Alignment of RNA with Structures of Unlimited Complexity

Alessandro Dal Palù, Mathias Möhl, Sebastian Will

WCB 2010, Edinburgh

Alignment of RNA with Structures of Unlimited Complexity

Solving the general problem of RNA sequence-structure alignment with CP

Alignment of RNA with Structures of Unlimited Complexity

Solving the general problem of RNA sequence-structure alignment with CP

Alignment of RNA with Structures of Unlimited Complexity

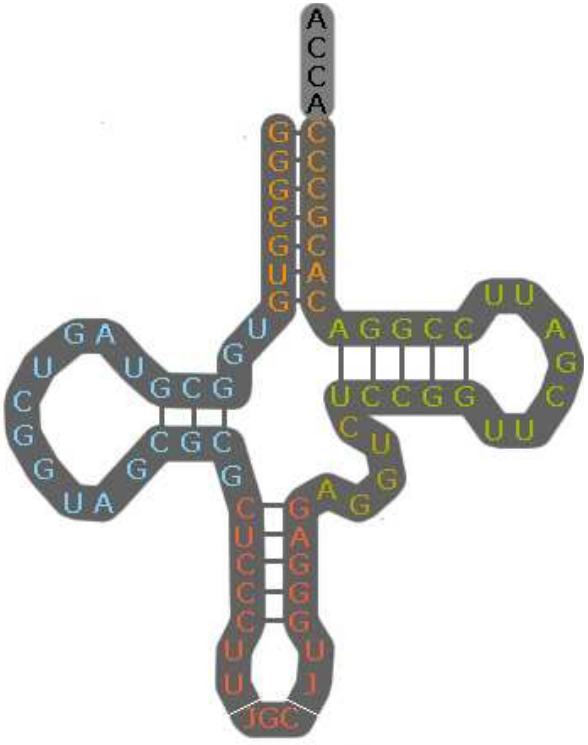
Solving the general problem of
RNA sequence-structure alignment with CP

RNA Structure

primary structure = sequence (of bases A,C,G,U)

GGGCGUGUGGCGUAGUCGGUA ... GUUCGAUUCCGGACACGCCCACCA

secondary structure



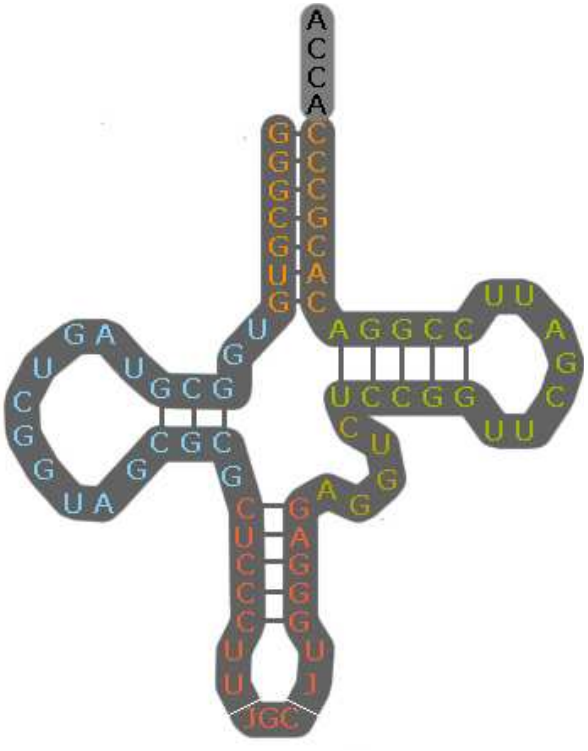
tertiary structure

RNA Structure

primary structure = sequence (of bases A,C,G,U)

GGGCGUGUGGCGUAGUCGGUA ... GUUCGAUUCCGGACACGCCCACCA

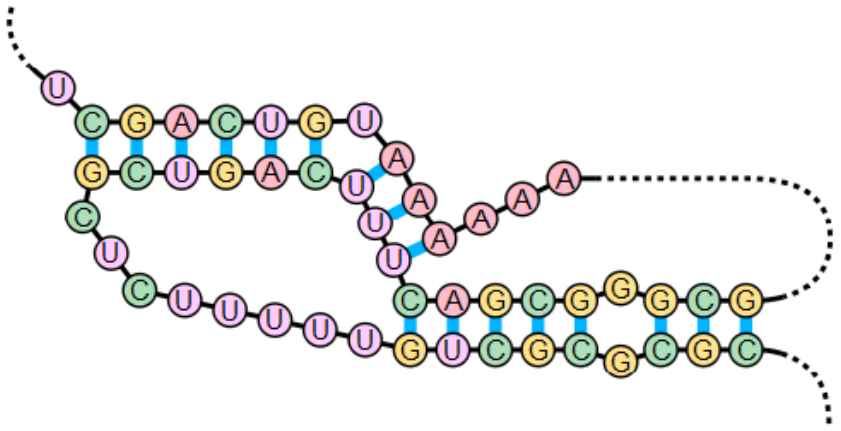
secondary structure



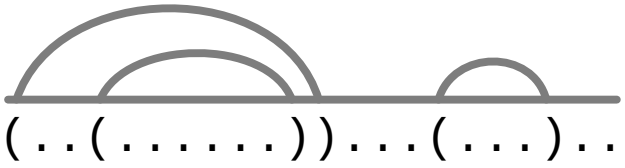
tertiary structure

secondary structure = set of base pairs (i, j)

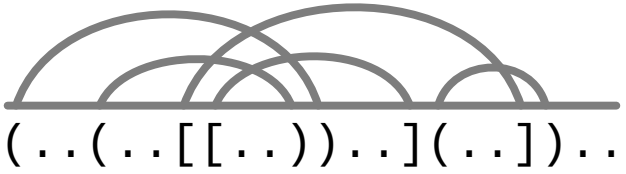
Pseudoknots



- RNA may contain pseudoknots



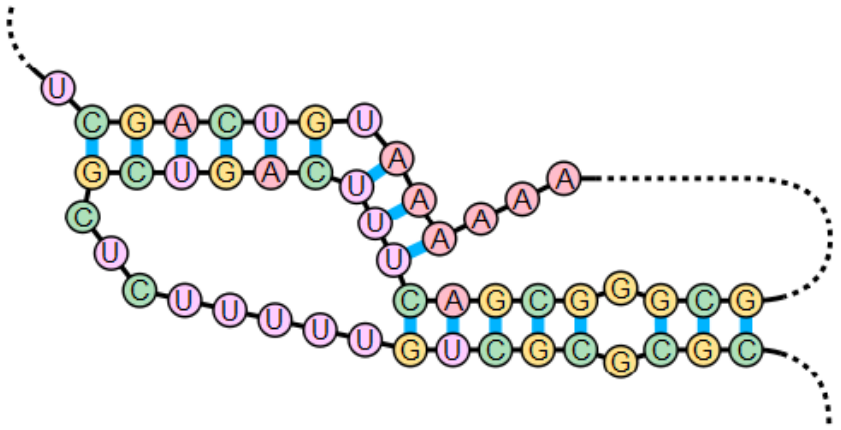
- no pseudoknots
- non−crossing
- well bracketed



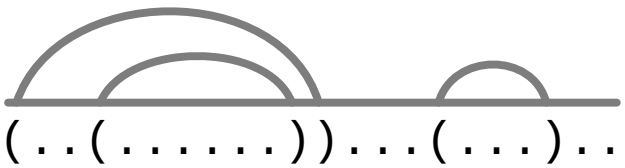
- with pseudoknots
- crossing
- not well bracketed

- Pseudoknots make RNA problems NP-hard
  - Structure Prediction
  - Alignment

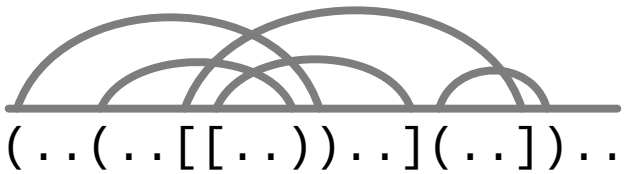
Pseudoknots



- RNA may contain pseudoknots



– no pseudoknots
– non−crossing
– well bracketed



– with pseudoknots
– crossing
– not well bracketed

- Pseudoknots make RNA problems NP-hard
  - Structure Prediction
  - Alignment

Structure Classes

arbitray set of base pairs (i, j)

(N) Nested Structure

no pseudoknots, degree $<= 1$

efficient DP-algorithms

(C) Crossing Structure

degree $<= 1$ (pseudoknots allowed)

NP-hard

(U) Unlimited Structure

unrestricted, NP-hard

RNA Alignment Example: Antizyme FSE

IN:

```
>RNA1
UGAUGUCCCUCUCCCACCCCUGAAGAUCCCAGGUGGGCGAGGGAAUGAUCAGCGGGAUC
.....((((((.((((((......[[[[[[.))))))).))))))........]]]]]]
>RNA2
UGAUGCCCCUCACCCACUGUUGAAGACCCUCAGUGGGUGAGGGGGCGGCAAGGAUC
.....((((((.((((((......[[[[[[.))))))).))))))......]]]]]]
```

OUT:

```
.....((((((.((((((......[[[[[[.))))))).))))))........]]]]]]
UGAUGUCCCUCUCCCACCCCUGAAGAUCCCAGGUGGGCGAGGGAAUGAUCAGCGGGAUC
UGAUGCCCCUCACCCACUGUUGAAGACCCUCAGUGGGUGAGGGGGCG-GCA--AGGAUC
.....((((((.((((((......[[[[[[.))))))).))))))...-...--]]]]]]
```

RNA Alignment Toy Example

IN:

```
>RNA1
GCCAUACGGCAUAC
(((.[[.))).]].
>RNA2
GGUUGCCGCCAACAC
((([[..)))..]].
```

OUT:

```
     (((.[[-.)))-.]].
RNA1 GCCAUA-CGGC-AUAC
RNA2 GGU-UGCCGCCAACAC
     (((-[[..)))..]].
```

alignment = set of edges
between 'aligned' bases of RNA 1 and RNA 2

The RNA Alignment Problem

Given: sequence-structure pairs (A, PA) and (B, PB).

The *alignment problem* is

$$\underset{A \text{ alignment of } (A,PA) \text{ and } (B,PB)}{\operatorname{argmax}} \quad \textbf{score}(A).$$

$$\begin{aligned}
\textbf{score}(A_m \cup A_g) := & \sum_{(i,i') \in A_m} \sigma(i,i') && \text{(sequence)} \\
& + \sum_{\substack{(i,j) \in P_a, (i',j') \in PB, \\ (i,i') \in A_m, (j,j') \in A_m}} \tau(i,j,i',j') && \text{(structure)} \\
& + \quad \gamma |A_g| && \text{(gaps)}
\end{aligned}$$

Related Work

- Easy: alignment of nested RNA
  - $O(n^4)$ Jiang et al., General Edit Distance, JCB 2002.
  - $O(n^3)$ Demaine et al., Optimal Decomposition for Tree Edit Distance, TALG 2009.
- Hard: crossing/pseudoknots
  - Evans, Finding Common Pseudoknot Structures in Polynomial Time, CPM 2006. (restricted crossing, efficient)
  - Möhl et al., Lifting prediction to alignment of RNA pseudoknots, JCB 2010. (restricted crossing, efficient)
  - Möhl et al., FPT alignment of RNA structures including arbitrary pseudoknots, CPM 2008. (arbitrary crossing, FPT)

Related Work II: SA&F

Simultaneous Alignment and Folding (SA&F)

- aligns RNA with unlimited structure
- but selects ($\rightarrow$ F!) a nested or crossing structure

Approaches

selecting nested structure (dynamic programming)

- Sankoff, original SA&F, J Appl Math 1985.
- Mathews et al., Dynalign, JMB 2002.
- Havgaard et al., Foldalign, Plos Comp Biol 2007.
- Will et al., LocARNA, Plos Comp Biol, 2007.

selecting crossing structure (ILP)

Bauer et al., Lara, BMC Bioinformatics 2007.

Recall: here, we wont select a sub-structure for scoring,
but score all aligned structural elements

Constraint Model

For sequence-structure pairs (A, PA) and (B, PB),

define variables $MD_1, \ldots, MD_{|A|}$ and $M_1, \ldots, M_{|B|}$.

Semantics

- $MD_i = j$: "match or deletion of A_i to/after B_j"
- $M_i = 1$: "match of A_i"

For example, the alignment $\begin{matrix}\texttt{A--CUG}\\ \texttt{ACAC-G}\end{matrix}$,

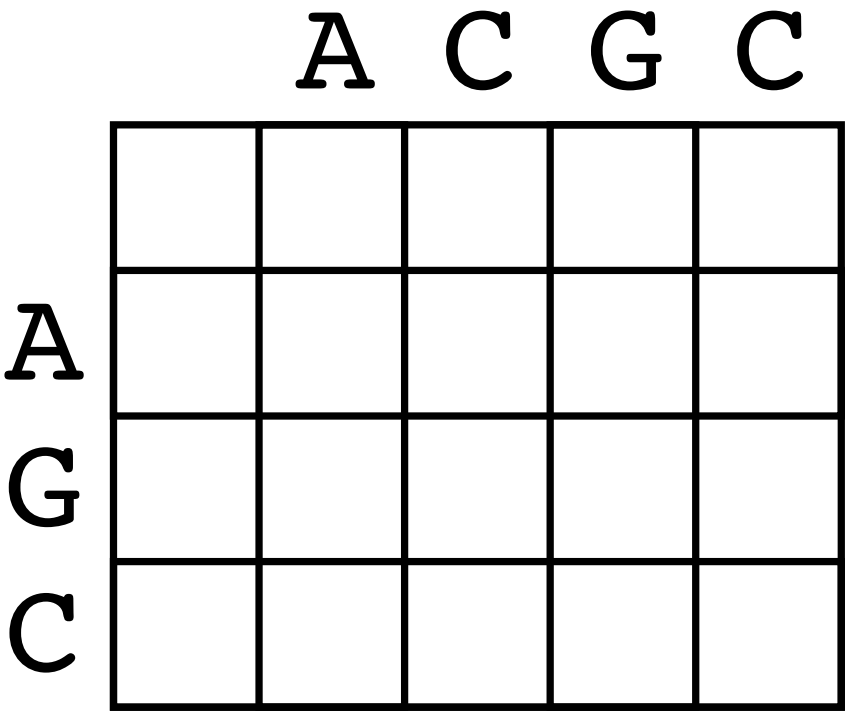
corresponds to variables and values

$(MD_1, \ldots, MD_4) = (1, 4, 4, 5)$ and $(M_1, \ldots, M_4) = (1, 1, 0, 1)$.

We don't need all details. Vars define traces in "alignment graph".

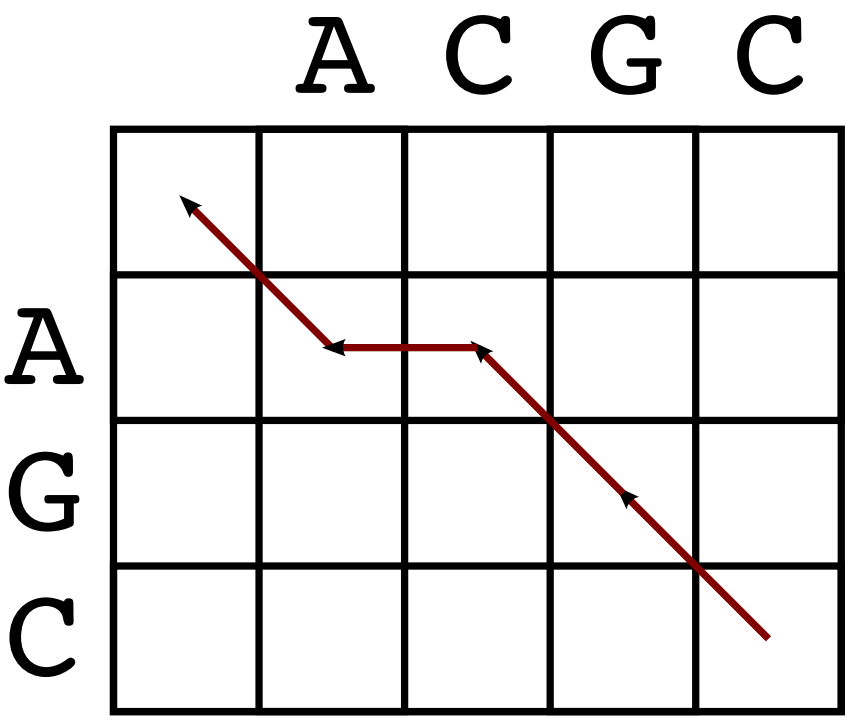
Alignment Graph

Sequences A =AGC, B =ACGC



Alignment Graph

Sequences A =AGC, B =ACGC

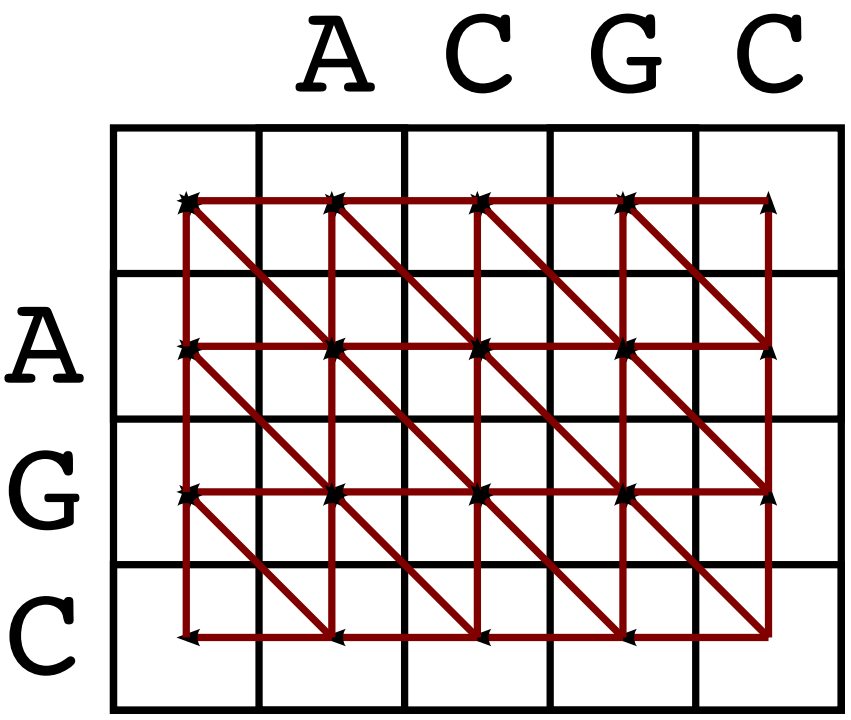


A-GC
ACGC

1 Trace $\equiv$ 1 Alignment

Alignment Graph

Sequences A =AGC, B =ACGC

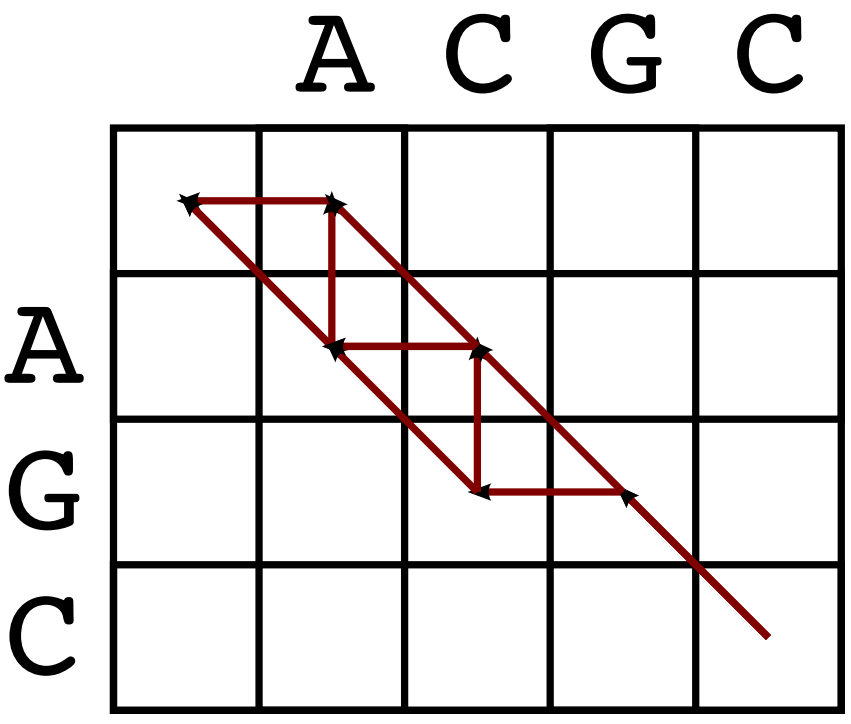


ALL

Graph of all Match/Delete/Insert edges $\equiv$ All alignments

Alignment Graph

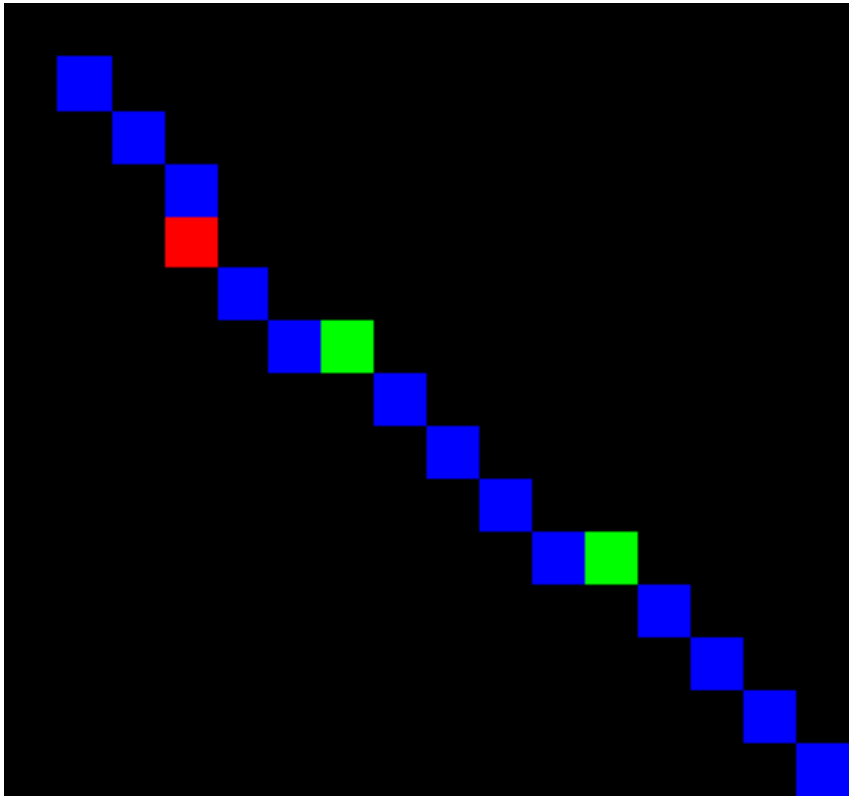
Sequences A =AGC, B =ACGC




```
A-GC
ACGC

AG-C
ACGC

AG-C
ACGC

-AG-C
AC-GC

...
```

Graph of M/I/D edges ≡ Set of alignments
(specified by variable domain or "Constraint Store")

Alignment Graph Color Code



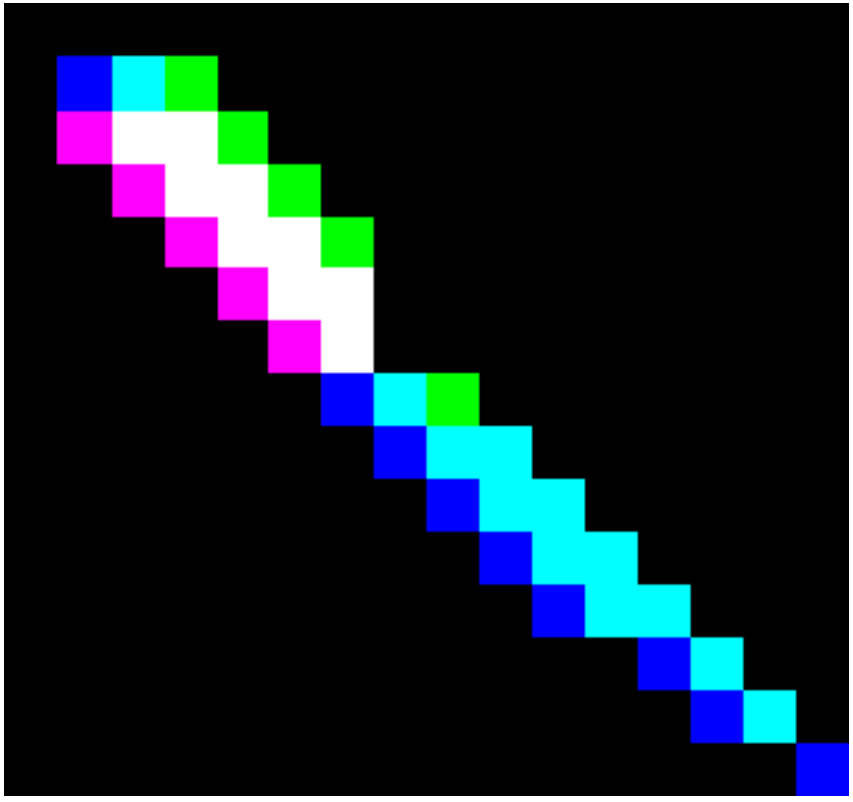
red = deletion

green = insertion

blue = match

Use to represent valuation of variables, i.e. one alignment

Alignment Graph Color Code

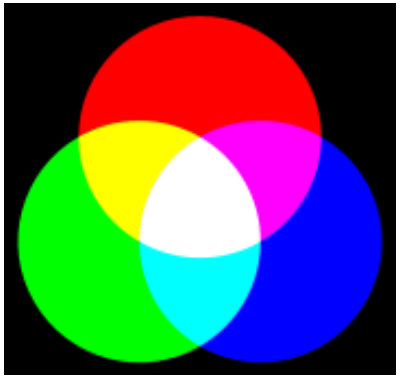


red = deletion

green = insertion

blue = match

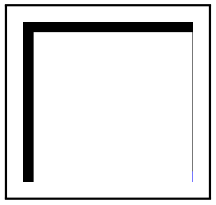
. . . and synthesize colors



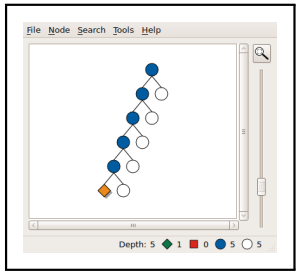
Use to represent set of valuations of variables, i.e. Constraint Store

Solving Strategy

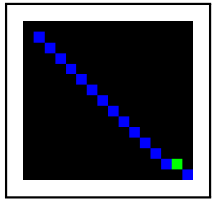
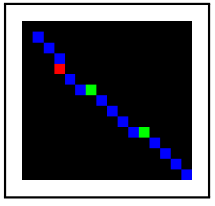
- Start with all possible alignments:



- Enumerate: binary intervall splitting

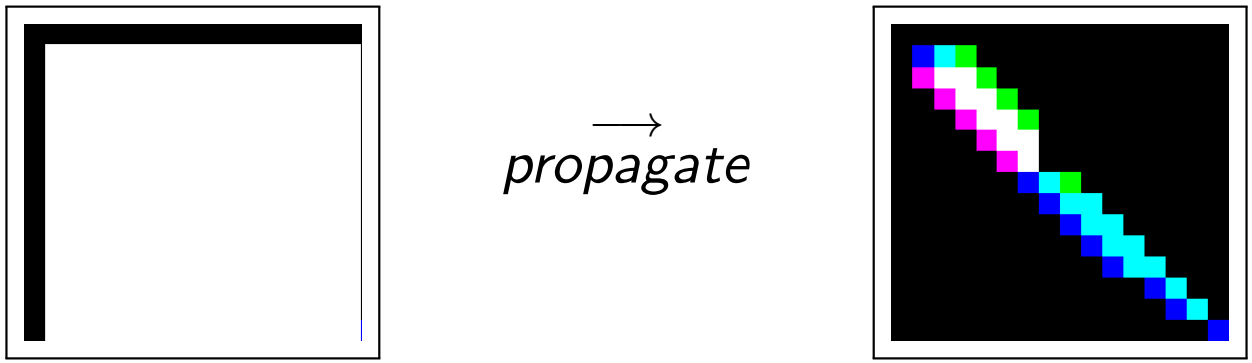


- Branch-and-Bound: search better and better alignments


, 
, . . .

- Crucial: new global constraint propagator

Constraint Propagator



Propagates two constraints:

- $\mathit{Alignment}(\mathbf{MD}, \mathbf{M})$

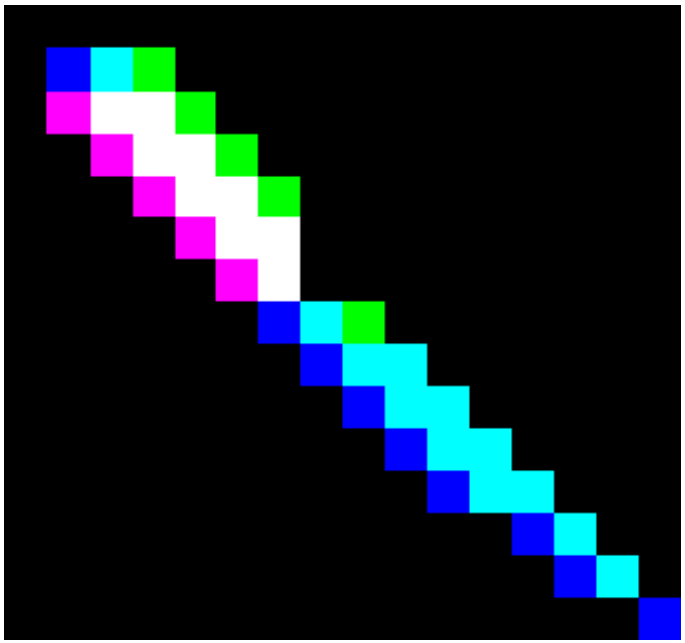
  "variables encode alignment"

- $\mathit{AlignmentScore}_{(A,PA),(B,PB)}(\mathbf{MD}, \mathbf{M}, \mathit{Score})$

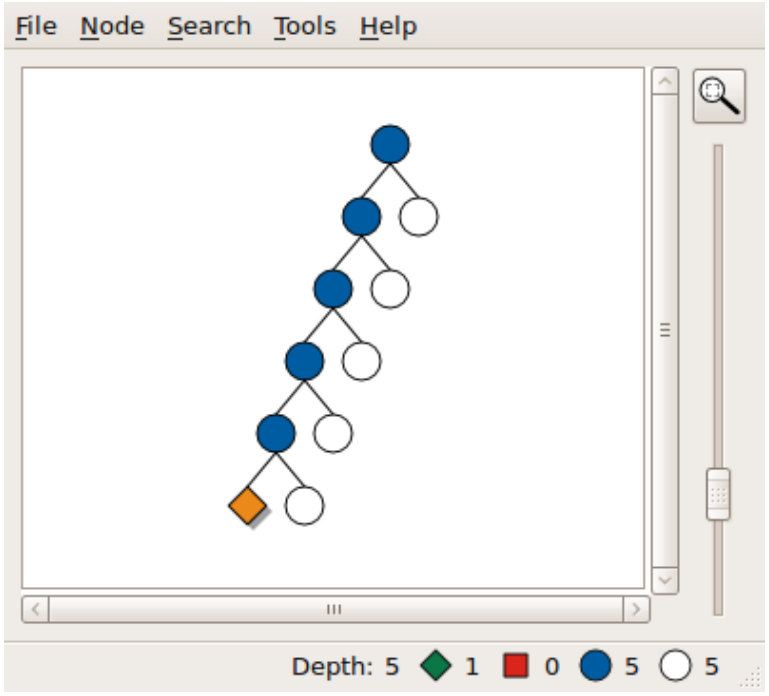
  "*Score* is score of alignment by $\mathbf{MD}, \mathbf{M}$"

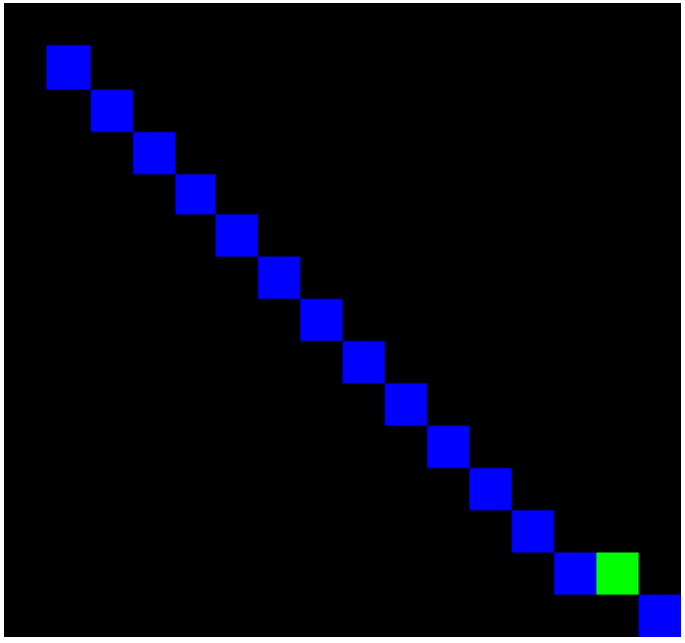
Example
(Run of Gecode Implementation on Toy Example)

```
>RNA1
GCCAUACGGCAUAC
(((.[[.))).]].
>RNA2
GGUUGCCGCCAACAC
((([[..)))..]].
```



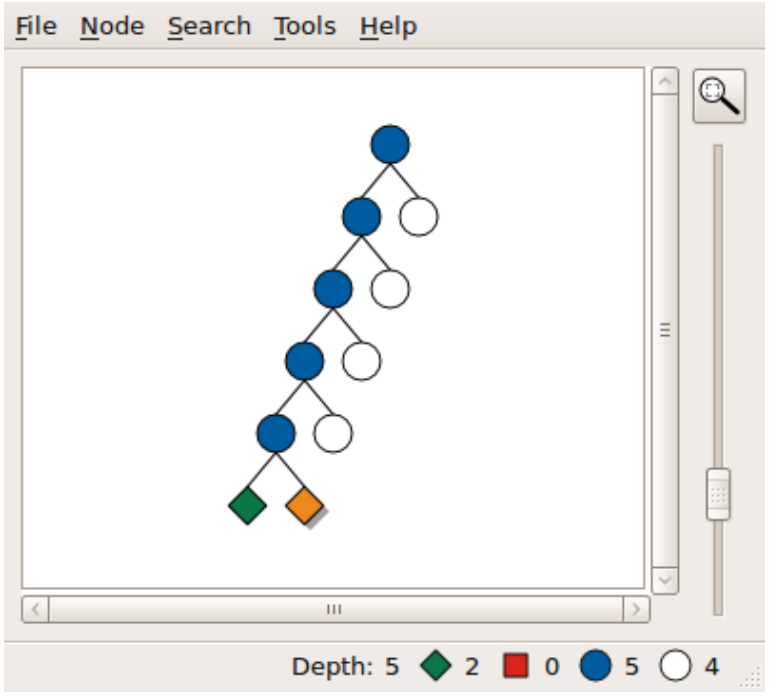
Example

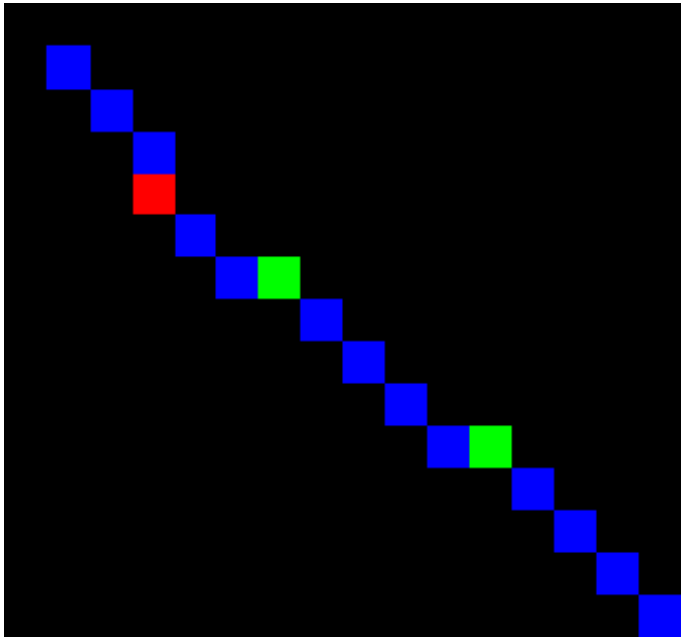




```
(((.[[.))).]]-.
GCCAUACGGCAUA-C
GGUUGCCGCCAACAC
((([[..)))..]].
```

Example

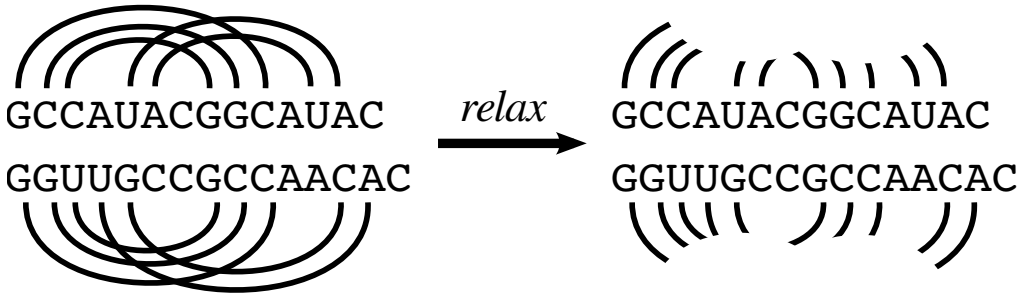




```
(((.[[-.)))-.]].
GCCAUA-CGGC-AUAC
GGU-UGCCGCCAACAC
(((-[[..)))..]].
```

Propagator: Upper Bound by Relaxation

```
>RNA1
GCCAUACGGCAUAC
(((.[[.))).]].
>RNA2
GGUUGCCGCCAACAC
((([[..)))..]].
```



$$\textbf{score}_{\text{relaxed}}(A_m \cup A_g) := \sum_{(i,i') \in A_m} \left[\sigma(i,i') + \text{ub}(i,i')\right] \quad + \quad \gamma|A_g|,$$

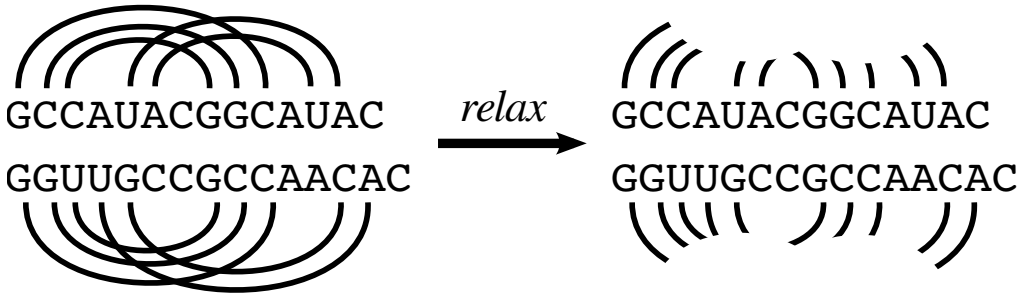
$$\text{ub}(i,i') = \begin{cases} \frac{1}{2}\tau(i,j;i',j') & \text{iff match } (j,j') \text{ allowed by constraint store} \\ 0 & \text{otherwise} \end{cases}$$

$\textbf{score}_{\text{relaxed}}$ is upper bound of **score**. Computed in $O(n^2)$ by DP!

Given ub for crossing. For unlimited, sum over all (j,j'). Use structural sparsity to limit complexity!

Propagator: Upper Bound by Relaxation

```
>RNA1
GCCAUACGGCAUAC
(((.[[.))).]].
>RNA2
GGUUGCCGCCAACAC
((([[..)))..]].
```



$$\textbf{score}_{\text{relaxed}}(A_m \cup A_g) := \sum_{(i,i') \in A_m} \left[\sigma(i,i') + \text{ub}(i,i')\right] \quad + \quad \gamma|A_g|,$$

$$\text{ub}(i,i') = \begin{cases} \frac{1}{2}\tau(i,j;i',j') & \text{iff match } (j,j') \text{ allowed by constraint store} \\ 0 & \text{otherwise} \end{cases}$$

score$_{\text{relaxed}}$ is upper bound of **score**. Computed in $O(n^2)$ by DP!

Given ub for crossing. For unlimited, sum over all (j,j'). Use structural sparsity to limit complexity!

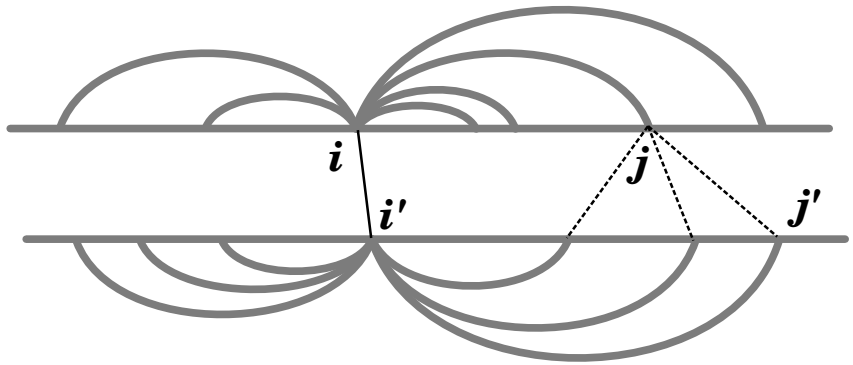
Upper Bound UB Revisited: Structural Sparsity

Full form of upper bound *ub*:

$$\mathrm{ub}_D(i,i') = \frac{1}{2} \max_{A \in \mathcal{A}(D)} \left\{ \begin{array}{c} \sum_{\substack{(i,j)\in PA,(i',j')\in PB \\ (i,i')\in A,(j,j')\in A}} \tau(i,j;i',j') \\ +\text{symm. term for } \tau(j,i;j',i') \end{array} \right\}$$

D domain

$\mathcal{A}(D)$ allowed alignment for domain D



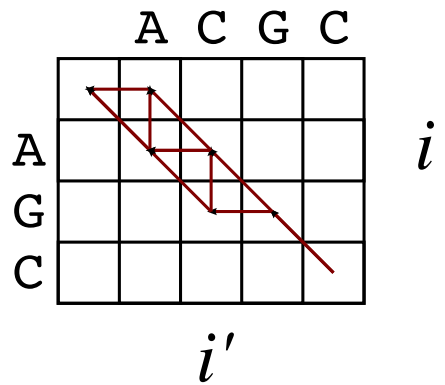
In many applications: structure is sparse, i.e. degree is limited by constant. Then: propagator complexity is $O(n^2)$. (E.g. aligning pair probability matrices, confer SA&F by Will et al., LocARNA)

Propagator: Bound for each match/deletion/insertion

- $\textbf{score}_{\text{relaxed}}$ is a "sequence alignment score"
- compute the relaxed scores of prefix alignments by DP (Smith-Waterman), filling a matrix: "forward algorithm"
- perform symmetric "backward algorithm", relaxed scores of suffix alignment
- putting together: e.g.

$$\text{Prefix}\,(i, i') + \text{Suffix}\,(i, i')$$

is upper bound for trace through (i, i').



- upper bound $<$ lower bound (from B&B) $\Longrightarrow$ PRUNE
- trace through DP-matrix yields alignment A_0; $\textbf{score}(A_0)$ is lower bound for the alignment problem

Problem Decomposition and Symmetry Breaking

- decomposition into independent subproblems speed up branch-and-bound search (AND/OR-search, Marinescu&Dechter,AI2009.)
- in general, overhead for detecting components during search
- here: problem suggests decomposition along sequence order
- cheap check by global propagator
- upper and lower bounds of independent components available
- full decomposition during search is Future Work
- currently: decompose in simple case
  - subsequences $ii'; jj'$ without structure; ii' matched, jj' matched
  - optimal solution of subproblem = optimal sequence alignment
  - fix vars of subproblem (= form of symmetry breaking)

Results

Implementation `Carna` (Constraint Alignment of RNA) in Gecode compared to `Lara` (ILP) on crossing structure

Benchmark set = pairwise alignments from 16 PK-families in Rfam

Family	Lengths		Run-time (s)		Carna Search Tree		
	A	B	Carna	Lara	Depth	Fails	Size
Entero_OriR	126	130	0.03	0.18	38	13	50
IRES_Cripavirus	202	199	0.2	0.04	157	127	296
RNaseP_arch	303	367	0.46	1.4	63	8	64
RNaseP_nuc	317	346	0.07	2.9	14	4	16
tmRNA	384	367	63	3.7	433	14347	28785
RNaseP_bact_b	408	401	3.0	2.3	370	677	1463
Intron_gpI	443	436	0.1	0.2	0	0	1
Telomerase-vert	448	451	0.47	2.3	146	32	161

Results for hard instances from the Rfam PK-families

8 easy instances: both progs align in ≤ 0.1 seconds.

Results (Unlimited structure)

Folding of bi-stable, riboswitch RNAs (100 instances per family)

Family	Length	Base Pairs	Time (s)	Limit
SAH_riboswitch	79	81	0.13	2%
SAM_alpha	79	96	0.03	0%
Purine	101	74	0.07	0%
Glycine	101	83	0.44	3%
SAM	106	74	0.06	0%
TPP	107	96	0.43	13%
SAM-IV	116	128	0.05	2%
MOCO_RNA_motif	141	111	0.24	10%
Lysine	181	210	60	60%
Cobalamin	204	237	60	71%

Strict time limit 1 min.

Conclusion

- Successful CP-application to general RNA alignment

  crossing structure: On par with ILP/Lara
  unlimited structure: NEW
- Extensible to affine gap cost: use Gotoh-like DP in propagator
- Extensible by (low-overhead) AND/OR search
- Applicable to alignment of
  - Pseudoknotted RNA
  - RNA switches / bi-stable RNA
  - RNA with conserved dynamics
- Open Problem: simultaneous folding and alignment in CP
  — stronger bounds, but select structure
- Read more details in conference paper: Dal Palù et al., A Propagator for Maximum Weight String Alignment with Arbitrary Pairwise Dependencies, CP 2010.