Sequence-Structure RNA Alignments using Lagrangian Relaxation

Markus Bauer[1,2] Gunnar W. Klau[3] Knut Reinert[1]

[1]FU Berlin: Algorithmic Bioinformatics

[2]IMPRS on Computational Biology & Scientific Computing, Berlin

[3]FU Berlin: Mathematics in Life Sciences—DFG Research Center MATHEON

28 August 2009

Discrete Math lecture WS 09

Rediscovery of RNA ...

nature

MARS'S ANCIENT OCEAN
Polar wander solves an enigma

THE DEPTHS OF DISGUST
Understanding the ugliest emotion

MENTORING
How to be top

NATUREJOBS
Contract research

DECODING THE BLUEPRINT
The ENCODE pilot maps human genome function

Rediscovery of RNA ...

Rediscovery of RNA ...

Rediscovery of RNA ...

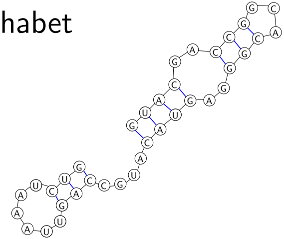
"It is beginning to dawn on biologists that they may have got it wrong. Not completely wrong, but wrong enough to be embarrassing." (The Economist, June 14th 2007)

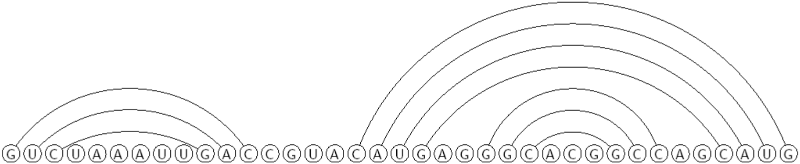
RNA

- On the sequence level: string over the alphabet $\{A, C, G, U\}$

RNA

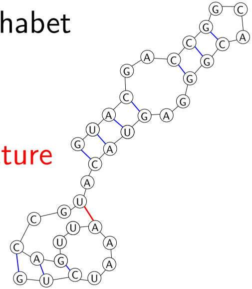
- On the sequence level: string over the alphabet $\{A, C, G, U\}$
- Folds onto itself → secondary structure

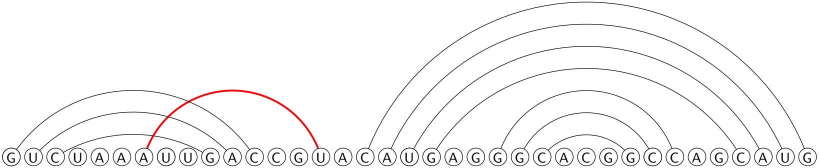




RNA

- On the sequence level: string over the alphabet $\{A, C, G, U\}$
- Folds onto itself → secondary structure
- Can contain pseudoknots → tertiary structure



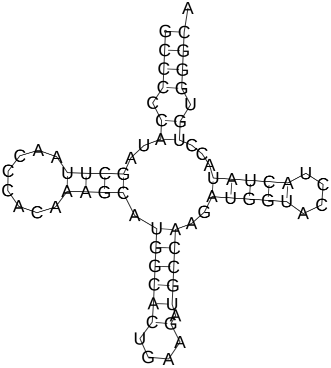


Real-world example: tRNA

- Tertiary structure:

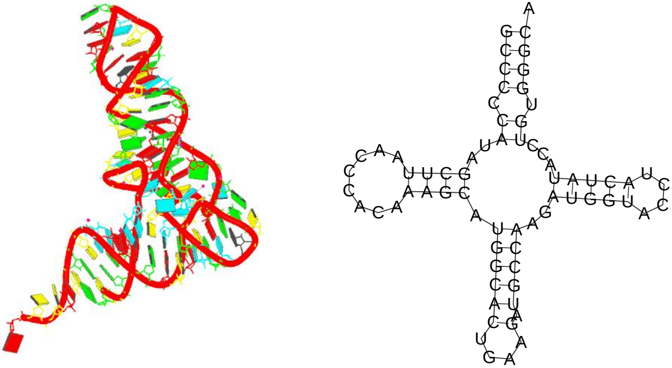
Real-world example: tRNA

- Secondary structure:



Real-world example: tRNA

- Primary structure:



GCCCCCAUAGCUUAACCCACAAAGCAUGGCACUGAAGAUGCCAAGAUGGUACCUACUAUACCUGUGGGCA

Sequence-structure alignments

- Function largely depends on structure
    - Goal: finding functional motifs, *i. e.*, conserved structures that play an important role
    - Related functional RNAs often have low sequence but high structural similarity

Sequence-structure alignments

- Function largely depends on structure
  - Goal: finding functional motifs, *i. e.*, conserved structures that play an important role
  - Related functional RNAs often have low sequence but high structural similarity
- Similar function can often be detected by finding structural similarities → need to compute sequence-structure alignments

Sequence-structure alignments

- Function largely depends on structure
  - Goal: finding functional motifs, *i. e.*, conserved structures that play an important role
  - Related functional RNAs often have low sequence but high structural similarity
- Similar function can often be detected by finding structural similarities → need to compute sequence-structure alignments
- Sequence-structure alignments serve as the basis for computing RNA consensus structures, finding RNA genes, structural clustering,. . .

Sequence-structure alignments: previous work

- Polynomial algorithms (mainly based on DP) exist for the nested pairwise case, *e.g.*, [Sankoff, 86],[Tai, 79],[Jiang, 95],[Eddy, 94],. . .
- NP-complete in the multiple case and in the general unnested case ([Reinert, 98])

Sequence-structure alignments: previous work

- Polynomial algorithms (mainly based on DP) exist for the nested pairwise case, *e.g.*, [Sankoff, 86],[Tai, 79],[Jiang, 95],[Eddy, 94],. . .
- NP-complete in the multiple case and in the general unnested case ([Reinert, 98])
- [Lancia/Caprara, 02] use *Lagrangian relaxation* to solve the maximal *contact map overlap* problem

Sequence-structure alignments: previous work

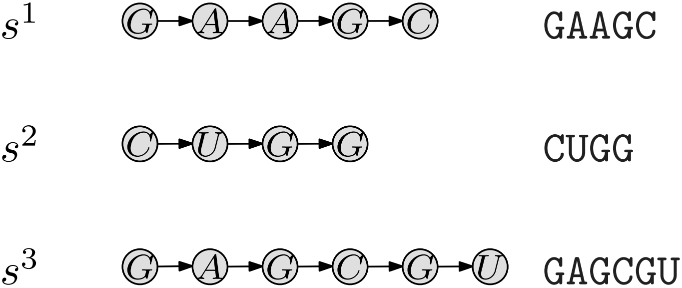
- Polynomial algorithms (mainly based on DP) exist for the nested pairwise case, *e.g.*, [Sankoff, 86],[Tai, 79],[Jiang, 95],[Eddy, 94],. . .
- NP-complete in the multiple case and in the general unnested case ([Reinert, 98])
- [Lancia/Caprara, 02] use *Lagrangian relaxation* to solve the maximal *contact map overlap* problem

Two lines of research

- A novel formulation for exact multiple sequence-structure alignments of known and unknown structures (combining models from [Althaus, 06] and [Bauer, 04])
- Computing fast multiple sequence-structure alignments based on the pairwise alignment case

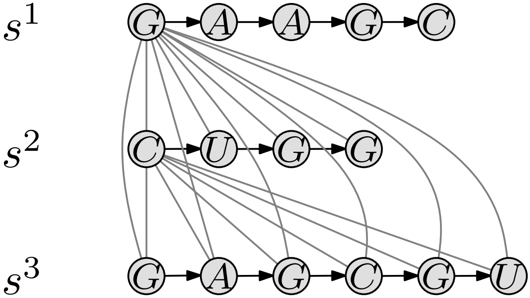
Graph-based formulation 1/5

- Vertices:



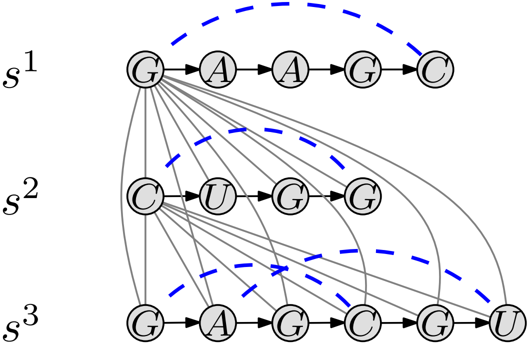
Graph-based formulation 2/5

- Alignment edges:



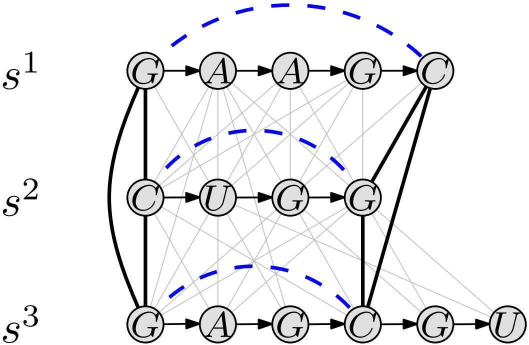
Graph-based formulation 3/5

- Interaction edges:



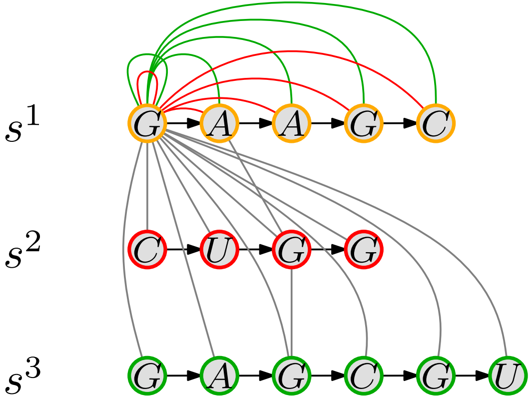
Graph-based formulation 3/5

- Interaction match:



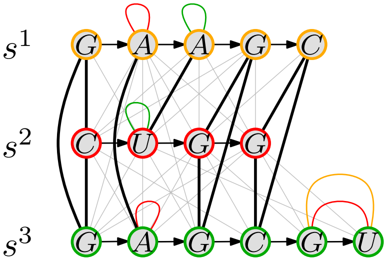
Graph-based formulation 4/5

- Gap edges:



Graph-based formulation 4/5

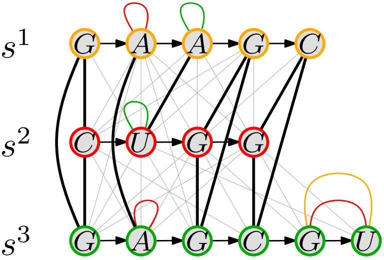
- Realized gap edges:



```
GAAGC--
C-UGG--
GA-GCGU
```

Graph-based formulation 4/5

- Realized gap edges:

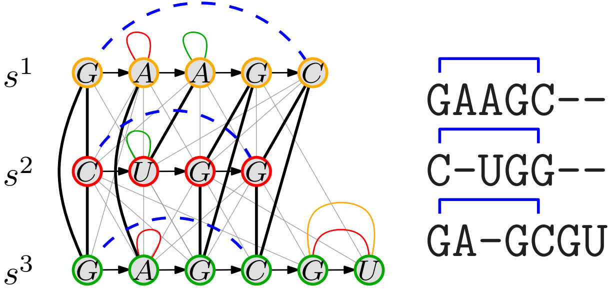


```
GAAGC--
C-UGG--
GA-GCGU
```

- Summary of the different edges:
  - alignment edges (alignment)
  - interaction edges (structure)
  - gap edges (gaps)

Graph-based formulation 5/5

- Objective function of sequence-structure alignments:



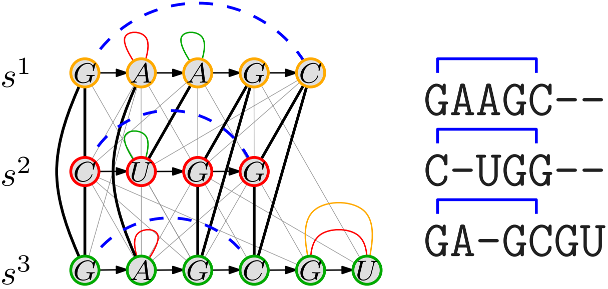
maximize the sum of realized sequence plus structure scores, *i.e.*, award matches, penalize mismatches and gaps

Gapped Structural Traces

- Not all possible subsets of alignment, interaction, or gap edges correspond to proper alignments
- Adding constraints leads to the notion of a gapped structural trace
- A gapped structural trace corresponds to a proper multiple sequence-structure alignment

Gapped Structural Traces

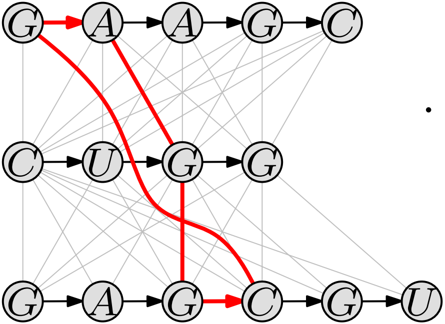
- Not all possible subsets of alignment, interaction, or gap edges correspond to proper alignments
- Adding constraints leads to the notion of a gapped structural trace
- A gapped structural trace corresponds to a proper multiple sequence-structure alignment, *e.g.*,



```
GAAGC--
C-UGG--
GA-GCGU
```

Gapped Structural Traces 1/5

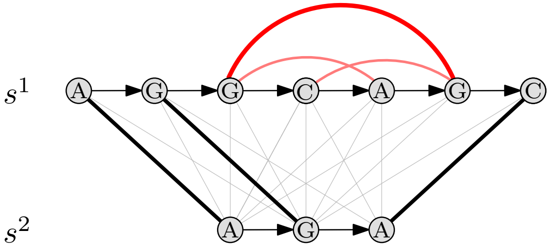
- We do not allow mixed cycles:



```
··· G A ···
  ··· G ···
  ··· G C ···
```

Gapped Structural Traces 2/5

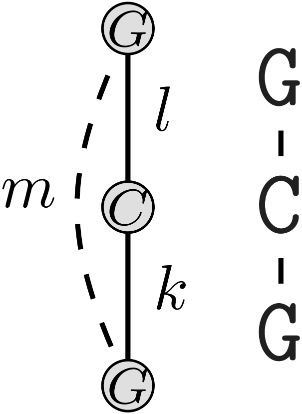
- We do not allow conflicting gap edges, *i.e.*, gaps are realized by one single gap edge:



```
AGGCAGC
AG----A
```

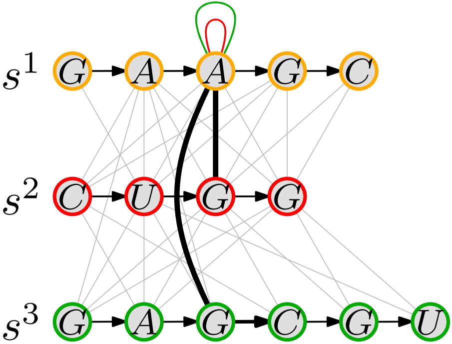
Gapped Structural Traces 3/5

- We have to realize transitive edges:



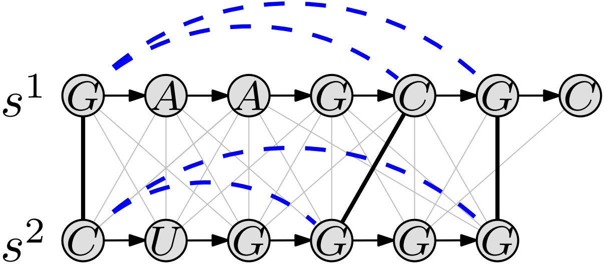
Gapped Structural Traces 4/5

- Every vertex has to be incident to an alignment or gap edge:



Gapped Structural Traces 5/5

- At most one interaction match counts:



What we have so far...

- We have a graph-based framework modelling multiple sequence-structure alignments
- But: we do not have an algorithm yet for determining the subsets of alignment, interaction, and gap edges

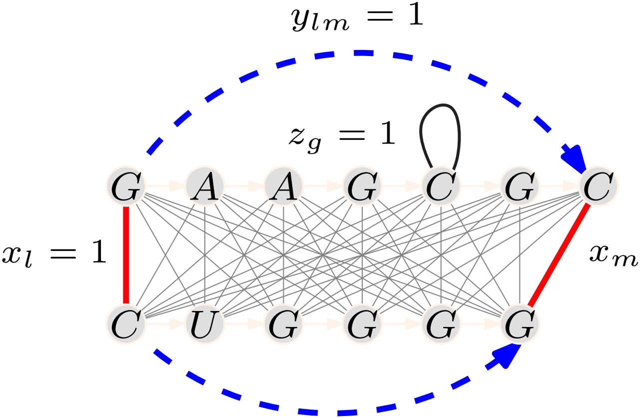
What we have so far...

- We have a graph-based framework modelling multiple sequence-structure alignments
- But: we do not have an algorithm yet for determining the subsets of alignment, interaction, and gap edges
- Combinatorial optimization deals with determining the best solution out of a finite set of feasible solutions
- Integer linear programs are one of the main tools to solve combinatorial optimization problems
- The graph-based formulation gives rise to such an integer linear program

Integer linear program variables

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L\times L}, z \in \{0,1\}^G$

$$x_l = \begin{cases} 1 & l \in \mathcal{L} \\ 0 & \text{else} \end{cases} \quad y_{lm} = \begin{cases} 1 & \text{(l,m) match realized} \\ 0 & \text{else} \end{cases} \quad z_g = \begin{cases} 1 & g \in \mathcal{G} \\ 0 & \text{else} \end{cases}$$



Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

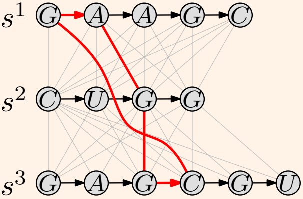
$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- There is no mixed cycle induced by the alignment:



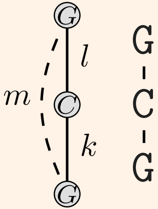
$$\sum_{l \in L \cap M} x_l \leq |L \cap M| - 1$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- We realize transitive edges:



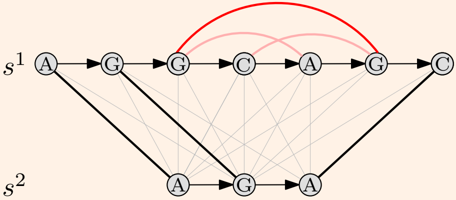
$$x_l + x_k - x_m \leq 1$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- There are no two gap edges in conflict with each other:



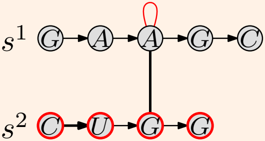
$$\sum_{a \in C} z_a \leq 1$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- Each vertex is incident to an alignment edge or spanned by a gap edge (w.r.t. every other input sequence):



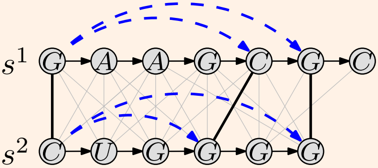
$$\sum_{l \in L^{ij}_{s(m)}} x_l + \sum_{a \in G^{ij}_{s(l) \leftrightarrow s(l)}} z_a = 1$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- An alignment edge can realize at most one single interaction match:



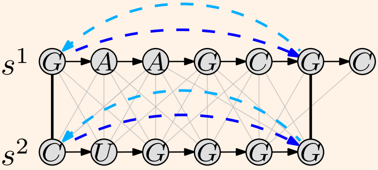
$$\sum_{m \in L} y_{lm} \leq x_l$$

Gapped structural traces

Given a weighted alignment graph $G = (V, L \cup I \cup G, w)$, we aim at finding the sequence-structure alignment of maximal weight, *i.e.*, select $\mathcal{L} \subseteq L$, $\mathcal{I} \subseteq I$, and $\mathcal{G} \subseteq G$ with

$$\max \sum_{l \in \mathcal{L}} w_l x_l + \sum_{l \in \mathcal{L}} \sum_{m \in \mathcal{L}} w_{lm} y_{lm} + \sum_{g \in \mathcal{G}} w_g z_g$$

- Directed interaction matches have to match, *i.e.*, they have to be realized from both sides:



$$y_{lm} = y_{ml}$$

ILP modelling gapped structural traces

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L \times L}, z \in \{0,1\}^G$

$$\max \quad \sum_{l \in L} w_l x_l + \sum_{l \in L} \sum_{m \in L} w_{lm} y_{lm} + \sum_{g \in G} w_g z_g$$

$$\text{s. t.} \quad \sum_{l \in L \cap M} x_l \leq |L \cap M| - 1 \qquad \forall M \in \mathcal{M}$$

$$x_l + x_k - x_m \leq 1 \qquad (x_l, x_k, x_m) \text{ forming a cycle}$$

$$\sum_{a \in C} z_a \leq 1 \qquad \forall C \in \mathcal{C}$$

$$\sum_{l \in L^{ij}_{s(m)}} x_l + \sum_{a \in G^{ij}_{s(l) \leftrightarrow s(l)}} z_a = 1 \qquad 1 \leq i, j \leq k, i \neq j, \forall m \in L^{ij}$$

$$\sum_{m \in L} y_{lm} \leq x_l \qquad \forall\, l \in L$$

$$y_{lm} = y_{ml} \qquad \forall\, l, m \in L$$

Computing optimal solution for ILPs

- Solving general integer linear programs is NP-complete
- Popular techniques tackling ILPs:
    - branch-and-cut
    - Lagrangian relaxation

Computing optimal solution for ILPs

- Solving general integer linear programs is NP-complete
- Popular techniques tackling ILPs:
  - branch-and-cut
  - Lagrangian relaxation

- Lagrangian relaxation:
  - we are able to divide the constraints into good and bad constraints
  - dropping the bad constraints makes the problem easier to solve
  - move the bad constraints into the objective function associated with a penalty vector

Integer linear program

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L \times L}, z \in \{0,1\}^G$

$$\max \quad \sum_{l \in L} w_l x_l + \sum_{l \in L} \sum_{m \in L} w_{lm} y_{lm} + \sum_{g \in G} w_g z_g$$

$$\begin{array}{lll}
\text{s. t.} & \displaystyle\sum_{l \in L \cap M} x_l \leq |L \cap M| - 1 & \forall M \in \mathcal{M} \\
& x_l + x_k - x_m \leq 1 & (x_l, x_k, x_m) \text{ forming a cycle} \\
& \displaystyle\sum_{a \in C} z_a \leq 1 & \forall C \in \mathcal{C} \\
& \displaystyle\sum_{l \in L^{ij}_{s(m)}} x_l + \sum_{a \in G^{ij}_{s(l) \leftrightarrow s(l)}} z_a = 1 & 1 \leq i, j \leq k, i \neq j, \forall m \in L^{ij} \\
& \displaystyle\sum_{m \in L} y_{lm} \leq x_l & \forall\, l \in L \\
& y_{lm} = y_{ml} & \forall\, l, m \in L
\end{array}$$

Integer linear program

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L \times L}, z \in \{0,1\}^G$

$$
\begin{array}{lll}
\max & \displaystyle\sum_{l \in L} w_l x_l + \sum_{l \in L} \sum_{m \in L} w_{lm} y_{lm} + \sum_{g \in G} w_g z_g & \\
\text{s. t.} & \displaystyle\sum_{l \in L \cap M} x_l \leq |L \cap M| - 1 & \forall M \in \mathcal{M} \\
 & x_l + x_k - x_m \leq 1 & (x_l, x_k, x_m) \text{ forming a cycle} \\
 & \displaystyle\sum_{a \in C} z_a \leq 1 & \forall C \in \mathcal{C} \\
 & \displaystyle\sum_{l \in L^{ij}_{s(m)}} x_l + \sum_{a \in G^{ij}_{s(l) \leftrightarrow s(l)}} z_a = 1 & 1 \leq i, j \leq k, i \neq j, \forall m \in L^{ij} \\
 & \displaystyle\sum_{m \in L} y_{lm} \leq x_l & \forall\, l \in L \\
 & \xcancel{y_{lm} = y_{ml}} & \forall\, l, m \in L
\end{array}
$$

Integer linear program

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L\times L}, z \in \{0,1\}^G$

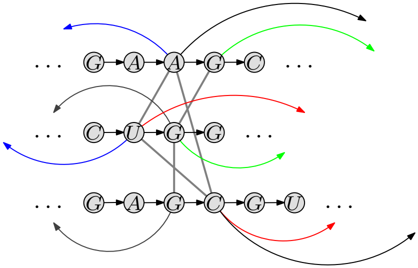
$$
\begin{array}{lll}
\max & \sum\limits_{l\in L} w_l x_l + \sum\limits_{l\in L}\sum\limits_{m\in L} w_{lm} y_{lm} + \sum\limits_{g\in G} w_g z_g \qquad + \sum\limits_{l\in L}\sum\limits_{m\in L} \lambda_{lm}(y_{lm} - y_{ml}) & \\
\text{s. t.} & \sum\limits_{l\in L\cap M} x_l \le |L\cap M| - 1 & \forall M \in \mathcal{M} \\
 & x_l + x_k - x_m \le 1 & (x_l, x_k, x_m) \text{ forming a cycle} \\
 & \sum\limits_{a\in C} z_a \le 1 & \forall C \in \mathcal{C} \\
 & \sum\limits_{l\in L^{ij}_{s(m)}} x_l + \sum\limits_{a\in G^{ij}_{s(l)\leftrightarrow s(l)}} z_a = 1 & 1 \le i,j \le k, i \ne j, \forall m \in L^{ij} \\
 & \sum\limits_{m\in L} y_{lm} \le x_l & \forall\, l \in L \\
 & \xcancel{y_{lm} = y_{ml}} & \forall\, l,m \in L
\end{array}
$$

Solving the relaxation intuitively. . .

- The remaining ILP describes a multiple sequence alignment with arbitrary gap costs
- We incorporate the weight of structure edges in the weight of the alignment edges

Solving the relaxation intuitively. . .

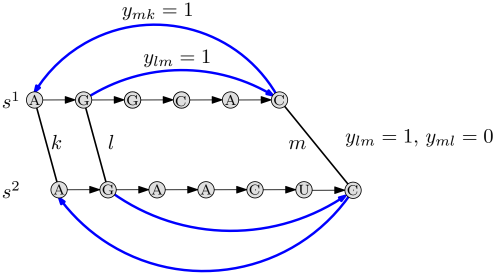
- The remaining ILP describes a multiple sequence alignment with arbitrary gap costs
- We incorporate the weight of structure edges in the weight of the alignment edges



- Each line chooses its highest scoring interaction match

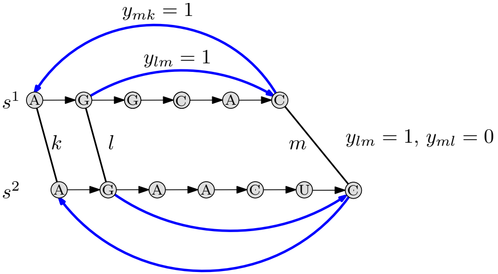
Solving the relaxation intuitively. . .

- The remaining ILP describes a multiple sequence alignment with arbitrary gap costs
- We incorporate the weight of structure edges in the weight of the alignment edges
- Vector λ acts as a penalty vector to punish violations:



Solving the relaxation intuitively...

- The remaining ILP describes a multiple sequence alignment with arbitrary gap costs
- We incorporate the weight of structure edges in the weight of the alignment edges
- Vector λ acts as a penalty vector to punish violations:



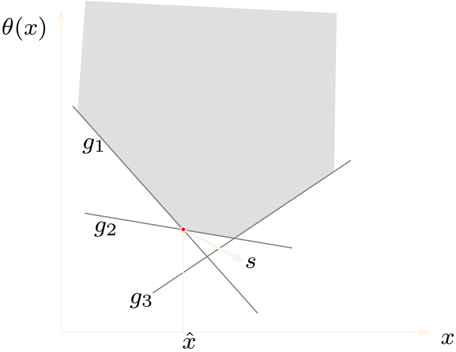
- The relaxation gives an upper bound on the original formulation

Subgradient and bundle methods

- There are two main methods for solving the Lagrangian dual, the subgradient or bundle method:

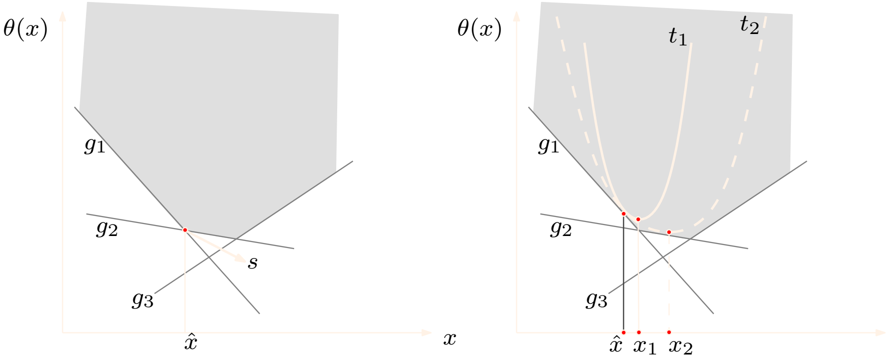
Subgradient and bundle methods

- There are two main methods for solving the Lagrangian dual, the subgradient or bundle method:



Subgradient and bundle methods

- There are two main methods for solving the Lagrangian dual, the subgradient or bundle method:



Computing the multipliers

We use subgradient optimization for this task:

- Start with $\lambda^0_{lm} = 0$ for all $l, m \in L$.

Computing the multipliers

We use subgradient optimization for this task:

- Start with $\lambda_{lm}^0 = 0$ for all $l, m \in L$.
- $\lambda_{lm}^{i+1} = \begin{cases} \lambda_{lm}^i & \text{if } s_{lm}^i := \overline{y_{lm}} - \overline{y_{ml}} = 0 \\ \max\{-w_{lm}, \lambda_{lm}^i - \gamma_i\} & \text{if } s_{lm}^i = 1 \\ \min\{w_{lm}, \lambda_{lm}^i + \gamma_i\} & \text{if } s_{lm}^i = -1 \end{cases}$

Computing the multipliers

We use subgradient optimization for this task:

- Start with $\lambda_{lm}^0 = 0$ for all $l, m \in L$.
- $\lambda_{lm}^{i+1} = \begin{cases} \lambda_{lm}^i & \text{if } s_{lm}^i := \overline{y_{lm}} - \overline{y_{ml}} = 0 \\ \max\{-w_{lm}, \lambda_{lm}^i - \gamma_i\} & \text{if } s_{lm}^i = 1 \\ \min\{w_{lm}, \lambda_{lm}^i + \gamma_i\} & \text{if } s_{lm}^i = -1 \end{cases}$
- Stepsize γ_i as in [Held/Karp, 71]

$$\gamma_i = \mu \frac{z_U - z_L}{\sum_{l,m \in L} (s_{lm}^i)^2}$$

Computing the multipliers

We use subgradient optimization for this task:

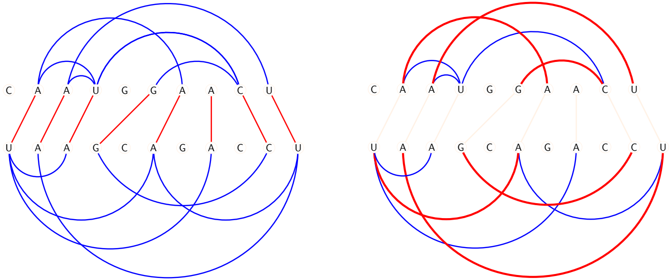
- Start with $\lambda_{lm}^0 = 0$ for all $l, m \in L$.
- $\lambda_{lm}^{i+1} = \begin{cases} \lambda_{lm}^i & \text{if } s_{lm}^i := \overline{y_{lm}} - \overline{y_{ml}} = 0 \\ \max\{-w_{lm}, \lambda_{lm}^i - \gamma_i\} & \text{if } s_{lm}^i = 1 \\ \min\{w_{lm}, \lambda_{lm}^i + \gamma_i\} & \text{if } s_{lm}^i = -1 \end{cases}$
- Stepsize γ_i as in [Held/Karp, 71]

$$\gamma_i = \mu \frac{z_U - z_L}{\sum\limits_{l,m \in L} (s_{lm}^i)^2}$$

Need good upper and lower bounds z_U and z_L.

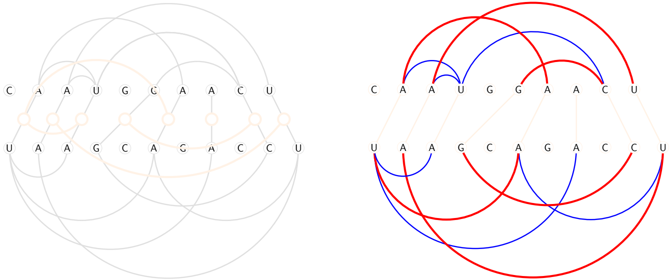
Computing the lower bound

Given: Lines from the solution of the last iteration
Find: Best interaction matches



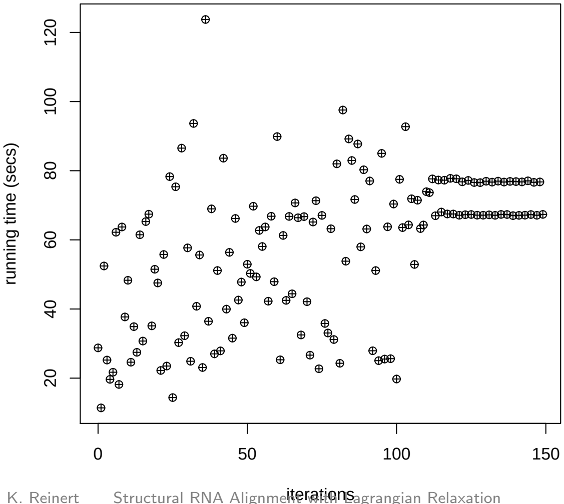
Computing the lower bound

Given: Lines from the solution of the last iteration
Find: Best interaction matches



This is a matching problem!

Computing the lower bound



A stripped-down ILP for the pairwise case

Variables $x \in \{0,1\}^L, y \in \{0,1\}^{L \times L}$

$$\begin{aligned}
\max \quad & \sum_{l \in L} w_l x_l + \sum_{l \in L} \sum_{m \in L} w_{lm} y_{lm} & \\
\text{s.t.} \quad & \sum_{l \in C_L} x_l \leq 1 & \forall C_L \in \mathcal{C}_L \\
& \sum_{m \in L} y_{lm} \leq x_l & \forall\, l \in L \\
& y_{lm} = y_{ml} & \forall l, m \in L \\
& x \in \{0,1\}^L \quad y \in \{0,1\}^{L \times L} &
\end{aligned}$$

Branch-and-Bound framework

- Use the best upper and lower bounds in a branch-and-bound setting
- Branching on the x variables, *i.e.*, the possible alignment edges, yields the enumeration tree
- Results are similar to a branch-and-bound implementation for the quadratic knapsack problem:

Branch-and-Bound framework

- Use the best upper and lower bounds in a branch-and-bound setting
- Branching on the x variables, *i.e.*, the possible alignment edges, yields the enumeration tree
- Results are similar to a branch-and-bound implementation for the quadratic knapsack problem:
  - Bounds have to be very tight to solve problem to optimality
  - Large number of variables has to be fixed to close the gap between lower and upper bound
  - Small improvement of the best lower bound and the optimal solution

Results for Branch-and-Bound

APSI	♮ vars	lb	ub	opt	ratio	time
35	679	181.93	185.18	182.24	(1.00)	1907.45
36	432	194.32	197.43	194.62	(1.00)	252.66
37	597	141.78	142.62	142.17	(1.00)	164.38
38	711	166.01	168.05	166.03	(1.00)	1545.21
39	782	164.40	169.33	165.78	(0.99)	2084.61
40	664	171.74	172.84	171.74	(1.00)	350.71
41	647	190.95	194.16	192.20	(0.99)	1713.44
42	737	167.84	169.31	167.87	(1.00)	816.75
43	873	163.88	165.40	163.93	(1.00)	1782.08
44	682	189.58	192.04	190.05	(1.00)	967.01
45	740	167.33	170.59	168.04	(1.00)	1178.52
46	601	188.49	189.95	188.53	(1.00)	537.22
47	791	183.21	185.76	183.59	(1.00)	1920.89
48	669	181.73	183.55	182.05	(1.00)	1135.23
49	844	177.13	178.73	177.28	(1.00)	1020.95

Computational results

Two main parts:

- A: Testing the limits of the exact multiple sequence-structure approach
- B: Fast computation of heuristic multiple alignments (using T-COFFEE or in a progressive fashion)

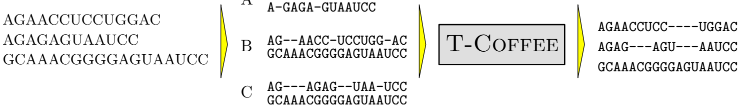
Computational results

Two main parts:

- A: Testing the limits of the exact multiple sequence-structure approach
- B: Fast computation of heuristic multiple alignments (using T-COFFEE or in a progressive fashion)

B: Heuristic multiple alignments

- Compute multiple alignments based on the pairwise case (which is solvable in $\mathcal{O}(n^2)$)
- `LaRA` and `sLaRA` use T-COFFEE:



AGAACCUCCUGGAC
AGAGAGUAAUCC
GCAAACGGGGAGUAAUCC

A
```
AGAACCUCCUGGAC
A-GAGA-GUAAUCC
```

B
```
AG--AACC-UCCUGG-AC
GCAAACGGGGAGUAAUCC
```

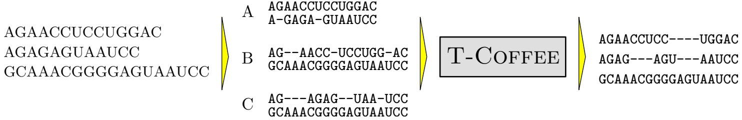
C
```
AG---AGAG--UAA-UCC
GCAAACGGGGAGUAAUCC
```

T-COFFEE

```
AGAACCUCC----UGGAC
AGAG---AGU---AAUCC
GCAAACGGGGAGUAAUCC
```

B: Heuristic multiple alignments

- Compute multiple alignments based on the pairwise case (which is solvable in $\mathcal{O}(n^2)$)
- LaRA and sLaRA use T-COFFEE:



AGAACCUCCUGGAC
AGAGAGUAAUCC
GCAAACGGGGAGUAAUCC

A
AGAACCUCCUGGAC
A-GAGA-GUAAUCC

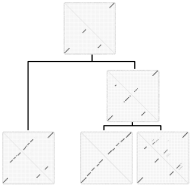
B
AG--AACC-UCCUGG-AC
GCAAACGGGGAGUAAUCC

C
AG---AGAG--UAA-UCC
GCAAACGGGGAGUAAUCC

T-COFFEE

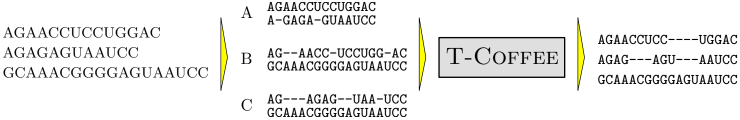
AGAACCUCC----UGGAC
AGAG---AGU---AAUCC
GCAAACGGGGAGUAAUCC

- pLaRA and psLaRA are progressive tools:



B: Heuristic multiple alignments

- Compute multiple alignments based on the pairwise case (which is solvable in $\mathcal{O}(n^2)$)
- LaRA and sLaRA use T-Coffee:



AGAACCUCCUGGAC
AGAGAGUAAUCC
GCAAACGGGGAGUAAUCC

A
AGAACCUCCUGGAC
A-GAGA-GUAAUCC

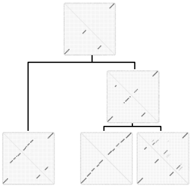
B
AG--AACC-UCCUGG-AC
GCAAACGGGGAGUAAUCC

C
AG---AGAG--UAA-UCC
GCAAACGGGGAGUAAUCC

T-Coffee

AGAACCUCC----UGGAC
AGAG---AGU---AAUCC
GCAAACGGGGAGUAAUCC

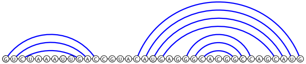
- pLaRA and psLaRA are progressive tools:



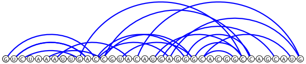
- Currently C^{++} using LEDA, part of the LiSA-framework (`http://www.planet-lisa.net`)

Input and training

- Interaction edges. Two modes:
  - Known structure:

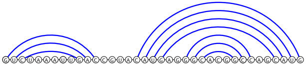


  - Weights based on base pair probabilities [McCaskill, 90], similar to PMCOMP [Hofacker, 04]:

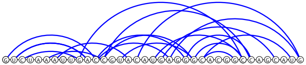


Input and training

- Interaction edges. Two modes:
    - Known structure:



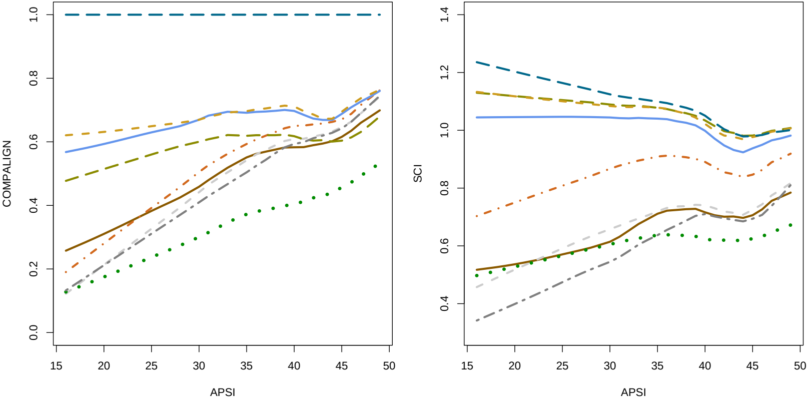
- Weights based on base pair probabilities [McCaskill, 90], similar to PMComp [Hofacker, 04]:



- Computational experiments performed on the BRAliBase 2.1 benchmark dataset, the MASTR data set [Lindgreen, 07] serves as the training set
- We only consider instances of an average pairwise sequence identity $< 50\%$

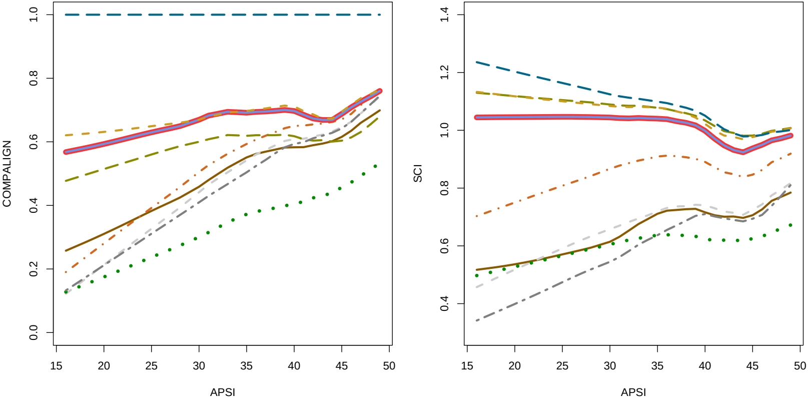
BRAliBase 2.1 - k2

- 2 input sequences per instance, 2251 instances:



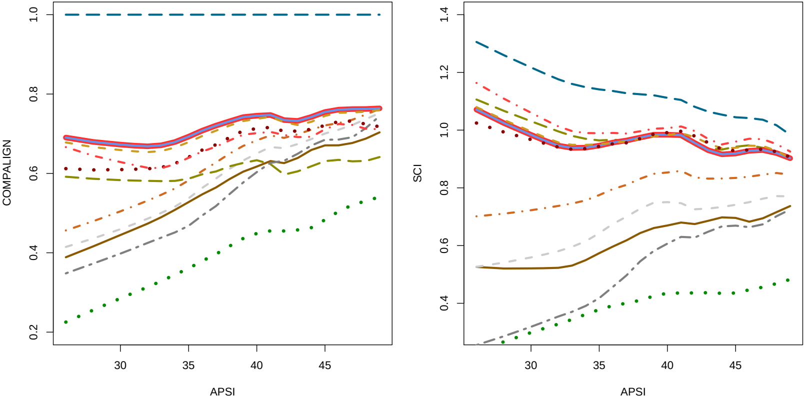
BRALIBASE 2.1 - k2

- 2 input sequences per instance, 2251 instances:



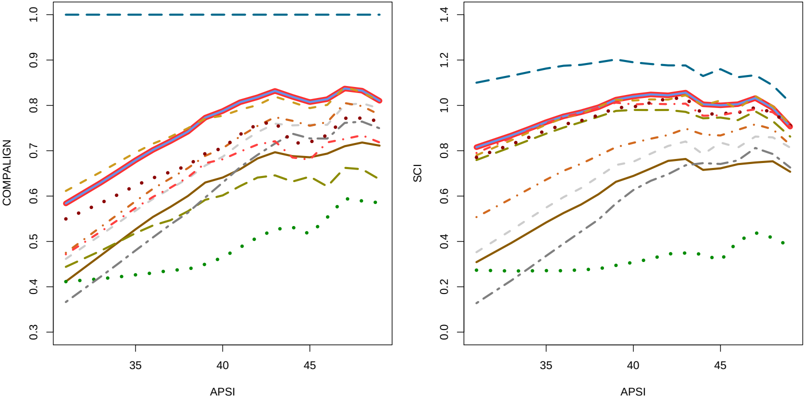
BRALIBASE 2.1 - k3

- 3 input sequences per instance, 1048 instances:



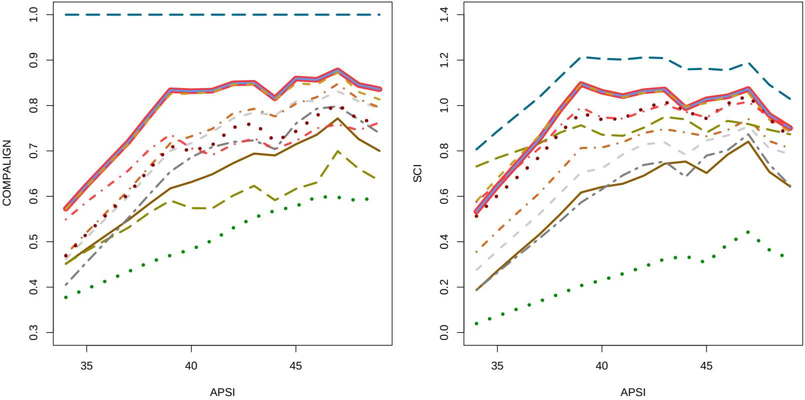
BRALIBASE 2.1 - k5

- 5 input sequences per instance, 512 instances:



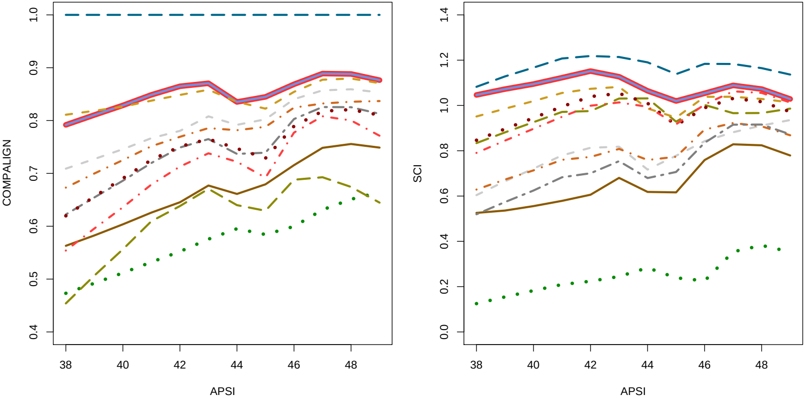
BRALIBASE 2.1 - k7

- 7 input sequences per instance, 323 instances:



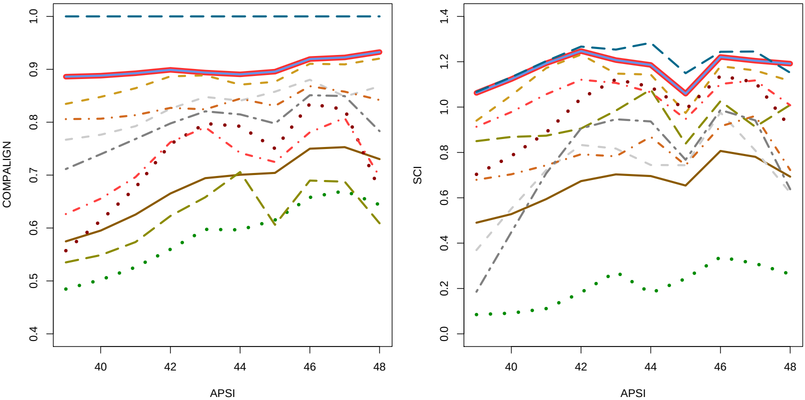
BRALIBASE 2.1 - k10

- 10 input sequences per instance, 189 instances:



BRALIBASE 2.1 - k15

- 15 input sequences per instance, 123 instances:



Comparison of running times

- Running time (in secs) for all 2251 pairwise alignment instances (k2) of the BRALIBASE dataset:

	running time	⌀ SPS	⌀ SCI
LARA	3157.74	0.68	0.98
sLARA	5234.15	0.69	1.01
FOLDALIGN	10360.44	0.61	1.02
MURLET	9575.54	0.60	0.73
MARNA	56434.11	0.42	0.63
MXSCARNA	478.74	0.64	0.87
STRAL	18.72	0.58	0.71

Insights

- With an increasing number of input sequences, the T-Coffee approach pays off (quadratic instead of linear information)

Insights

- With an increasing number of input sequences, the T-Coffee approach pays off (quadratic instead of linear information)
- Several parameter sets yield good results, a sound parameter training procedure is needed

Insights

- With an increasing number of input sequences, the T-COFFEE approach pays off (quadratic instead of linear information)
- Several parameter sets yield good results, a sound parameter training procedure is needed
- All the other tested programs do not allow pseudoknots, `LaRA` does, but:

Insights

- With an increasing number of input sequences, the T-Coffee approach pays off (quadratic instead of linear information)
- Several parameter sets yield good results, a sound parameter training procedure is needed
- All the other tested programs do not allow pseudoknots, `LaRA` does, but:
    - algorithms for computing pseudoknotted structures and base pair probabilities are computationally expensive
    - we are lacking sound energy parameters

Insights

- With an increasing number of input sequences, the T-COFFEE approach pays off (quadratic instead of linear information)
- Several parameter sets yield good results, a sound parameter training procedure is needed
- All the other tested programs do not allow pseudoknots, `LaRA` does, but:
  - algorithms for computing pseudoknotted structures and base pair probabilities are computationally expensive
  - we are lacking sound energy parameters
- We did not report on:
  - alternative ways to set the penalty vector (the bundle instead of subgradient method)
  - the performance of our approach within a branch-and-bound framework

Conclusions

- We presented a novel formulation for multiple sequence-structure alignments

Conclusions

- We presented a novel formulation for multiple sequence-structure alignments
- We incorporated stacking energies into the model

Conclusions

- We presented a novel formulation for multiple sequence-structure alignments
- We incorporated stacking energies into the model
- We compared our implementations to various other current programs, we perform best on the BRALIBASE 2.1 benchmark set

Conclusions

- We presented a novel formulation for multiple sequence-structure alignments
- We incorporated stacking energies into the model
- We compared our implementations to various other current programs, we perform best on the BRALIBASE 2.1 benchmark set
- We implemented variants based on the bundle method and branch-and-bound

THANKS FOR YOUR ATTENTION!

Check out

`http://www.planet-lisa.net`