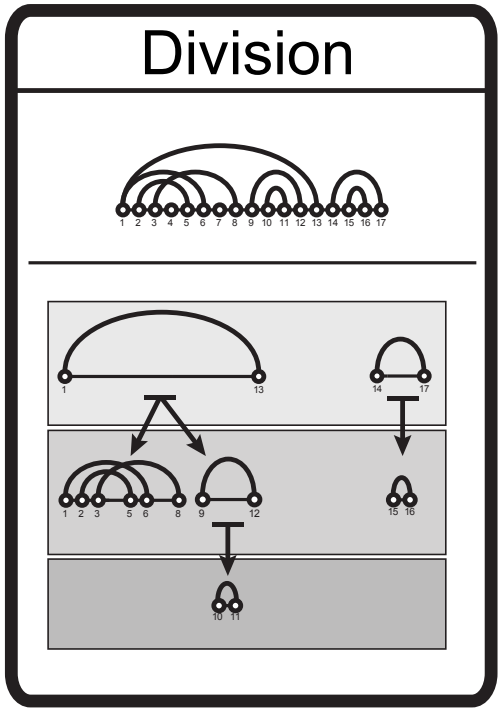
where $\Delta_l(f) := \phi(X_l, f) + \sum_{s \in children(l)} C^s_{f|_{X_{s,l}}}$

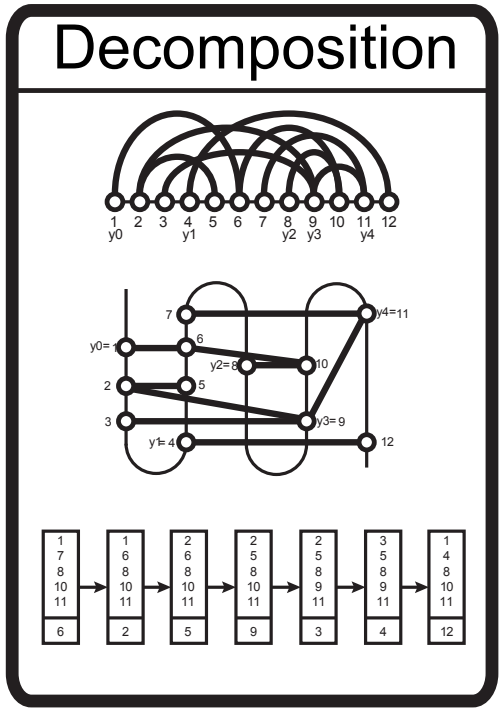
Fixed-parameter tractable algorithm

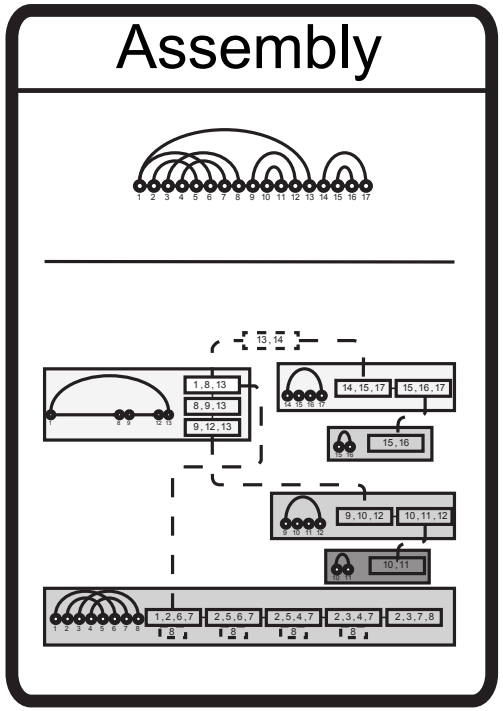


Building tree-decompositions

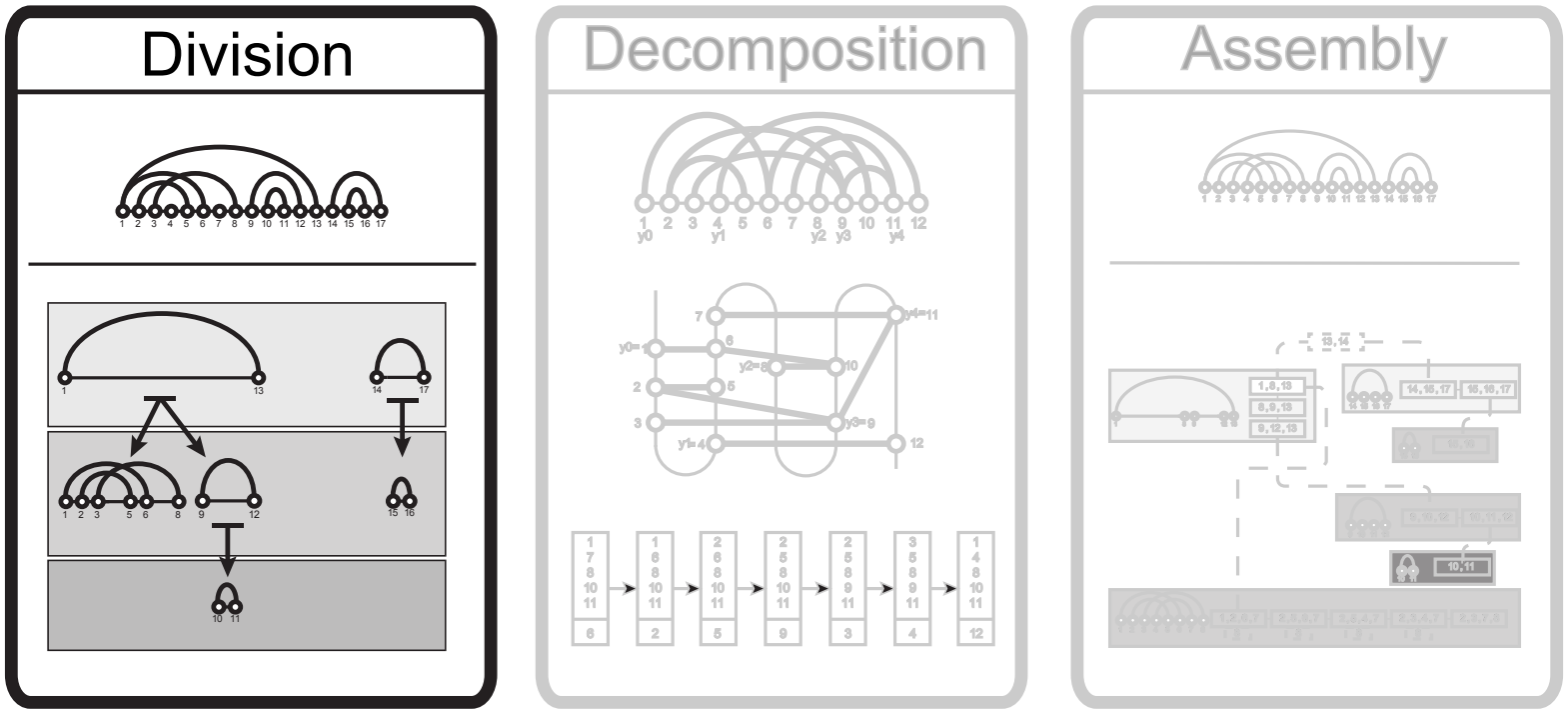
(aka solving an NP-hard problem in three easy steps. . .)





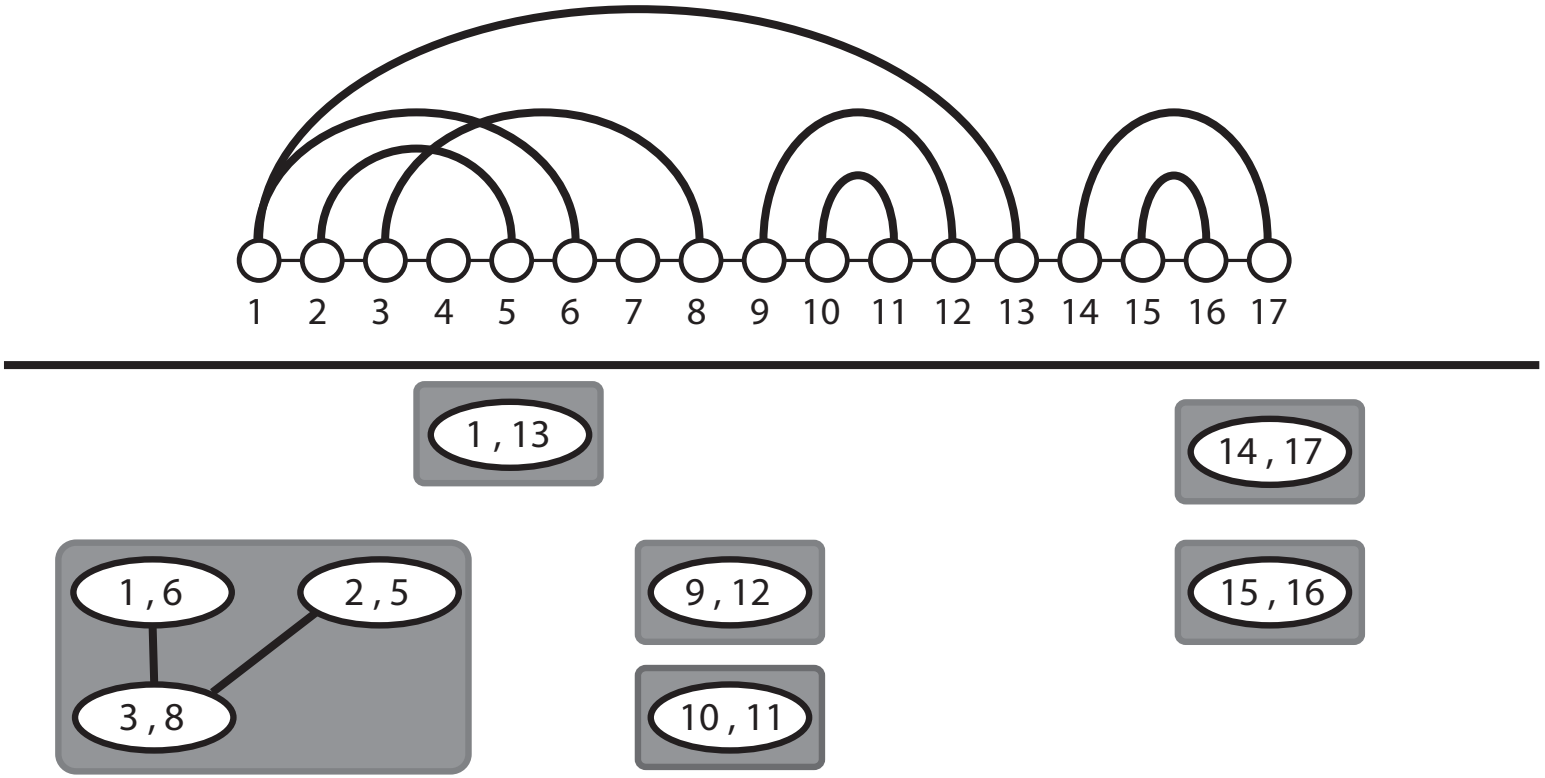


Division de la structure en primitives



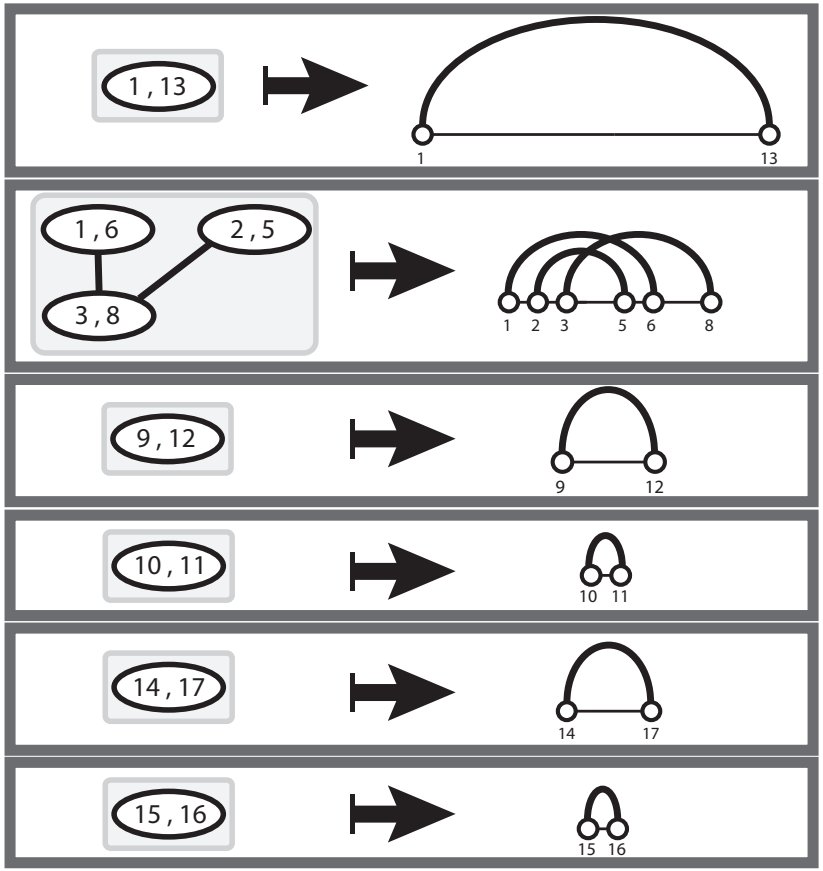
Conflict graph

Vertices = Interactions
Edges = Pairs of crossing interactions



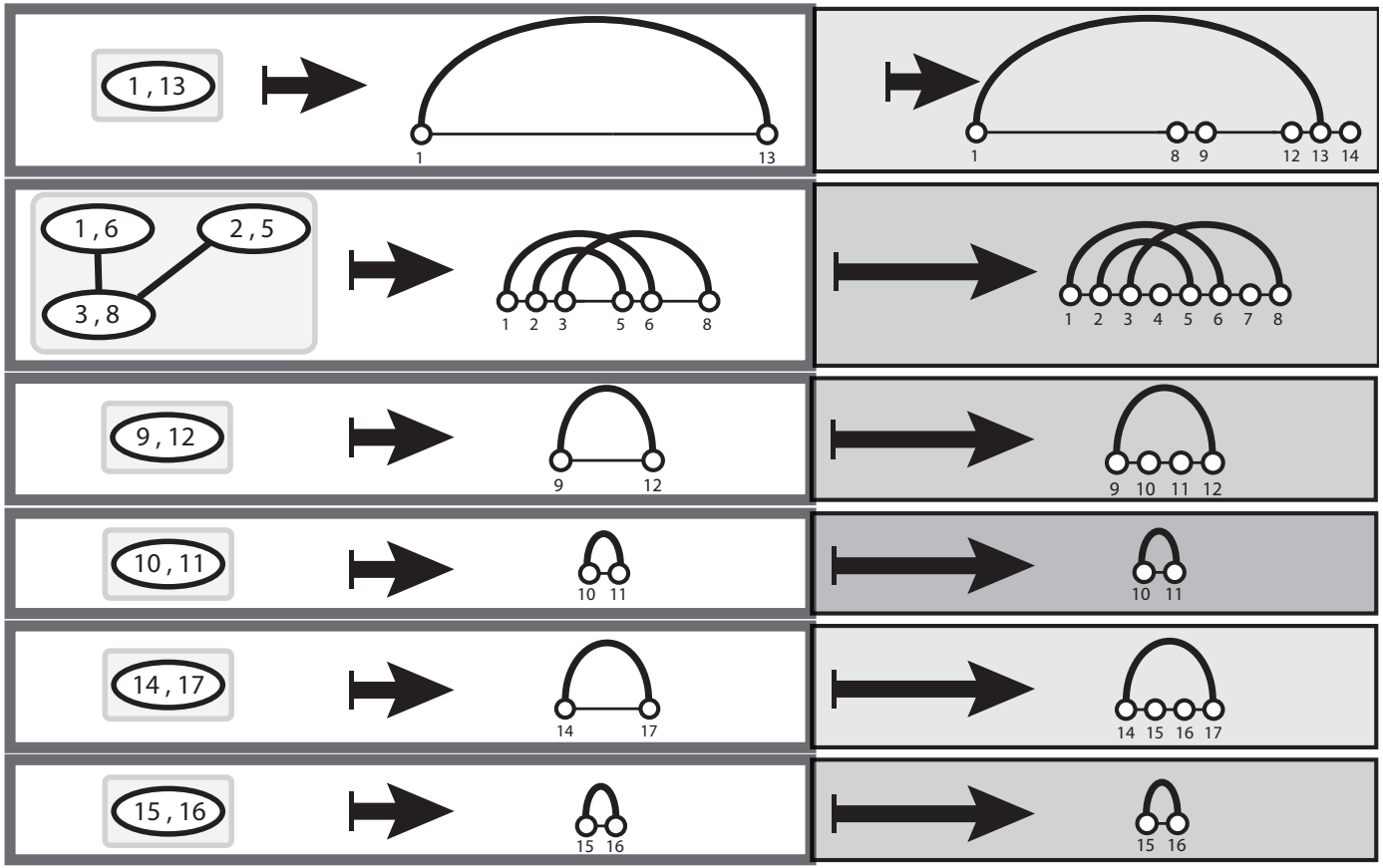
Primitive Structures

Primitives structures $\Leftrightarrow$ Connected components of conflict graph



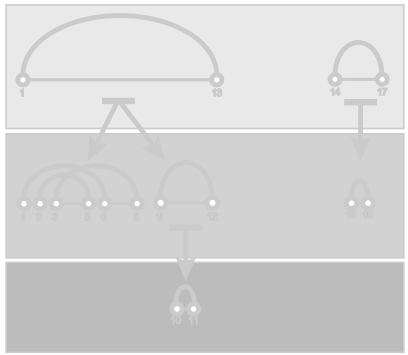
Primitive Structures

Primitives structures $\Leftrightarrow$ Connected components of conflict graph

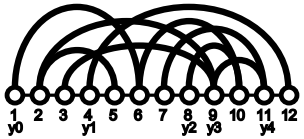


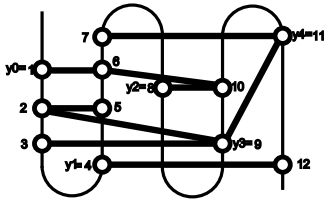
Decomposition through wave-embedding

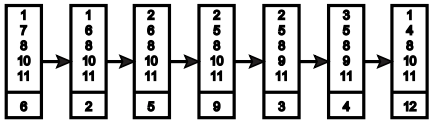
Division



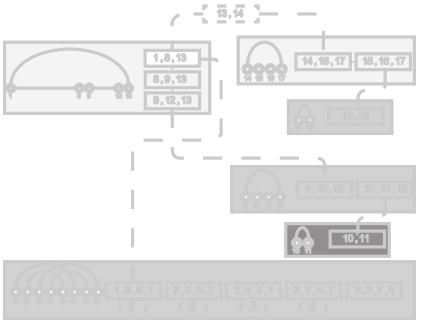
Decomposition







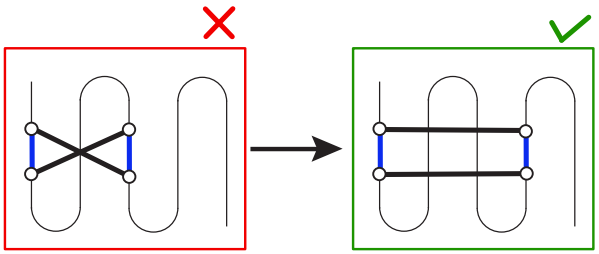
Assembly

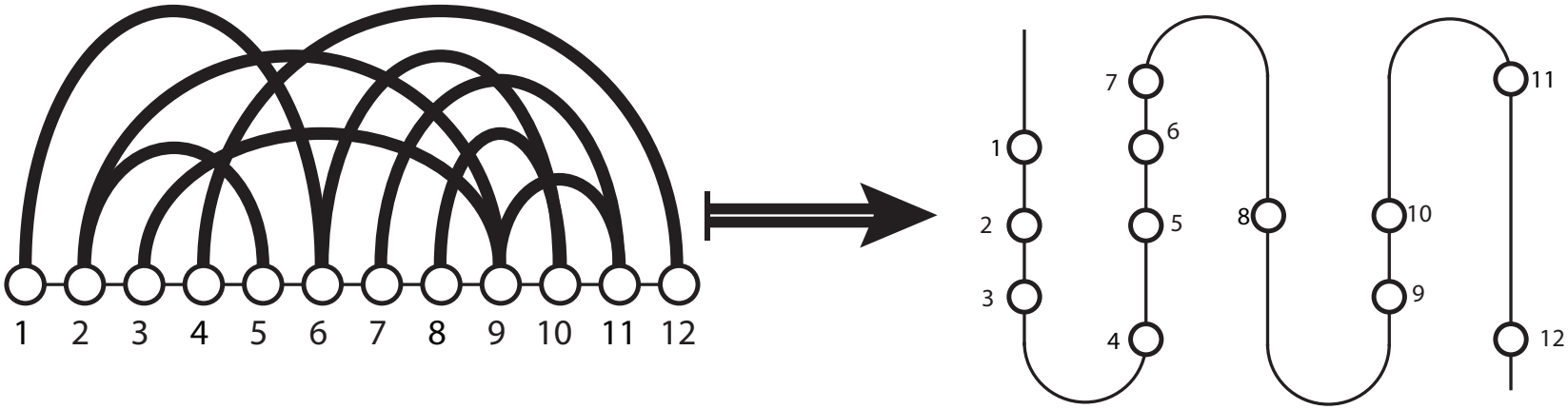


Decomposition through wave-embedding

Goal: Find embedding as waves such that

- Number of stems is minimized
- No twisted alternating cycles ($\neq$ planarity)
- Base-paired positions $\Rightarrow$ Different stems

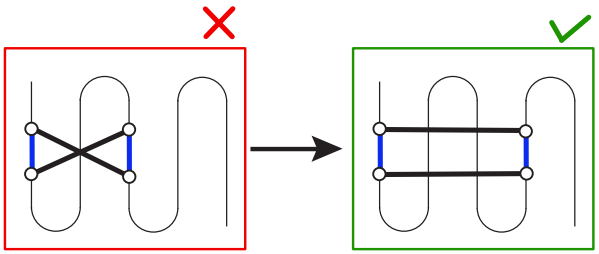


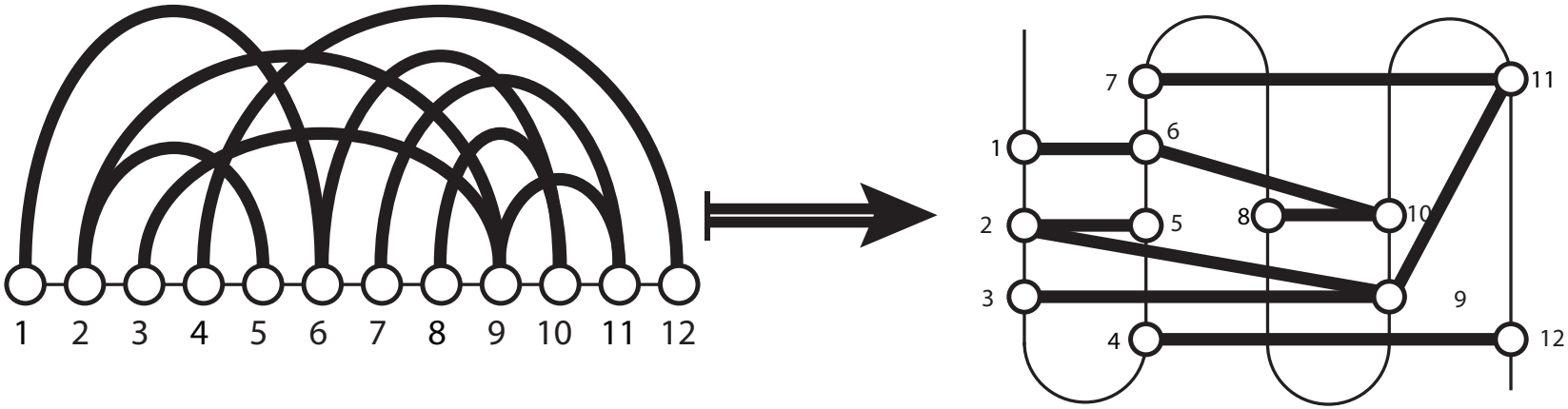


Decomposition through wave-embedding

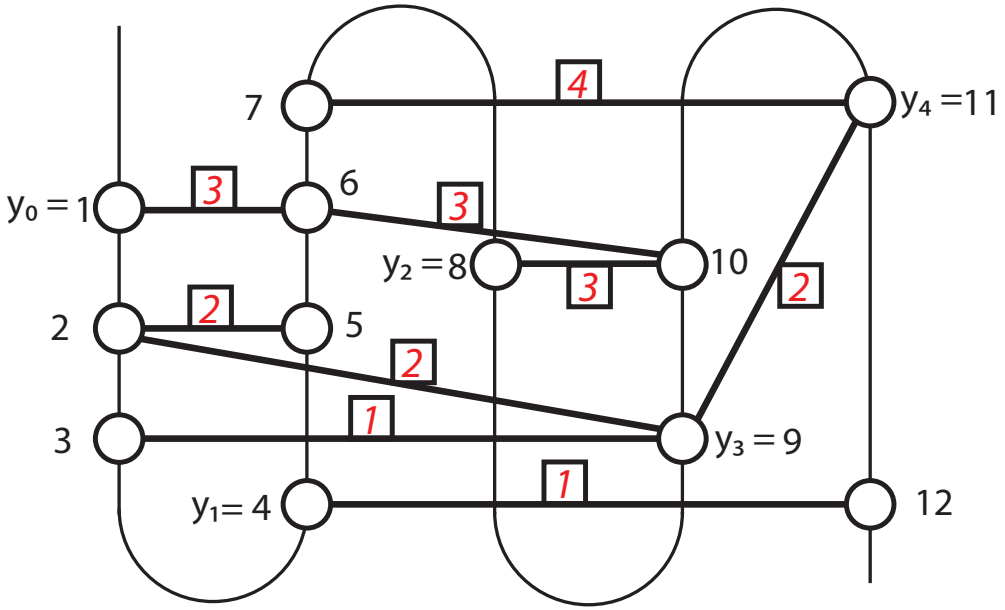
Goal: Find embedding as waves such that

- Number of stems is minimized
- No twisted alternating cycles ($\neq$ planarity)
- Base-paired positions $\Rightarrow$ Different stems





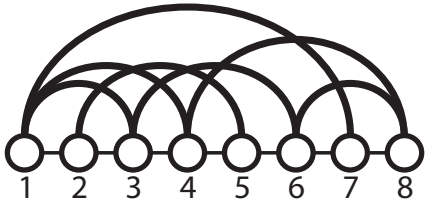
Decomposition through wave-embedding



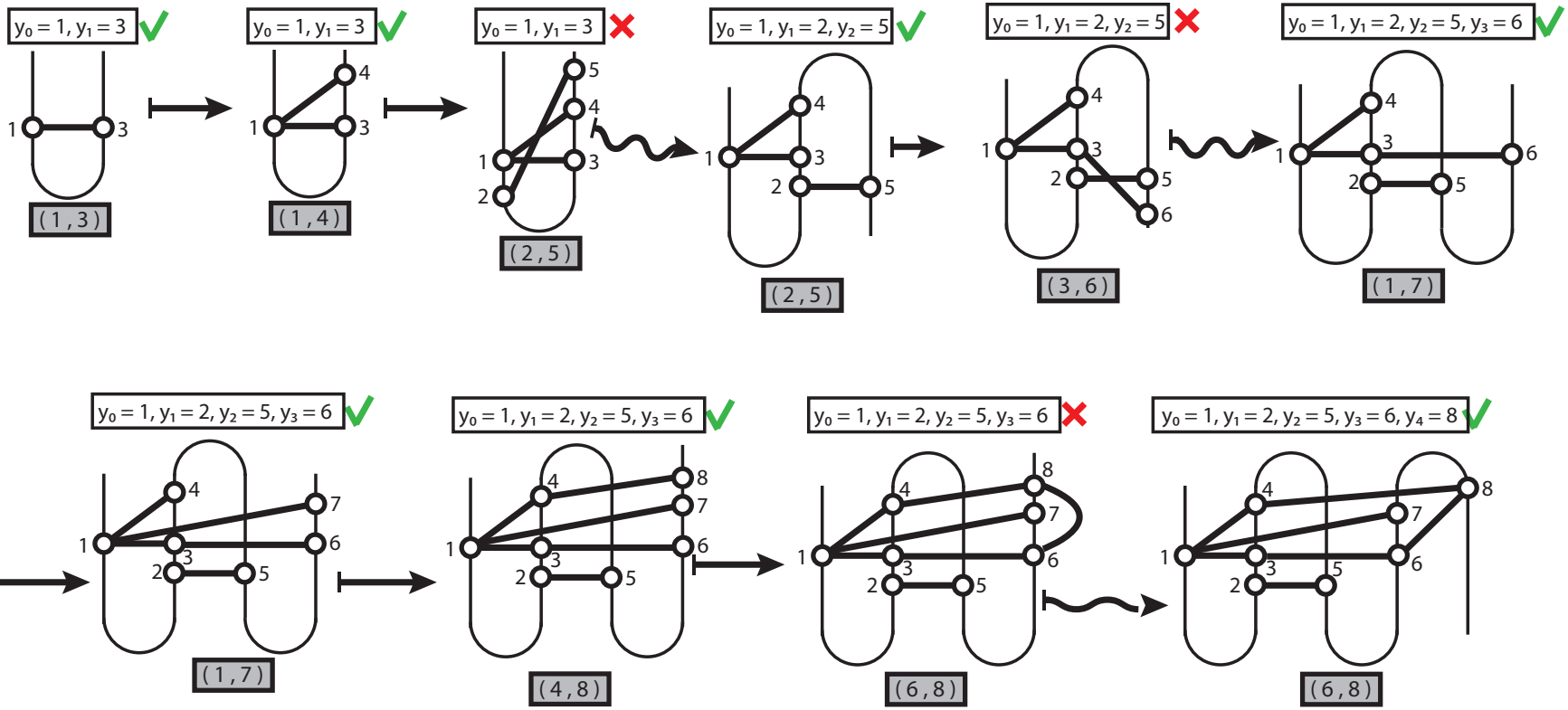
Niveau	4	3	3	3	3	2	2	2	1	1	1	1
Positions	7	1	6	8	10	2	5	11	3	4	9	12

6	1, ***7***, 8, 10, 11
2	***1***, 6, 8, 10, 11
5	2, ***6***, 8, 10, 11
9	2, 5, 8, ***10***, 11
3	***2***, 5, 8, 9, 11
4	3, ***5***, 8, 9, 11
12	3, 4, 8, 9, ***11***

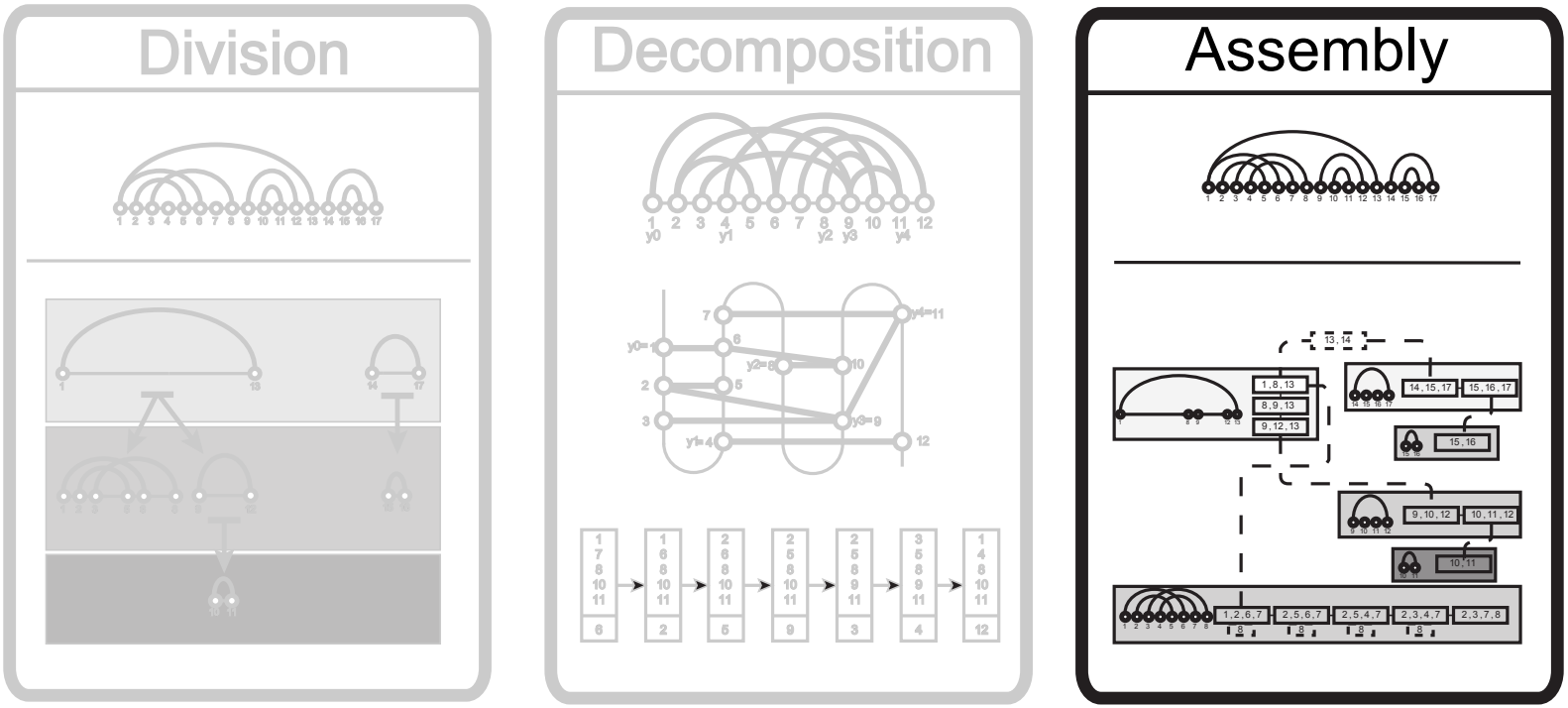
Building a wave embedding: A greedy heuristic



(1,3) (1,4) (2,5) (3,6) (1,7) (4,8) (6,8)

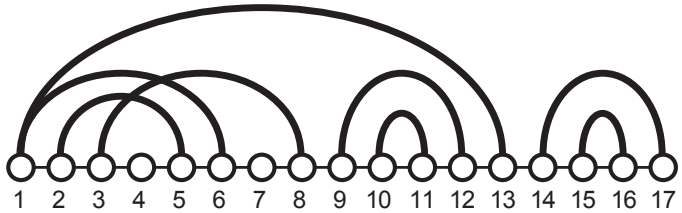


Assembling primitive structures

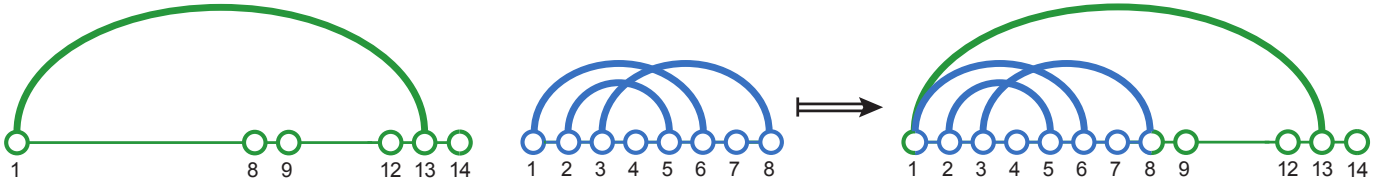


Assembling primitive structures

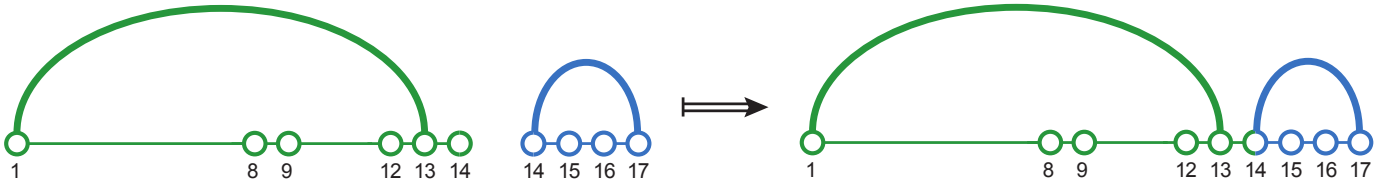
Primitive tree-decompositions are re-assembled hierarchically



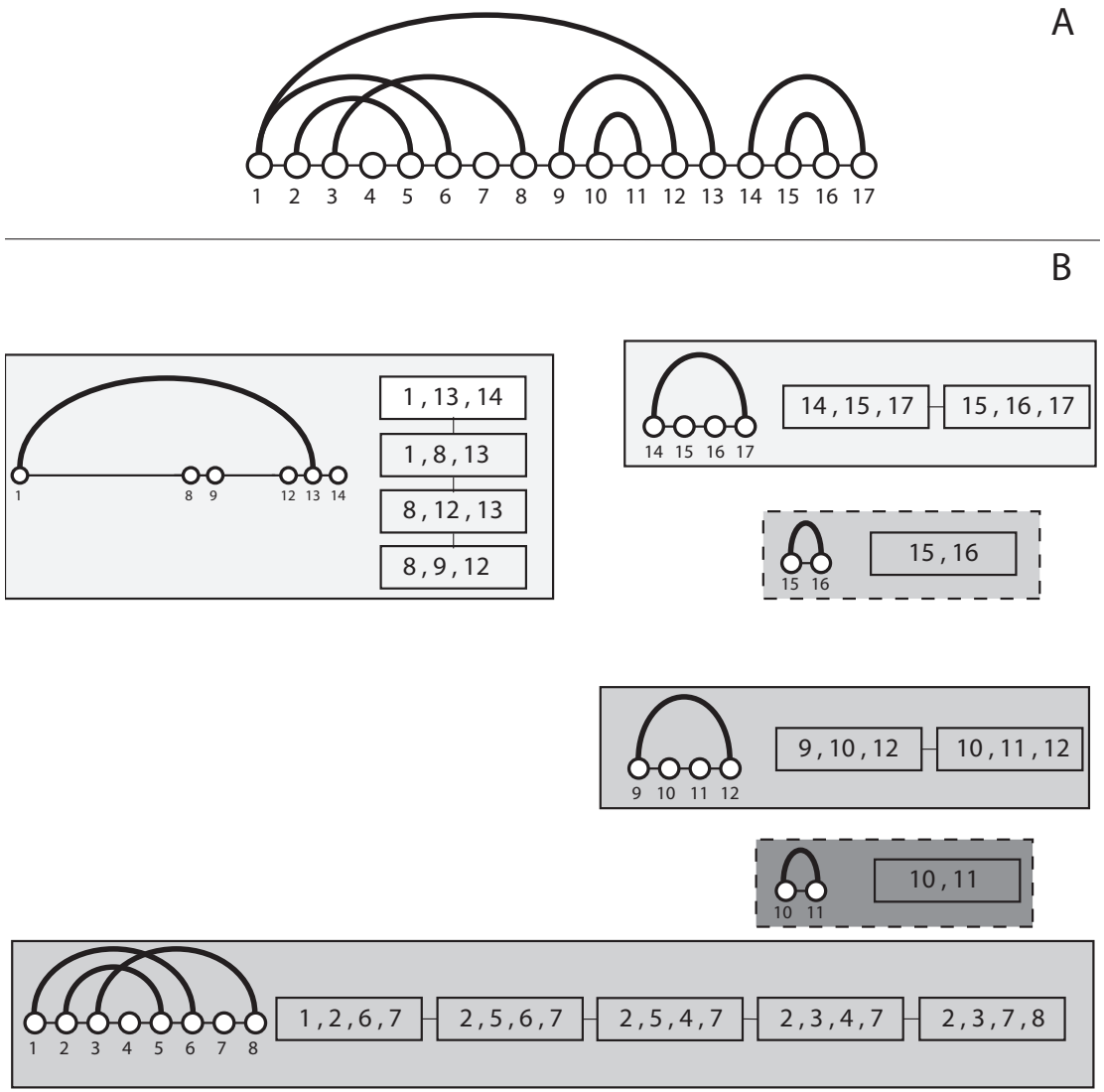
Directly nested



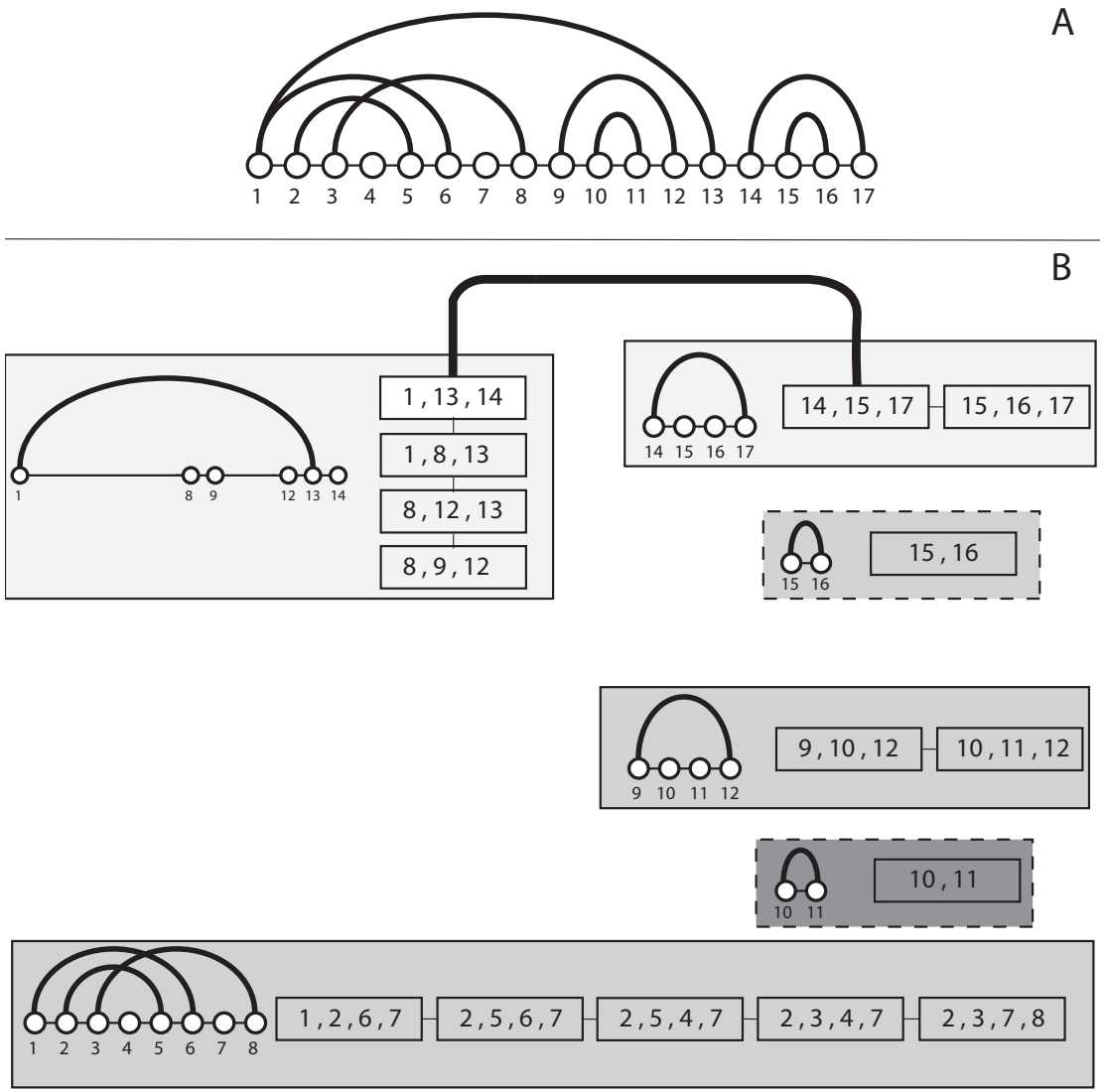
Consecutive



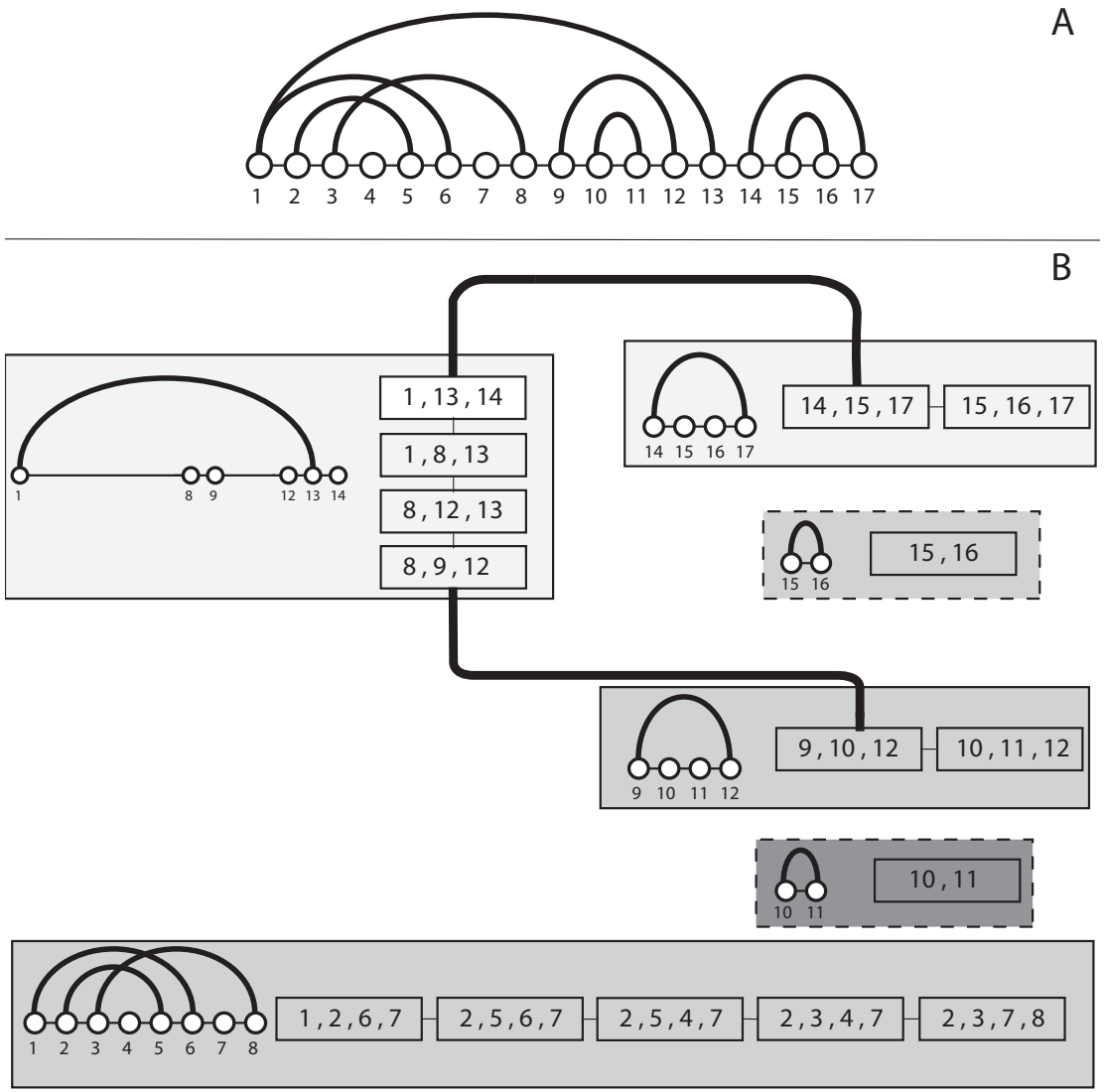
Assembling primitive structures



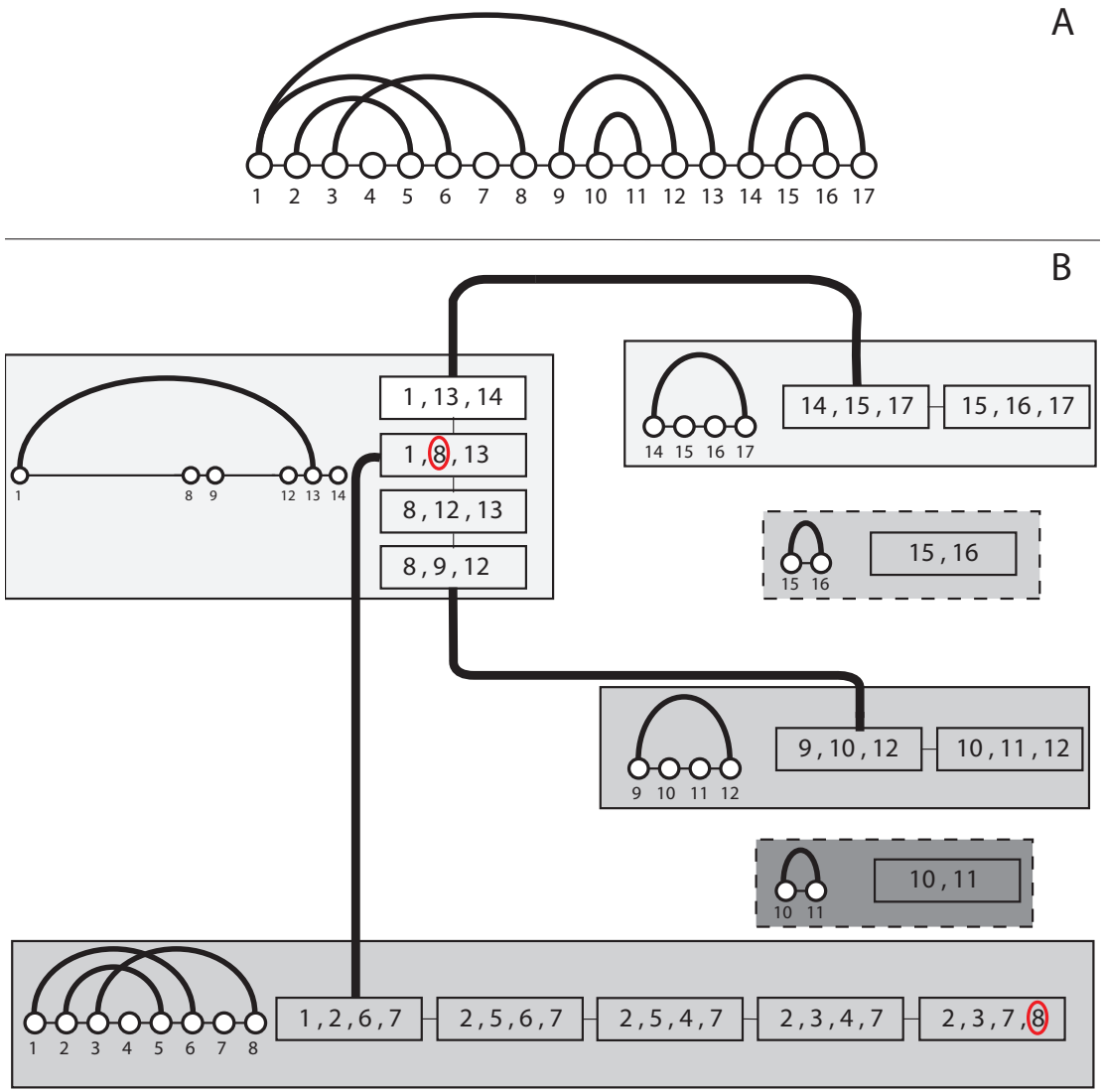
Assembling primitive structures



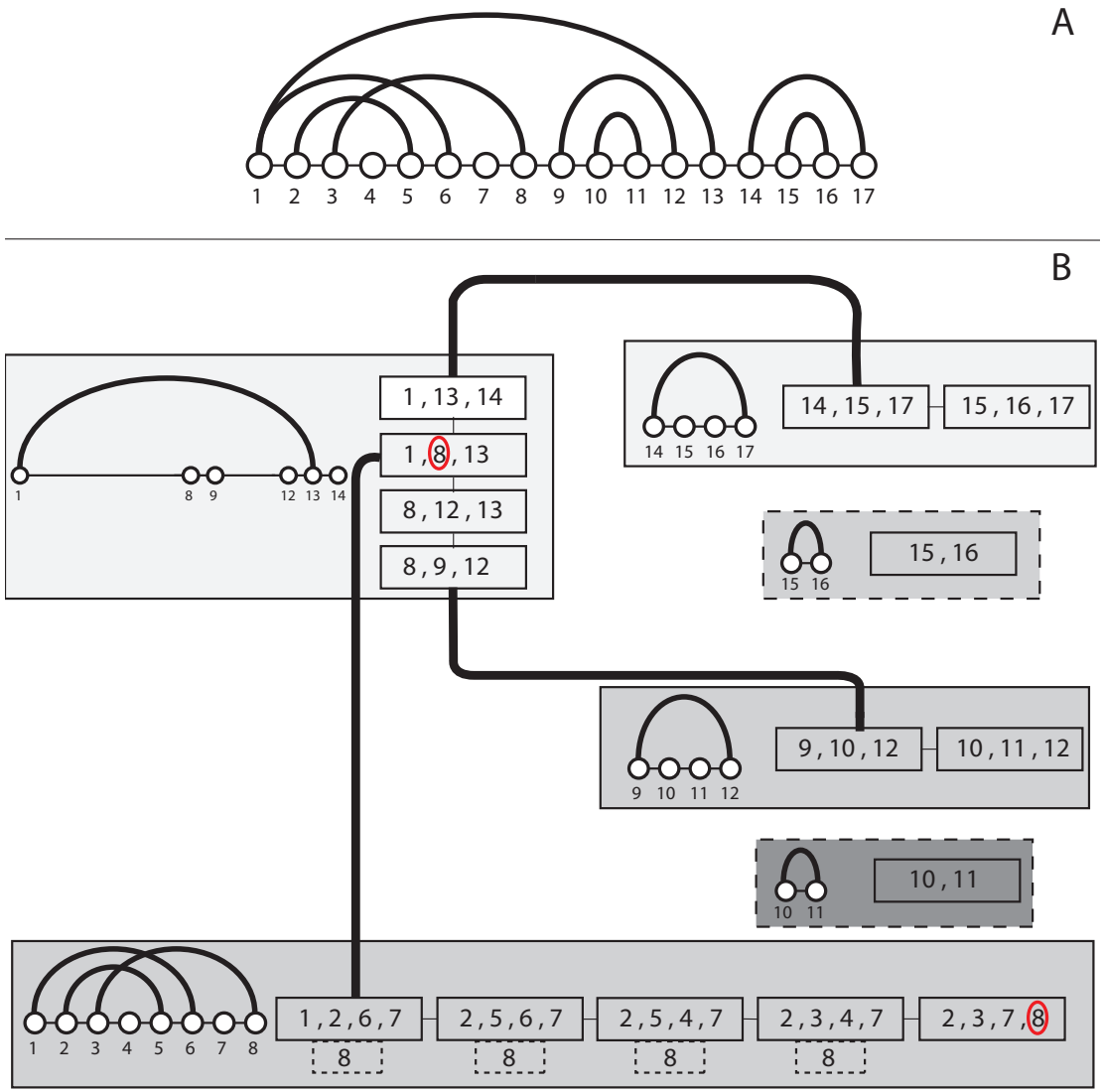
Assembling primitive structures



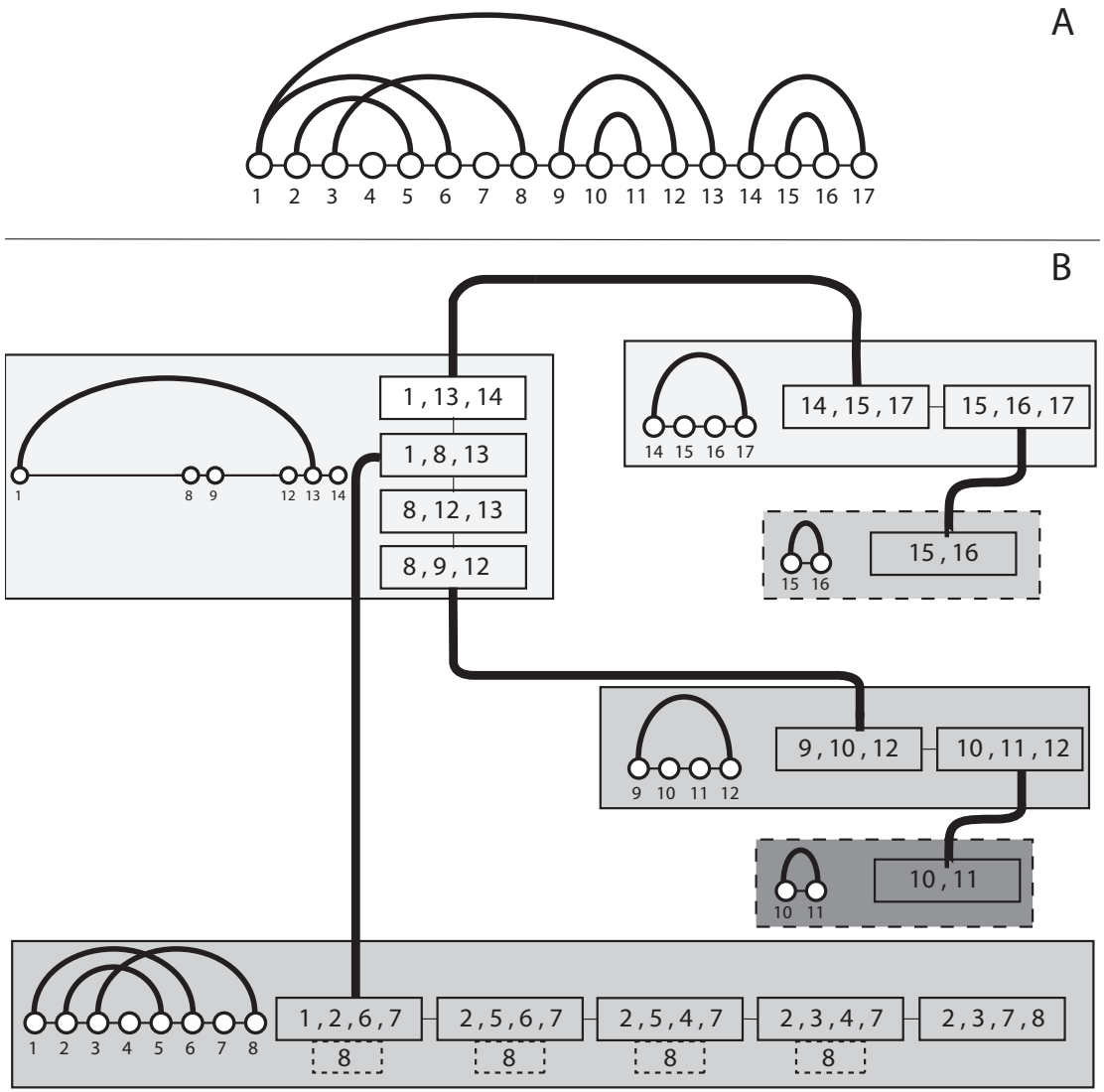
Assembling primitive structures



Assembling primitive structures

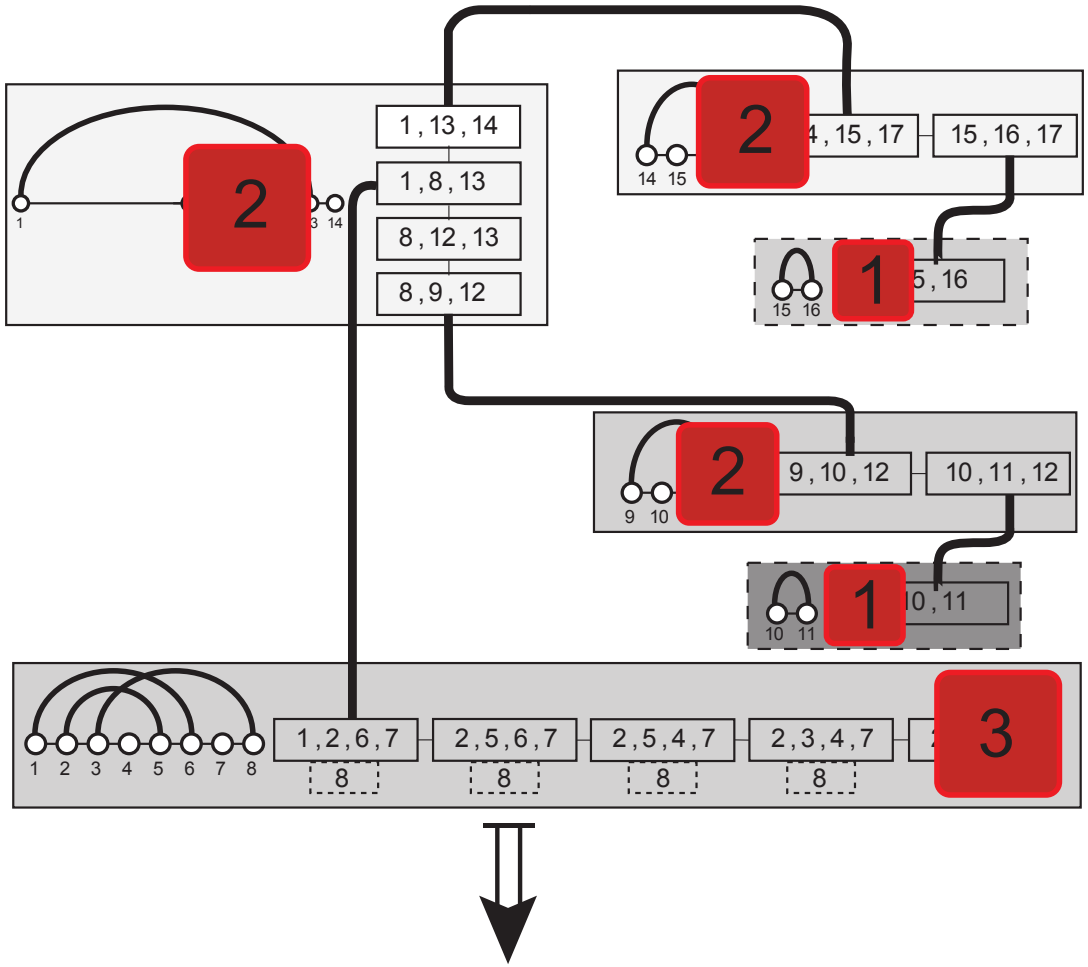


Assembling primitive structures

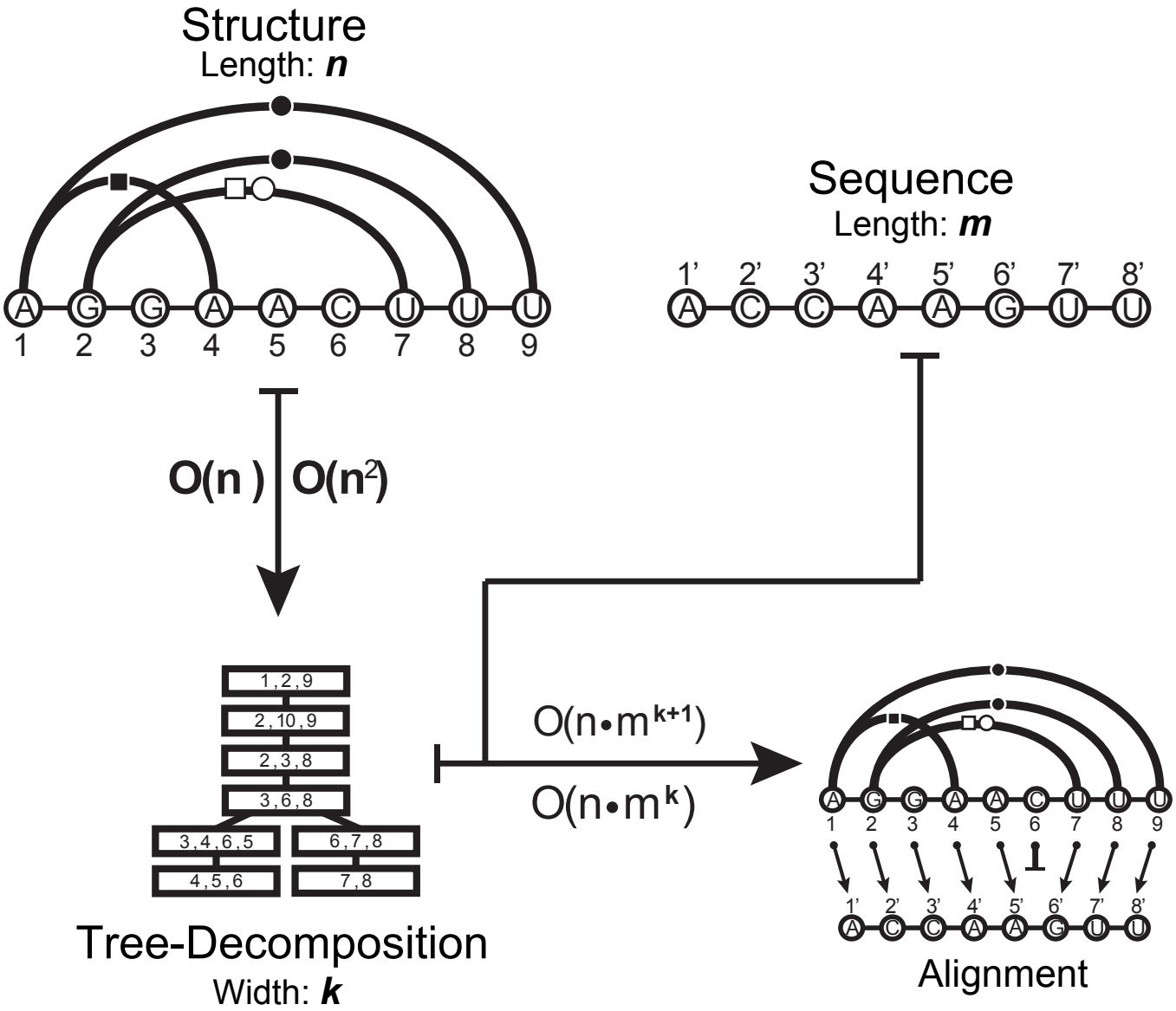


Assembling primitive structures

Final tree width $\leq$ Biggest tree-width of a primitive part $+1$



Final decomposition has width 4


Structure
Length: n
A G G A A C U U U
1 2 3 4 5 6 7 8 9
Sequence
Length: m
1' 2' 3' 4' 5' 6' 7' 8'
A C C A A G U U
O(n)
O(n²)
1,2,9
2,10,9
2,3,8
3,6,8
3,4,6,5
6,7,8
4,5,6
7,8
Tree-Decomposition
Width: k
O(n•m^{k+1})
O(n•m^k)
A G G A A C U U U
1 2 3 4 5 6 7 8 9
1' 2' 3' 4' 5' 6' 7' 8'
A C C A A G U U
Alignment

Message #4

Generic, one-size-fits-all, FPT algorithm based on Dynamic-Programming

Message #5

Same/better complexities than preexisting *ad-hoc* algorithms.
Pratical competitive time/memory consumption (prototype).

Message #6

- (Free!) extension of previous classes.

$\Rightarrow$ Handles previously ignored tertiary motifs/modules.

Message #4

Generic, one-size-fits-all, FPT algorithm based on Dynamic-Programming

Message #5

Same/better complexities than preexisting *ad-hoc* algorithms.
Pratical competitive time/memory consumption (prototype).

Message #6

- (Free!) extension of previous classes.

$\Rightarrow$ Handles previously ignored tertiary motifs/modules.

Message #4

Generic, one-size-fits-all, FPT algorithm based on Dynamic-Programming

Message #5

Same/better complexities than preexisting *ad-hoc* algorithms.
Pratical competitive time/memory consumption (prototype).

Message #6

- (Free!) extension of previous classes.

$\Rightarrow$ Handles previously ignored tertiary motifs/modules.

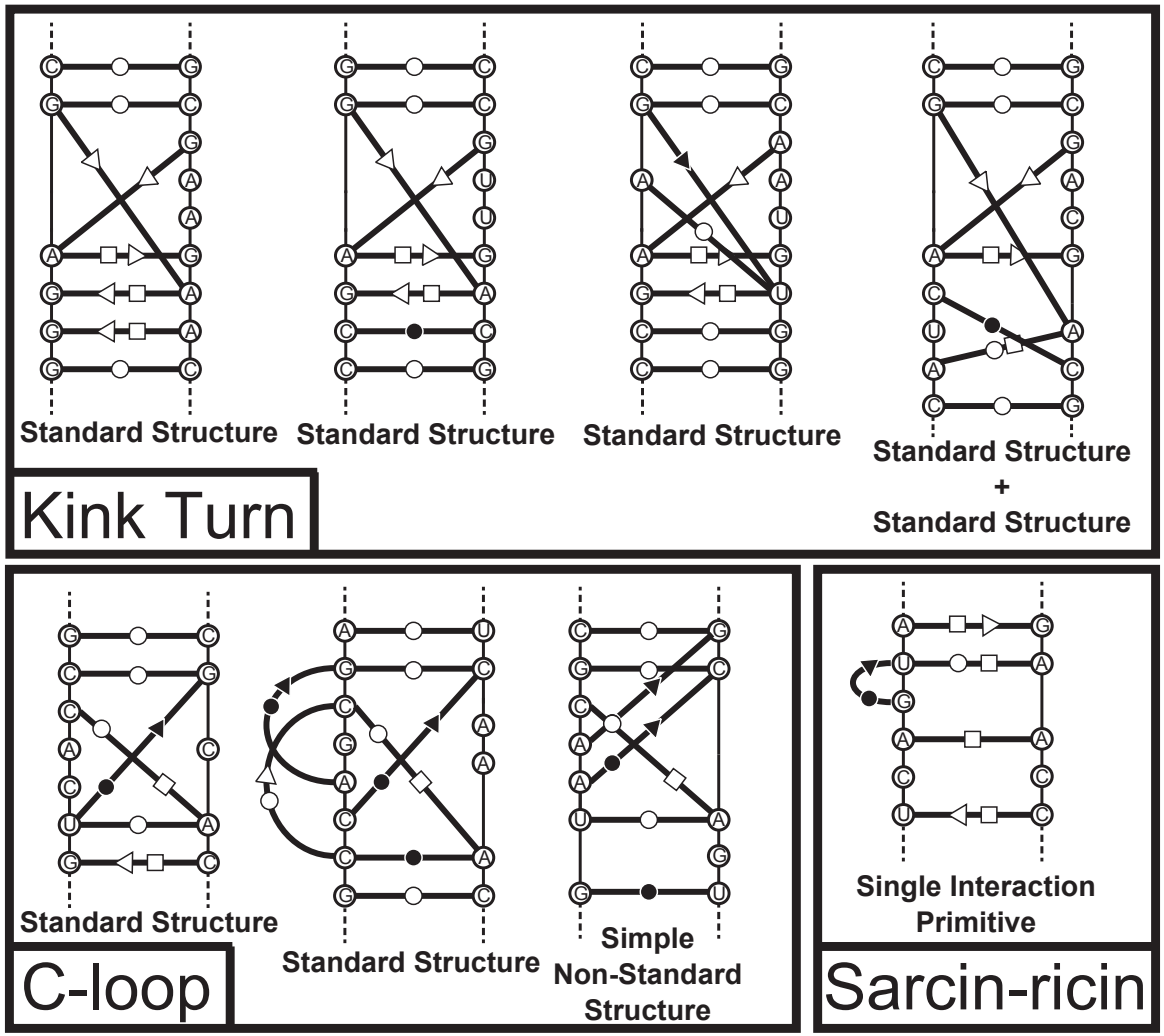
New classes of structures [*Rinaudo et al. 2012*]

Class of Structures	Time comp.	Multiple interactions	Ref.
Recursive Classical Structures	$O(n \cdot m^{k+2})$	$\surd$	–
⌊ Secondary Structures (Pseudoknot-free)	$O(n \cdot m^3)$		*[Jiang et al 02]*
⌊ Embedded Standard Pseudoknots	$O(n \cdot m^{k+1})$		*[Han et al 08]*
⌊ Standard Structures	$O(n \cdot m^k)$	$\surd$	–
⌊ Standard Pseudoknots	$O(n \cdot m^k)$		*[Han et al 08]*
⌊ 2-Level Recursive Simple Non-Standard PKs	$O(n \cdot m^{k+2})$		*[Wong et al 11]*
⌊ Simple Non-Standard Structures	$O(n \cdot m^{k+1})$	$\surd$	–
⌊ Simple Non-Standard Pseudoknots	$O(n \cdot m^{k+1})$		*[Wong et al 11]*
⌊ Extended Triple Helices	$O(n \cdot m^3)$	$\surd$	–
⌊ Triple Helices	$O(n \cdot m^3)$	$\surd$	*[Wong et al 12]*

New classes of structures [*Rinaudo et al. 2012*]

Class of Structures	Time comp.	Multiple interactions	Ref.
Recursive Classical Structures	$O(n \cdot m^{k+2})$	$\surd$	–
Secondary Structures (Pseudoknot-free)	$O(n \cdot m^3)$		*[Jiang et al 02]*
Embedded Standard Pseudoknots	$O(n \cdot m^{k+1})$		*[Han et al 08]*
Standard Structures	$O(n \cdot m^k)$	$\surd$	–
Standard Pseudoknots	$O(n \cdot m^k)$		*[Han et al 08]*
2-Level Recursive Simple Non-Standard PKs	$O(n \cdot m^{k+2})$		*[Wong et al 11]*
Simple Non-Standard Structures	$O(n \cdot m^{k+1})$	$\surd$	–
Simple Non-Standard Pseudoknots	$O(n \cdot m^{k+1})$		*[Wong et al 11]*
Extended Triple Helices	$O(n \cdot m^3)$	$\surd$	–
Triple Helices	$O(n \cdot m^3)$	$\surd$	*[Wong et al 12]*

New classes of structures [*Rinaudo et al. 2012*]



- Recursive Classical Structures ..... $O(n \cdot m^{k+2})$
  - Standard Structures ..... $O(n \cdot m^{k})$
  - Simple Non-Standard Structures ..... $O(n \cdot m^{k+1})$
  - Extended Triple Helices ..... $O(n \cdot m^{3})$

Perspectives

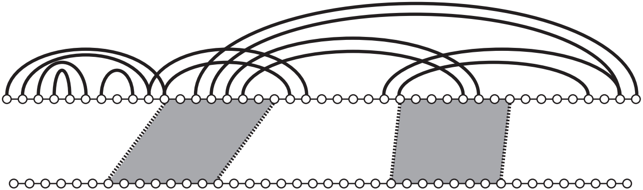
- Finalizing a generic implementation and applications.
- Improving alignment quality:
  - ⇒ Suboptimal alignments: Trivial! (unambiguous DP scheme)
  - ⇒ Good cost functions (e.g. isostericity, Boltz. prob., ML. . .).

AA	**2.30**															
AC	**1.78**	**3.49**														
AG	-0.62	-0.10	**1.32**													
AU	**0.06**	**0.07**	-0.67	**1.79**												
CA	**0.40**	**0.30**	**0.54**	-1.92	**1.78**											
CC	-0.33	-0.61	**0.41**	-1.17	**0.76**	**1.41**										
CG	-1.80	-3.00	**0.90**	-1.40	**0.90**	**0.78**	**1.13**									
CU	**1.43**	**1.75**	-0.93	**0.75**	-0.70	**0.39**	-0.56	**3.38**								
GA	-0.11	**0.36**	-1.71	-0.53	-1.48	-1.32	-2.31	-0.94	**3.25**							
GC	-0.62	**0.36**	-0.90	**0.13**	-1.82	-0.37	-0.96	-0.74	-0.25	**1.03**						
GG	-1.20	-0.43	**0.88**	-1.38	**1.24**	**0.95**	**0.72**	-0.40	**1.33**	-1.03	**2.99**					
GU	**0.49**	**0.32**	-1.34	-0.33	-3.21	-1.64	-2.59	**1.65**	**2.63**	-0.62	**0.60**	**2.93**				
UA	**1.50**	-2.70	-1.16	**0.09**	-0.14	-0.63	-1.16	-1.61	-1.28	-0.91	-2.27	-2.86	**2.06**			
UC	-0.17	**0.57**	-1.47	-3.13	-1.78	-0.82	-3.14	-1.25	-0.53	**0.77**	-1.37	-2.61	-0.77	**3.83**		
UG	-0.41	**0.76**	**0.16**	-2.44	**0.75**	-0.10	-0.14	-1.02	-1.67	-1.51	-0.75	-2.80	**0.05**	**1.15**	**2.24**	
UU	-0.57	-0.82	**0.17**	-0.49	-1.58	-0.64	-0.74	**0.18**	-1.24	-0.70	-0.76	-0.36	-0.54	**1.86**	**2.09**	**2.66**
	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU

A	**1,00**			
C	-1,70	**1,59**		
G	-0,86	-1,58	**1,60**	
U	-1,02	-0,40	-0,94	**1,28**
	A	C	G	U

cWW	Family	LSW 2002 isosteric groups	Updated isosteric groups	Count	CC	UU	CA	UG	UA	AU	GC	CG	GU	AC	UU	CC	UC	CU	AA	GA	AG	AA
CC	cWw	$I_{1.6}$	$I_{1.6}$	8	0.00	2.37	4.49	5.06	5.30	5.39	5.56	5.49	6.25	5.91	4.31	3.02	8.25	7.97	9.77	8.86	8.82	9.05
UU	cwW	$I_{1.6}$	$I_{1.7}$	96	2.37	0.00	2.39	2.89	3.63	3.80	3.94	3.80	5.27	5.21	4.36	4.31	6.46	5.97	8.18	6.96	6.91	6.91
CA	cWW	$I_{1.2}$	$I_{1.2b}$	16	4.49	2.39	0.00	0.80	2.47	2.75	2.78	2.55	4.76	4.93	5.21	5.91	5.30	4.58	6.70	5.14	5.05	4.80
UG	cWW	$I_{1.2}$	$I_{1.2b}$	772	5.06	2.89	0.80	0.00	2.11	2.40	2.39	2.14	4.48	4.76	5.27	6.25	4.59	3.80	6.07	4.44	4.33	4.10
UA	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.30	3.63	2.47	2.11	0.00	0.31	0.34	0.21	2.40	2.75	3.80	5.39	3.57	3.50	4.66	3.67	3.67	4.52
AU	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.39	3.80	2.75	2.40	0.31	0.00	0.21	0.34	2.11	2.47	3.63	5.30	3.50	3.57	4.52	3.67	3.67	4.66
GC	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.56	3.94	2.78	2.39	0.34	0.21	0.00	0.26	2.14	2.55	3.80	5.49	3.39	3.44	4.38	3.49	3.50	4.50
CG	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.49	3.80	2.55	2.14	0.21	0.34	0.26	0.00	2.39	2.78	3.94	5.56	3.44	3.39	4.50	3.50	3.49	4.38
GU	cWW	$i_{1.2}$	$I_{1.2a}$	772	6.25	5.27	4.76	4.48	2.40	2.11	2.14	2.39	0.00	0.80	2.89	5.06	3.80	4.59	4.10	4.33	4.44	6.07
AC	cWW	$i_{1.2}$	$I_{1.2a}$	16	5.91	5.21	4.93	4.76	2.75	2.47	2.55	2.78	0.80	0.00	2.39	4.49	4.58	5.30	4.80	5.05	5.14	6.70
UU	cWw	$I_{1.6}$	$I_{1.7}$	96	4.31	4.36	5.21	5.27	3.80	3.63	3.80	3.94	2.89	2.39	0.00	2.37	5.97	6.46	6.91	6.91	6.96	8.18
CC	cwW	$I_{1.6}$	$I_{1.6}$	8	3.02	4.31	5.91	6.25	5.39	5.30	5.49	5.56	5.06	4.49	2.37	0.00	7.97	8.25	9.05	8.82	8.86	9.77
UC	cWW	$I_{1.5}$	$I_{1.5}$	12	8.25	6.46	5.30	4.59	3.57	3.50	3.39	3.44	3.80	4.58	5.97	7.97	0.00	1.53	2.71	2.25	2.33	3.77
CU	cWW	$I_{1.5}$	$I_{1.5}$	12	7.97	5.97	4.58	3.80	3.50	3.57	3.44	3.39	4.59	5.30	6.46	8.25	1.53	0.00	3.77	2.33	2.25	2.71
AA	cWw	$I_{1.4}$	$I_{1.4}$	3	9.77	8.18	6.70	6.07	4.66	4.52	4.38	4.50	4.10	4.80	6.91	9.05	2.71	3.77	0.00	2.18	2.41	4.52
GA	cWW	$I_{1.3}$	$I_{1.3}$	121	8.86	6.96	5.14	4.44	3.67	3.67	3.49	3.50	4.33	5.05	6.91	8.82	2.25	2.33	2.18	0.00	0.33	2.41
AG	cWW	$I_{1.3}$	$I_{1.3}$	121	8.82	6.91	5.05	4.33	3.67	3.67	3.50	3.49	4.44	5.14	6.96	8.86	2.33	2.25	2.41	0.33	0.00	2.18
AA	cwW	$I_{1.4}$	$I_{1.4}$	3	9.05	6.91	4.80	4.10	4.52	4.66	4.50	4.38	6.07	6.70	8.18	9.77	3.77	2.71	4.52	2.41	2.18	0.00

- Improve runtime/memory ⇒ Heuristics.



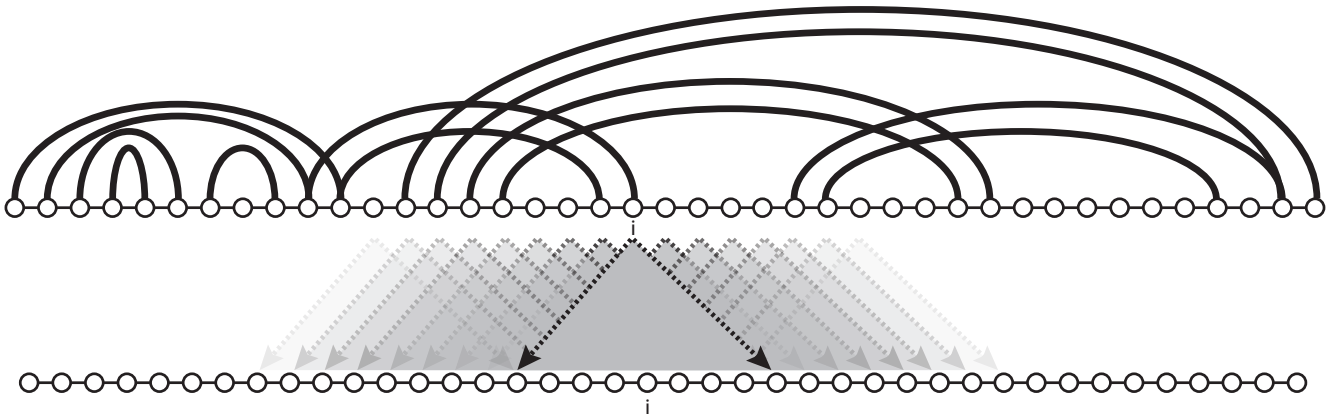
- Finalizing a generic implementation and applications.
- Improving alignment quality:
  - $\Rightarrow$ Suboptimal alignments: Trivial! (unambiguous DP scheme)
  - $\Rightarrow$ Good cost functions (e.g. isostericity, Boltz. prob., ML. . .).

	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU
AA	**2.30**															
AC	**1.78**	**3.49**														
AG	-0.62	-0.10	**1.32**													
AU	**0.06**	**0.07**	-0.67	**1.79**												
CA	**0.40**	**0.30**	**0.54**	-1.92	**1.78**											
CC	-0.33	-0.61	**0.41**	-1.17	**0.76**	**1.41**										
CG	-1.80	-3.00	**0.90**	-1.40	**0.90**	**0.78**	**1.13**									
CU	**1.43**	**1.75**	-0.93	**0.75**	-0.70	**0.39**	-0.56	**3.38**								
GA	-0.11	**0.36**	-1.71	-0.53	-1.48	-1.32	-2.31	-0.94	**3.25**							
GC	-0.62	**0.36**	-0.90	**0.13**	-1.82	-0.37	-0.96	-0.74	-0.25	**1.03**						
GG	-1.20	-0.43	**0.88**	-1.38	**1.24**	**0.95**	**0.72**	-0.40	**1.33**	-1.03	**2.99**					
GU	**0.49**	**0.32**	-1.34	-0.33	-3.21	-1.64	-2.59	**1.65**	**2.63**	-0.62	**0.60**	**2.93**				
UA	**1.50**	-2.70	-1.16	**0.09**	-0.14	-0.63	-1.16	-1.61	-1.28	-0.91	-2.27	-2.86	**2.06**			
UC	-0.17	**0.57**	-1.47	-3.13	-1.78	-0.82	-3.14	-1.25	-0.53	**0.77**	-1.37	-2.61	-0.77	**3.83**		
UG	-0.41	**0.76**	**0.16**	-2.44	**0.75**	-0.10	-0.14	-1.02	-1.67	-1.51	-0.75	-2.80	**0.05**	**1.15**	**2.24**	
UU	-0.57	-0.82	**0.17**	-0.49	-1.58	-0.64	-0.74	**0.18**	-1.24	-0.70	-0.76	-0.36	-0.54	**1.86**	**2.09**	**2.66**

	A	C	G	U
A	**1,00**			
C	-1,70	**1,59**		
G	-0,86	-1,58	**1,60**	
U	-1,02	-0,40	-0,94	**1,28**

cWW	Family	LSW 2002 isosteric groups	Updated isosteric groups	Count	CC	UU	CA	UG	UA	AU	GC	CG	GU	AC	UU	CC	UC	CU	AA	GA	AG	AA
CC	cWw	$I_{1.6}$	$I_{1.6}$	8	0.00	2.37	4.49	5.06	5.30	5.39	5.56	5.49	6.25	5.91	4.31	3.02	7.25	7.97	9.77	8.86	8.82	9.05
UU	cwW	$I_{1.6}$	$I_{1.7}$	96	2.37	0.00	2.39	2.89	3.63	3.80	3.94	3.80	5.27	5.21	4.36	4.31	6.46	5.97	8.18	6.96	6.91	6.91
CA	cWW	$I_{1.2}$	$I_{1.2b}$	16	4.49	2.39	0.00	0.80	2.47	2.75	2.78	2.55	4.76	4.93	5.21	5.91	5.30	4.58	6.70	5.14	5.05	4.80
UG	cWW	$I_{1.2}$	$I_{1.2b}$	772	5.06	2.89	0.80	0.00	2.11	2.40	2.39	2.14	4.48	4.76	5.27	6.25	4.59	3.80	6.07	4.44	4.33	4.10
UA	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.30	3.63	2.47	2.11	0.00	0.31	0.34	0.21	2.40	2.75	3.80	5.39	3.57	3.50	4.66	3.67	3.67	4.52
AU	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.39	3.80	2.75	2.40	0.31	0.00	0.21	0.34	2.11	2.47	3.63	5.30	3.50	3.57	4.52	3.67	3.67	4.66
GC	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.56	3.94	2.78	2.39	0.34	0.21	0.00	0.26	2.14	2.55	3.80	5.49	3.39	3.44	4.38	3.49	3.50	4.50
CG	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.49	3.80	2.55	2.14	0.21	0.34	0.26	0.00	2.39	2.78	3.94	5.56	3.44	3.39	4.50	3.50	3.49	4.38
GU	cWW	$i_{1.2}$	$I_{1.2a}$	772	6.25	5.27	4.76	4.48	2.40	2.11	2.14	2.39	0.00	0.80	2.89	5.06	3.80	4.59	4.10	4.33	4.44	6.07
AC	cWW	$i_{1.2}$	$I_{1.2a}$	16	5.91	5.21	4.93	4.76	2.75	2.47	2.55	2.78	0.80	0.00	2.39	4.49	4.58	5.30	4.80	5.05	5.14	6.70
UU	cWw	$I_{1.6}$	$I_{1.7}$	96	4.31	4.36	5.21	5.27	3.80	3.63	3.80	3.94	2.89	2.39	0.00	2.37	5.97	6.46	6.91	6.91	6.96	8.18
CC	cwW	$I_{1.6}$	$I_{1.6}$	8	3.02	4.31	5.91	6.25	5.39	5.30	5.49	5.56	5.06	4.49	2.37	0.00	7.97	8.25	9.05	8.82	8.86	9.77
UC	cWW	$I_{1.5}$	$I_{1.5}$	12	7.25	6.46	5.30	4.59	3.57	3.50	3.39	3.44	3.80	4.58	5.97	7.97	0.00	1.53	2.71	2.25	2.33	3.77
CU	cWW	$I_{1.5}$	$I_{1.5}$	12	7.97	5.97	4.58	3.80	3.50	3.57	3.44	3.39	4.59	5.30	6.46	8.25	1.53	0.00	3.77	2.33	2.25	2.71
AA	cWw	$I_{1.4}$	$I_{1.4}$	3	9.77	8.18	6.70	6.07	4.66	4.52	4.38	4.50	4.10	4.80	6.91	9.05	2.71	3.77	0.00	2.18	2.41	4.52
GA	cWW	$I_{1.3}$	$I_{1.3}$	121	8.86	6.96	5.14	4.44	3.67	3.67	3.49	3.50	4.33	5.05	6.91	8.82	2.25	2.33	2.18	0.00	0.33	2.41
AG	cWW	$I_{1.3}$	$I_{1.3}$	121	8.82	6.91	5.05	4.33	3.67	3.67	3.50	3.49	4.44	5.14	6.96	8.86	2.33	2.25	2.41	0.33	0.00	2.18
AA	cwW	$I_{1.4}$	$I_{1.4}$	3	9.05	6.91	4.80	4.10	4.52	4.66	4.50	4.38	6.07	6.70	8.18	9.77	3.77	2.71	4.52	2.41	2.18	0.00

- Improve runtime/memory $\Rightarrow$ Heuristics.



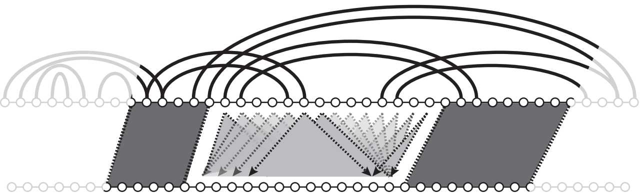
- Finalizing a generic implementation and applications.
- Improving alignment quality:
  - ⇒ Suboptimal alignments: Trivial! (unambiguous DP scheme)
  - ⇒ Good cost functions (e.g. isostericity, Boltz. prob., ML. . .).

AA	**2.30**															
AC	**1.78**	**3.49**														
AG	-0.62	-0.10	**1.32**													
AU	**0.06**	**0.07**	-0.67	**1.79**												
CA	**0.40**	**0.30**	**0.54**	-1.92	**1.78**											
CC	-0.33	-0.61	**0.41**	-1.17	**0.76**	**1.41**										
CG	-1.80	-3.00	**0.90**	-1.40	**0.90**	**0.78**	**1.13**									
CU	**1.43**	**1.75**	-0.93	**0.75**	-0.70	**0.39**	-0.56	**3.38**								
GA	-0.11	**0.36**	-1.71	-0.53	-1.48	-1.32	-2.31	-0.94	**3.25**							
GC	-0.62	**0.36**	-0.90	**0.13**	-1.82	-0.37	-0.96	-0.74	-0.25	**1.03**						
GG	-1.20	-0.43	**0.88**	-1.38	**1.24**	**0.95**	**0.72**	-0.40	**1.33**	-1.03	**2.99**					
GU	**0.49**	**0.32**	-1.34	-0.33	-3.21	-1.64	-2.59	**1.65**	**2.63**	-0.62	**0.60**	**2.93**				
UA	**1.50**	-2.70	-1.16	**0.09**	-0.14	-0.63	-1.16	-1.61	-1.28	-0.91	-2.27	-2.86	**2.06**			
UC	-0.17	**0.57**	-1.47	-3.13	-1.78	-0.82	-3.14	-1.25	-0.53	**0.77**	-1.37	-2.61	-0.77	**3.83**		
UG	-0.41	**0.76**	**0.16**	-2.44	**0.75**	-0.10	-0.14	-1.02	-1.67	-1.51	-0.75	-2.80	**0.05**	**1.15**	**2.24**	
UU	-0.57	-0.82	**0.17**	-0.49	-1.58	-0.64	-0.74	**0.18**	-1.24	-0.70	-0.76	-0.36	-0.54	**1.86**	**2.09**	**2.66**
	AA	AC	AG	AU	CA	CC	CG	CU	GA	GC	GG	GU	UA	UC	UG	UU

A	**1,00**			
C	-1,70	**1,59**		
G	-0,86	-1,58	**1,60**	
U	-1,02	-0,40	-0,94	**1,28**
	A	C	G	U

cWW	Family	LSW 2002 isosteric groups	Updated isosteric groups	Count	CC	UU	CA	UG	UA	AU	GC	CG	GU	AC	UU	CC	UC	CU	AA	GA	AG	AA
CC	cWw	$I_{1.6}$	$I_{1.6}$	8	0.00	2.37	4.49	5.06	5.30	5.39	5.56	5.49	6.25	5.91	4.31	3.02	8.25	7.97	9.77	8.86	8.82	9.05
UU	cwW	$I_{1.6}$	$I_{1.7}$	96	2.37	0.00	2.39	2.89	3.63	3.80	3.94	3.80	5.27	5.21	4.36	4.31	6.46	5.97	8.18	6.96	6.91	6.91
CA	cWW	$I_{1.2}$	$I_{1.2b}$	16	4.49	2.39	0.00	0.80	2.47	2.75	2.78	2.55	4.76	4.93	5.21	5.91	5.30	4.58	6.70	5.14	5.05	4.80
UG	cWW	$I_{1.2}$	$I_{1.2b}$	772	5.06	2.89	0.80	0.00	2.11	2.40	2.39	2.14	4.48	4.76	5.27	6.25	4.59	3.80	6.07	4.44	4.33	4.10
UA	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.30	3.63	2.47	2.11	0.00	0.31	0.34	0.21	2.40	2.75	3.80	5.39	3.57	3.50	4.66	3.67	3.67	4.52
AU	cWW	$I_{1.1}$	$I_{1.1}$	2410	5.39	3.80	2.75	2.40	0.31	0.00	0.21	0.34	2.11	2.47	3.63	5.30	3.50	3.57	4.52	3.67	3.67	4.66
GC	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.56	3.94	2.78	2.39	0.34	0.21	0.00	0.26	2.14	2.55	3.80	5.49	3.39	3.44	4.38	3.49	3.50	4.50
CG	cWW	$I_{1.1}$	$I_{1.1}$	7222	5.49	3.80	2.55	2.14	0.21	0.34	0.26	0.00	2.39	2.78	3.94	5.56	3.44	3.39	4.50	3.50	3.49	4.38
GU	cWW	$i_{1.2}$	$I_{1.2a}$	772	6.25	5.27	4.76	4.48	2.40	2.11	2.14	2.39	0.00	0.80	2.89	5.06	3.80	4.59	4.10	4.33	4.44	6.07
AC	cWW	$i_{1.2}$	$I_{1.2a}$	16	5.91	5.21	4.93	4.76	2.75	2.47	2.55	2.78	0.80	0.00	2.39	4.49	4.58	5.30	4.80	5.05	5.14	6.70
UU	cWw	$I_{1.6}$	$I_{1.7}$	96	4.31	4.36	5.21	5.27	3.80	3.63	3.80	3.94	2.89	2.39	0.00	2.37	5.97	6.46	6.91	6.91	6.96	8.18
CC	cwW	$I_{1.6}$	$I_{1.6}$	8	3.02	4.31	5.91	6.25	5.39	5.30	5.49	5.56	5.06	4.49	2.37	0.00	7.97	8.25	9.05	8.82	8.86	9.77
UC	cWW	$I_{1.5}$	$I_{1.5}$	12	8.25	6.46	5.30	4.59	3.57	3.50	3.39	3.44	3.80	4.58	5.97	7.97	0.00	1.53	2.71	2.25	2.33	3.77
CU	cWW	$I_{1.5}$	$I_{1.5}$	12	7.97	5.97	4.58	3.80	3.50	3.57	3.44	3.39	4.59	5.30	6.46	8.25	1.53	0.00	3.77	2.33	2.25	2.71
AA	cWw	$I_{1.4}$	$I_{1.4}$	3	9.77	8.18	6.70	6.07	4.66	4.52	4.38	4.50	4.10	4.80	6.91	9.05	2.71	3.77	0.00	2.18	2.41	4.52
GA	cWW	$I_{1.3}$	$I_{1.3}$	121	8.86	6.96	5.14	4.44	3.67	3.67	3.49	3.50	4.33	5.05	6.91	8.82	2.25	2.33	2.18	0.00	0.33	2.41
AG	cWW	$I_{1.3}$	$I_{1.3}$	121	8.82	6.91	5.05	4.33	3.67	3.67	3.50	3.49	4.44	5.14	6.96	8.86	2.33	2.25	2.41	0.33	0.00	2.18
AA	cwW	$I_{1.4}$	$I_{1.4}$	3	9.05	6.91	4.80	4.10	4.52	4.66	4.50	4.38	6.07	6.70	8.18	9.77	3.77	2.71	4.52	2.41	2.18	0.00

- Improve runtime/memory ⇒ Heuristics.



Thanks for listening

(aka hey wake up, it's finally over!)