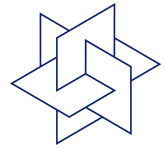
LaRA: RNA Sequence-Structure Alignments by Lagrangian Relaxation

Gunnar W. Klau

(joint work with Markus Bauer and Knut Reinert)



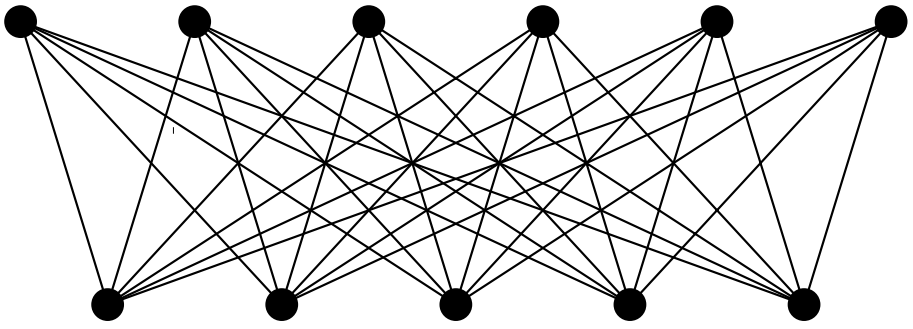
DFG Research Center MATHEON
Mathematics for key technologies

Lecture P1, winter 07/08
Freie Universität Berlin

20 Dec 2007

Graph-Based View on Sequence-Structure Alignment

Graph-Theoretical Reformulation



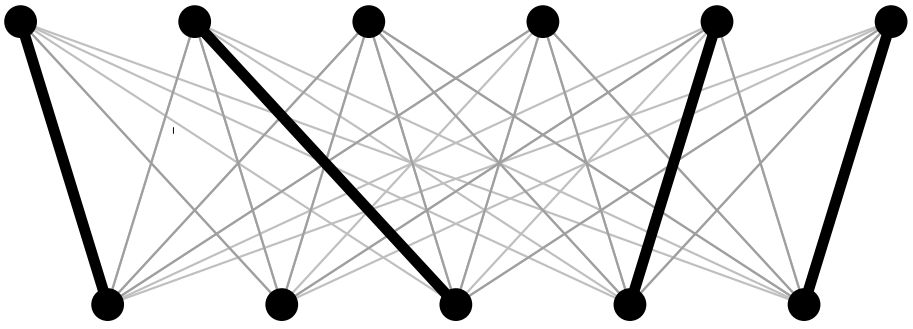
A C C G U G

G A U C C

Pairwise sequence alignment problem:

Find alignment of maximum weight (award matches, penalize mismatches and gaps)

Graph-Theoretical Reformulation



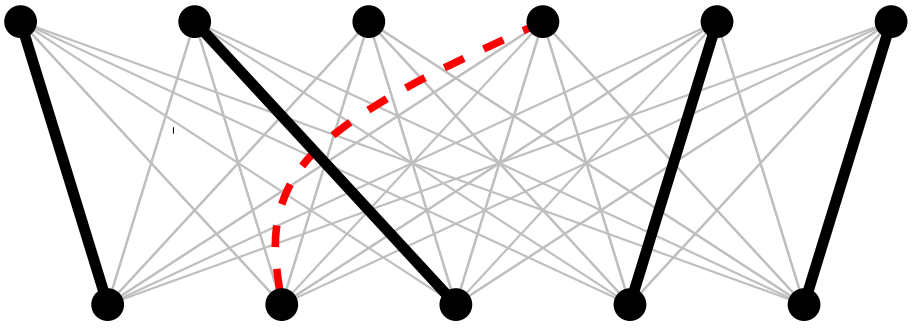
A - C C G U G

G A U - - C C

Pairwise sequence alignment problem:

Find alignment of maximum weight (award matches, penalize mismatches and gaps)

Graph-Theoretical Reformulation



A - C C G U G

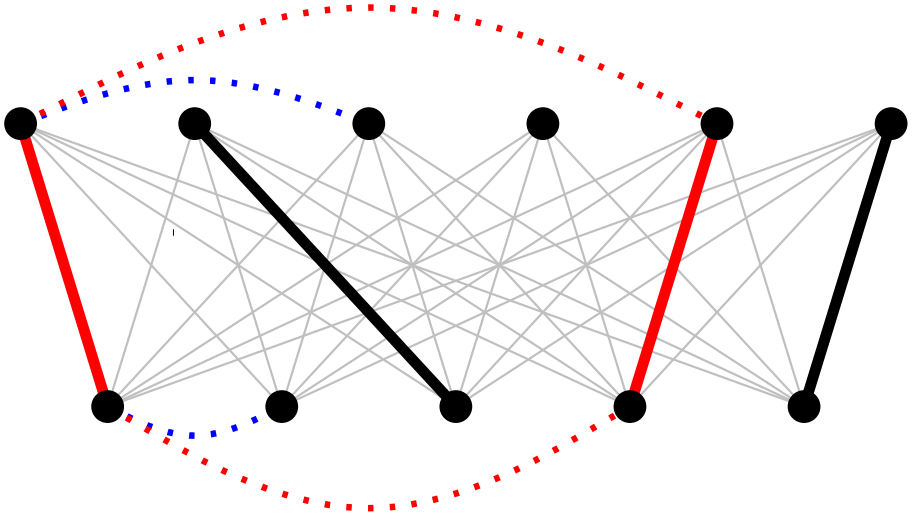
G A U - - C C

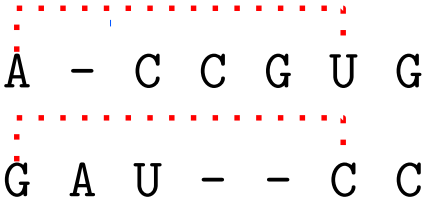
Pairwise sequence alignment problem:

Find alignment of maximum weight (award matches, penalize mismatches and gaps)

Graph-Theoretical Reformulation

interaction matches $M := \left\{ \{l, m\} \in \binom{L}{2} \mid l \text{ and } m \text{ do not cross} \right\}$





Pairwise sequence-structure alignment problem:

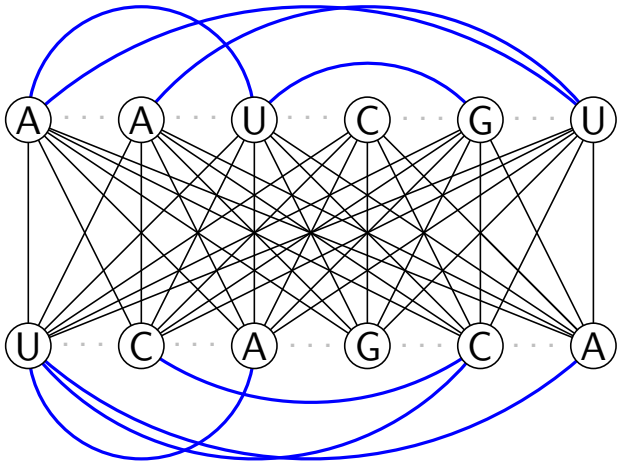
Find structural alignment of maximum weight (alignment costs + awards for realized interaction matches)

Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

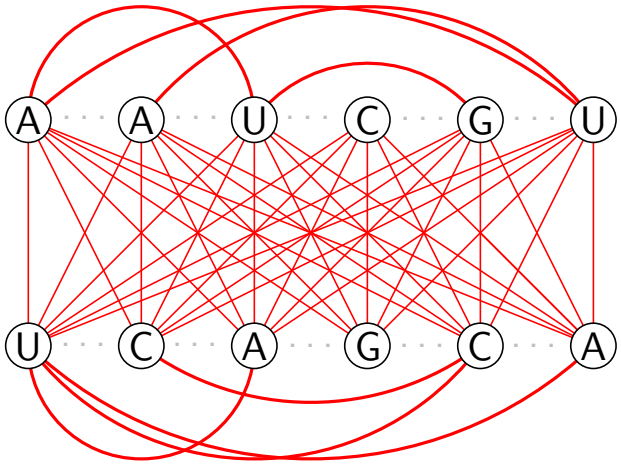


Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

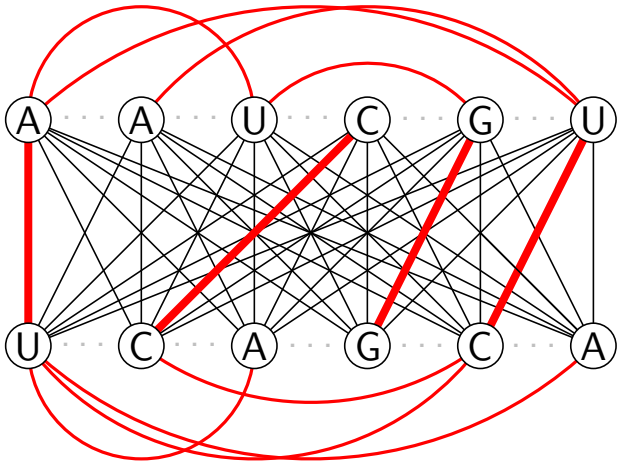


Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

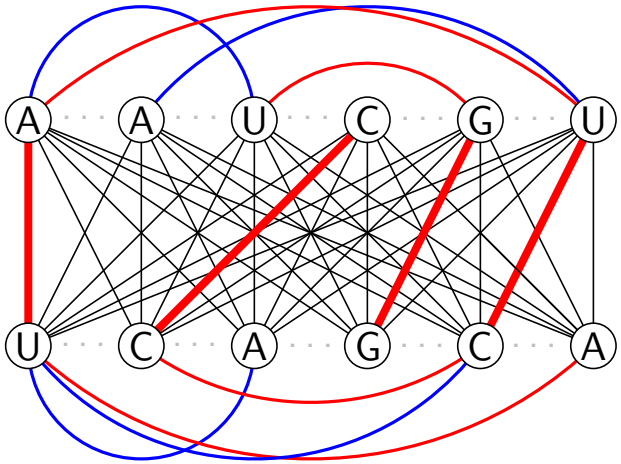


Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

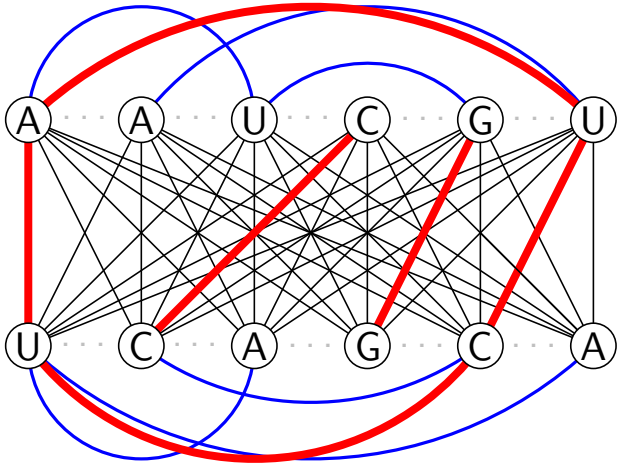


Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

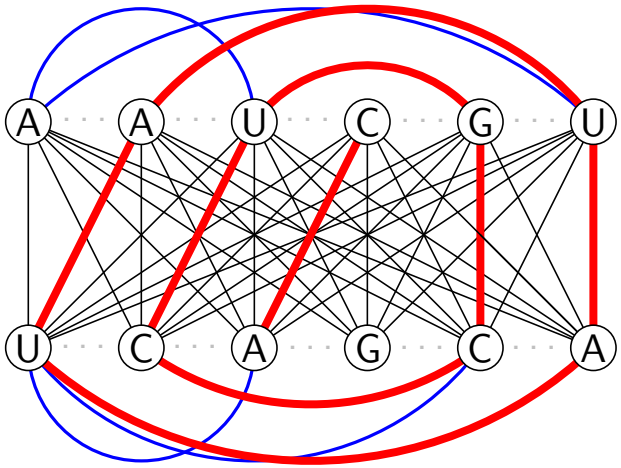


Graph-Theoretical Reformulation

Given: Match graph $G_M = (V_1 \cup V_2, E_1 \cup E_2 \cup L)$, matches M, and weights $w^{L \cup M}$

Find: Lines $L' \subseteq L$ and matches $M' \subseteq M$ such that

1. $\sum_{l \in L'} w_l + \sum_{\{l,m\} \in M'} w_{lm}$ is maximal.
2. Lines in L' are conflict-free.
3. Every line is incident to at most one interaction match
4. Matches in M' are realized, i.e., $\forall \{l,m\} \in M' : l, m \in L'$.

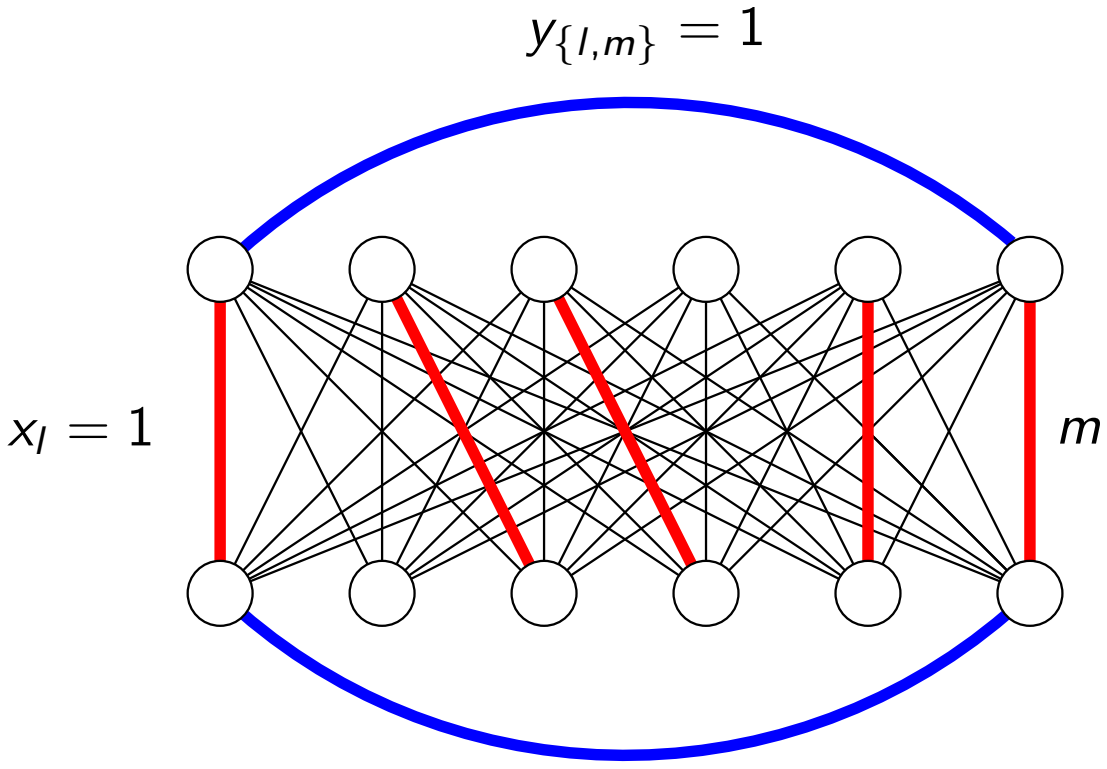


Integer Linear Programming Formulation

Integer Linear Programming Formulation

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

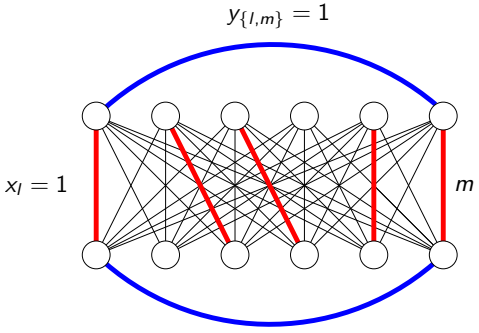
$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad y_{\{l,m\}} = \begin{cases} 1 & \text{match } \{l,m\} \in M' \\ 0 & \text{otherwise} \end{cases}$$



Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad y_{\{l,m\}} = \begin{cases} 1 & \text{match } \{l,m\} \in M' \\ 0 & \text{otherwise} \end{cases}$$



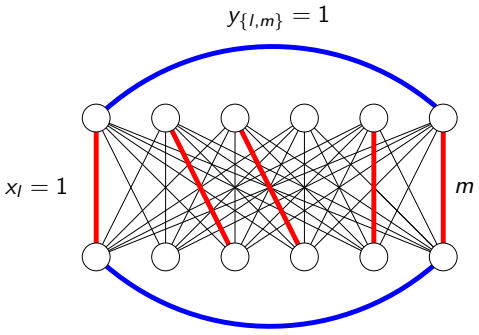
$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{\{l,m\} \in M} w_{lm} \cdot y_{\{l,m\}}$$

$$x \in \{0,1\}^L, \quad y \in \{0,1\}^M$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad y_{\{l,m\}} = \begin{cases} 1 & \text{match } \{l,m\} \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{\{l,m\} \in M} w_{lm} \cdot y_{\{l,m\}}$$

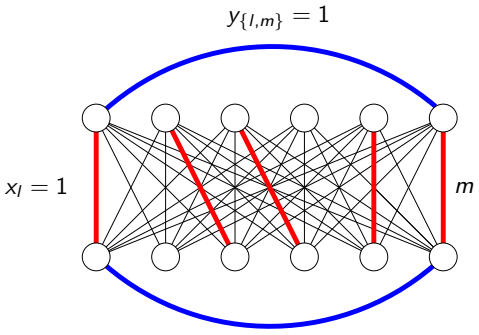
$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

$$x \in \{0,1\}^L, \quad y \in \{0,1\}^M$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad y_{\{l,m\}} = \begin{cases} 1 & \text{match } \{l,m\} \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{\{l,m\} \in M} w_{lm} \cdot y_{\{l,m\}}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

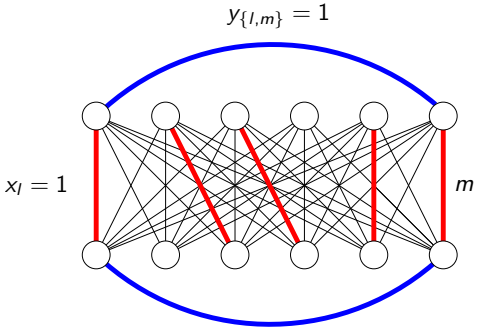
$$\sum_{m \in L} y_{\{l,m\}} \leq x_l \qquad \forall l \in L$$

$$x \in \{0,1\}^L, \quad y \in \{0,1\}^M$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad y_{\{l,m\}} = \begin{cases} 1 & \text{match } \{l,m\} \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{\{l,m\} \in M} w_{lm} \cdot y_{\{l,m\}}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

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

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

$$x \in \{0,1\}^L, \quad y \in \{0,1\}^M$$

Solving the ILP with Lagrangian Relaxation

Unifying Graph-Based View Revisited

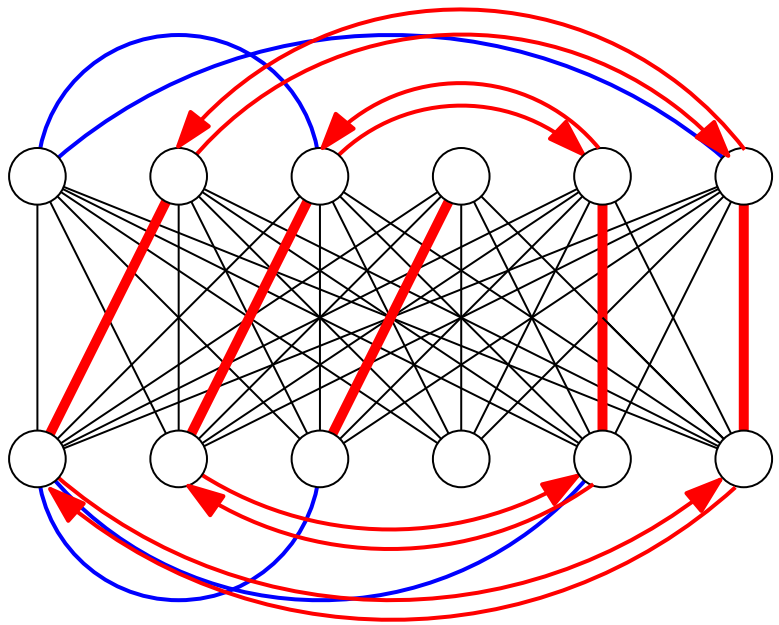
Split matches into directed matches

$$\vec{M} := \{(l,m),(m,l) \mid \{l,m\} \in M\} \quad .$$

A directed match (l,m) is realized iff $l \in L'$.

Select $L' \subseteq L$ and $M' \subseteq \vec{M}$ with

1. $\sum_{l \in L'} w_l + \sum_{(l,m) \in M'} \vec{w}_{(l,m)}$ maximal
2. Lines in L' are conflict-free
3. Matches in M' are realized
4. $(l,m) \in M' \Leftrightarrow (m,l) \in M'$
5. $\vec{w}_{(l,m)} + \vec{w}_{(m,l)} = w_{\{l,m\}}$



Idea is due to [Caprara & Lancia, 04], who did this for CMO

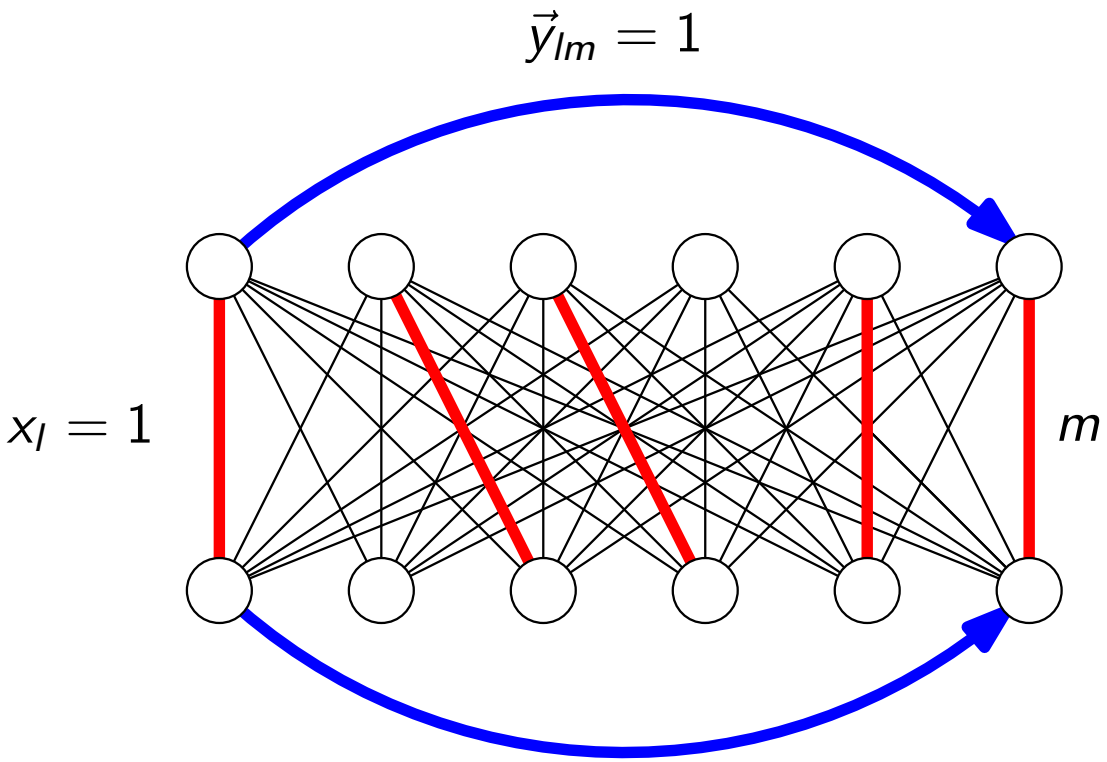
Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$

Weights

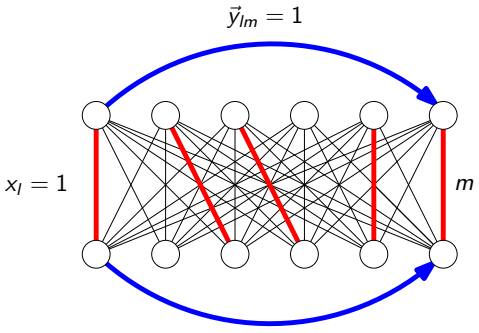
$$\vec{w}_{lm} = \vec{w}_{ml} = \frac{1}{2} w_{lm}$$



Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{(l,m) \in \vec{M}} \vec{w}_{lm} \cdot \vec{y}_{(l,m)}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

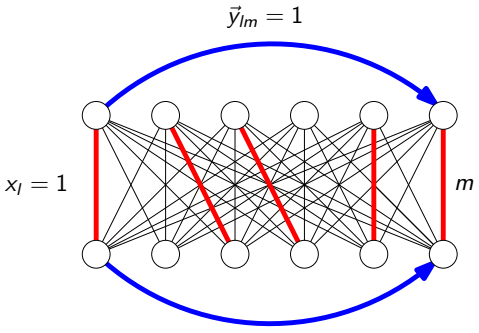
$$\sum_{m \in L} \vec{y}_{(l,m)} \leq x_l \qquad \forall l \in L$$

$$x \in \{0,1\}^L, \quad \vec{y} \in \{0,1\}^{\vec{M}}$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{(l,m) \in \vec{M}} \vec{w}_{lm} \cdot \vec{y}_{(l,m)}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

$$\sum_{m \in L} \vec{y}_{(l,m)} \leq x_l \qquad \forall l \in L$$

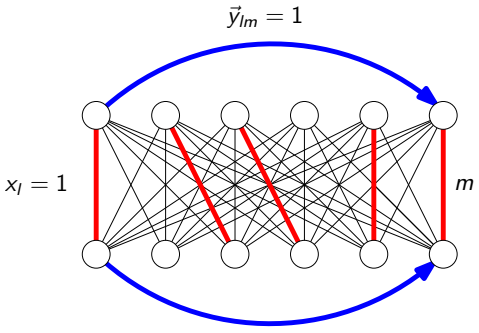
$$\vec{y}_{(l,m)} = \vec{y}_{(m,l)} \qquad \forall (l,m) \in \vec{M}, l < m$$

$$x \in \{0,1\}^L, \quad \vec{y} \in \{0,1\}^{\vec{M}}$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{(l,m) \in \vec{M}} \vec{w}_{lm} \cdot \vec{y}_{(l,m)}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

$$\sum_{m \in L} \vec{y}_{(l,m)} \leq x_l \qquad \forall l \in L$$

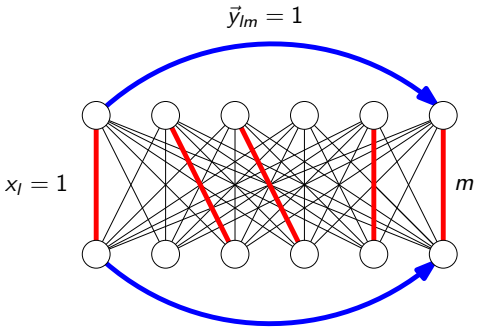
$$\cancel{\vec{y}_{(l,m)} = \vec{y}_{(m,l)}} \qquad \forall (l,m) \in \vec{M}, l < m$$

$$x \in \{0,1\}^L, \quad \vec{y} \in \{0,1\}^{\vec{M}}$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{(l,m) \in \vec{M}} \vec{w}_{lm} \cdot \vec{y}_{(l,m)} + \sum_{(l,m) \in \vec{M}, l<m} \lambda_{lm} (\vec{y}_{(l,m)} - \vec{y}_{(m,l)})$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

$$\sum_{m \in L} \vec{y}_{(l,m)} \leq x_l \qquad \forall l \in L$$

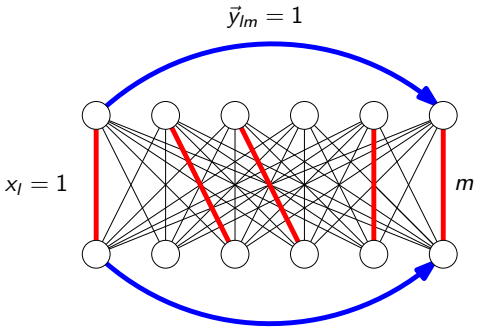
$$\vec{y}_{(l,m)} = \vec{y}_{(m,l)} \qquad \forall (l,m) \in \vec{M}, l < m$$

$$x \in \{0,1\}^L, \quad \vec{y} \in \{0,1\}^{\vec{M}}$$

Integer Linear Programming Formulation

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

$$x_l = \begin{cases} 1 & l \in L' \\ 0 & \text{otherwise} \end{cases} \qquad \vec{y}_{(l,m)} = \begin{cases} 1 & \text{match } (l,m) \in M' \\ 0 & \text{otherwise} \end{cases}$$



$$\max \sum_{l \in L} w_l \cdot x_l + \sum_{(l,m) \in \vec{M}} (\vec{\lambda}_{lm} + \vec{w}_{lm}) \vec{y}_{(l,m)}$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \qquad \forall \text{ sets of mutually crossing lines } C$$

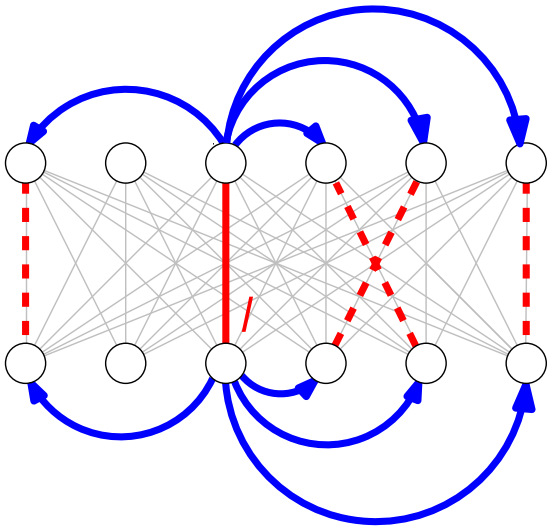
$$\sum_{m \in L} \vec{y}_{(l,m)} \leq x_l \qquad \forall l \in L$$

$$\cancel{\vec{y}_{(l,m)} = \vec{y}_{(m,l)}} \qquad \forall (l,m) \in \vec{M}, l < m$$

$$x \in \{0,1\}^L, \quad \vec{y} \in \{0,1\}^{\vec{M}}$$

Solving the Relaxation

Local view:

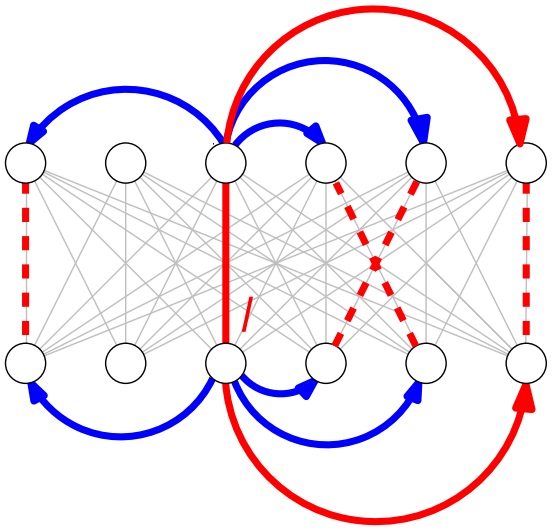


We can easily compute the profit of a line l:

$$p_l = w_l + \max_{\substack{m \in L, \\ (l,m) \in \vec{M}}} (\vec{\lambda}_{lm} + \vec{w}_{lm})$$

Solving the Relaxation

Local view:

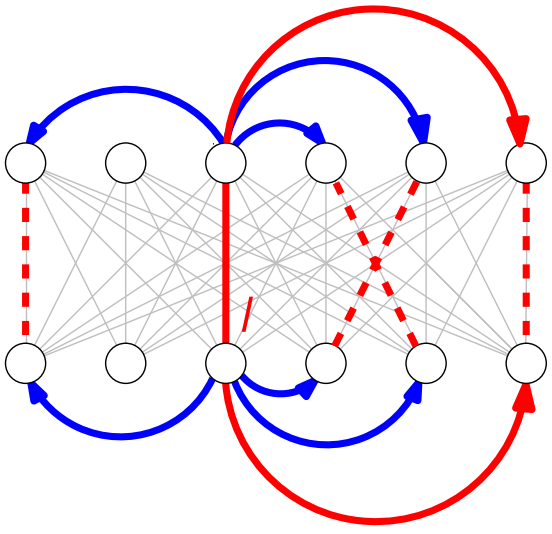


We can easily compute the profit of a line l:

$$p_l = w_l + \max_{\substack{m \in L, \\ (l,m) \in \vec{M}}} (\vec{\lambda}_{lm} + \vec{w}_{lm})$$

Solving the Relaxation

Local view:



Switch back to global view:

$$\max \sum_{l \in L} p_l \cdot x_l$$

$$\text{s.t.} \sum_{l \in C} x_l \leq 1 \quad \forall C$$

$$x \in \{0,1\}^L$$

This is classical sequence alignment!

We can easily compute the profit of a line l:

$$p_l = w_l + \max_{\substack{m \in L, \\ (l,m) \in \vec{M}}} (\vec{\lambda}_{lm} + \vec{w}_{lm})$$

Computing the Multipliers

We use subgradient optimization for this task:

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

Computing the Multipliers

We use subgradient optimization for this task:

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

Computing the Multipliers

We use subgradient optimization for this task:

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

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

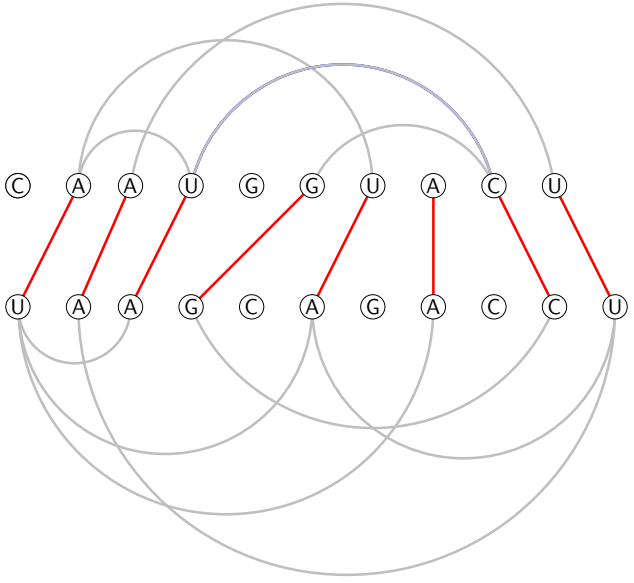
Need good upper and lower bounds z_U and z_L.

- z_U = lowest relaxation sol. value seen so far
- z_L = ?

Computing the Lower Bound z_L

In each iteration, we would like to compute a new good structural alignment.

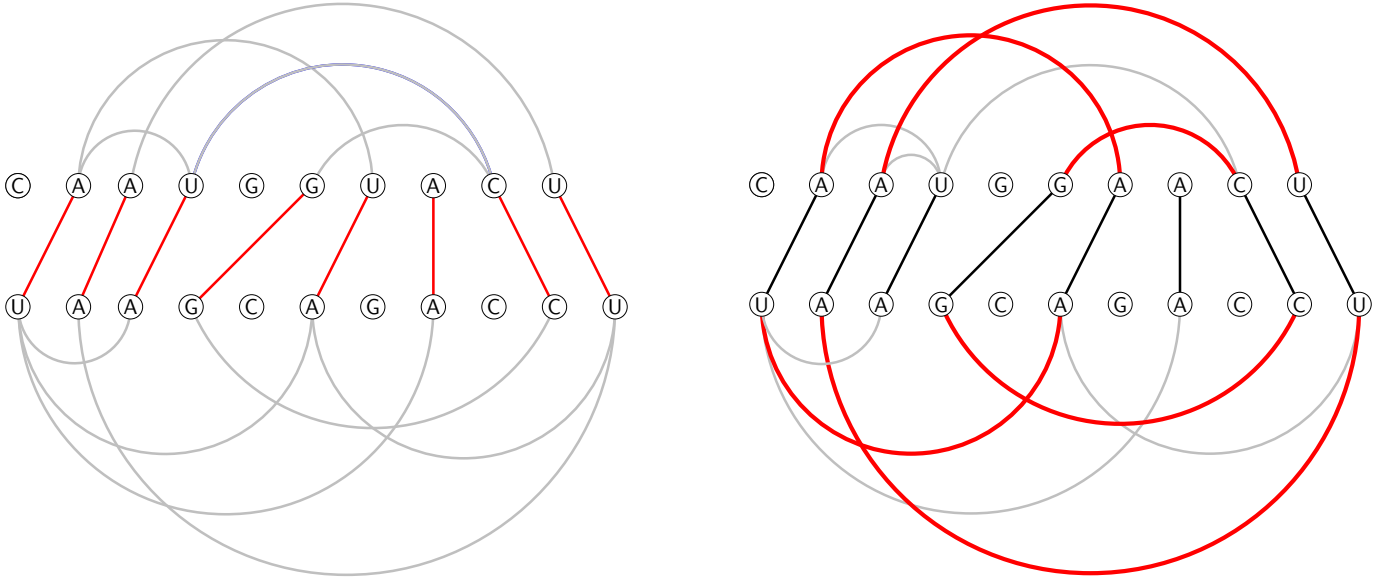
Given: Alignment from the solution of the last iteration
Find: Best completion with interaction matches



Computing the Lower Bound z_L

In each iteration, we would like to compute a new good structural alignment.

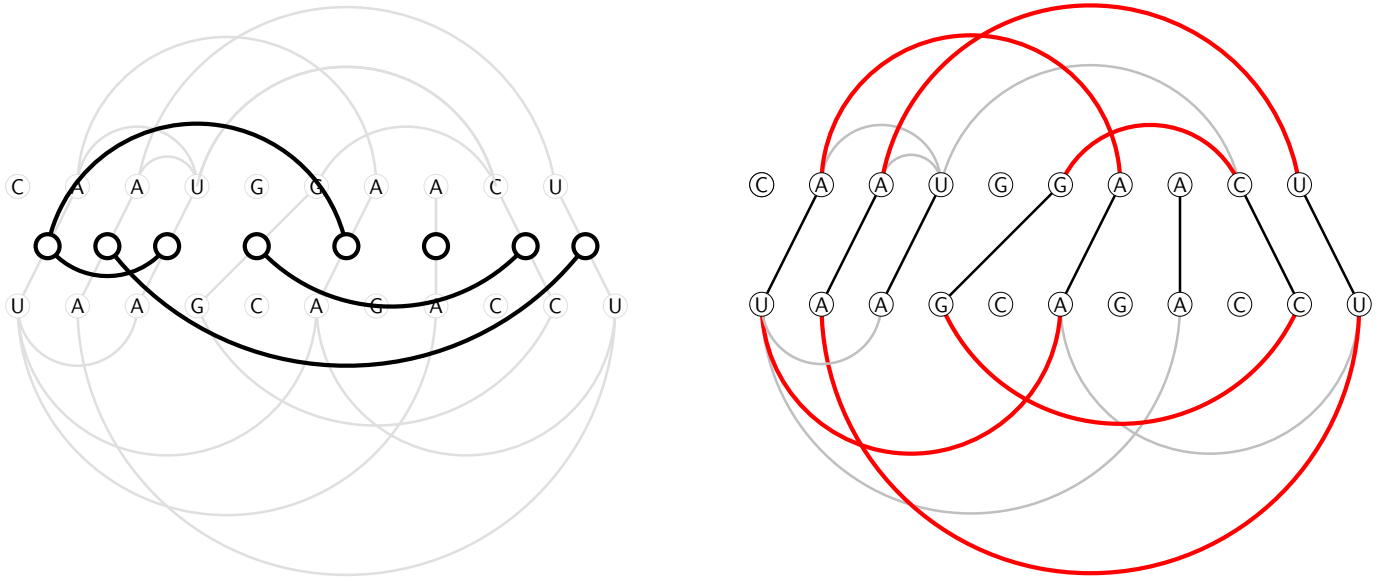
Given: Alignment from the solution of the last iteration
Find: Best completion with interaction matches



Computing the Lower Bound z_L

In each iteration, we would like to compute a new good structural alignment.

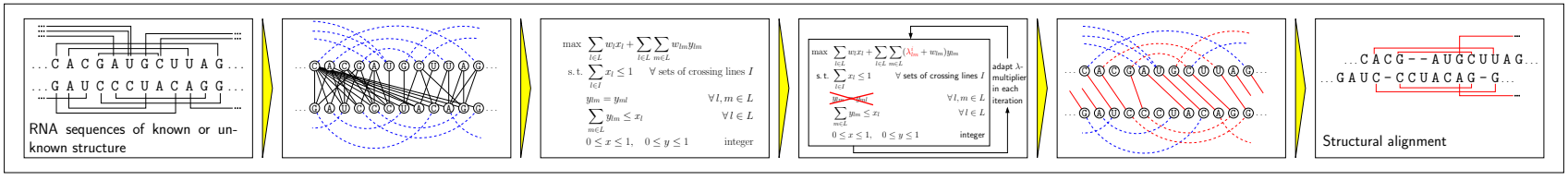
Given: Alignment from the solution of the last iteration
Find: Best completion with interaction matches



This is the maximum weight matching problem in general graphs!

T-Lara: Overall Approach

- Input: k RNA sequences
- We compute $\binom{k}{2}$ pairwise structural alignments by Lagrangian relaxation:

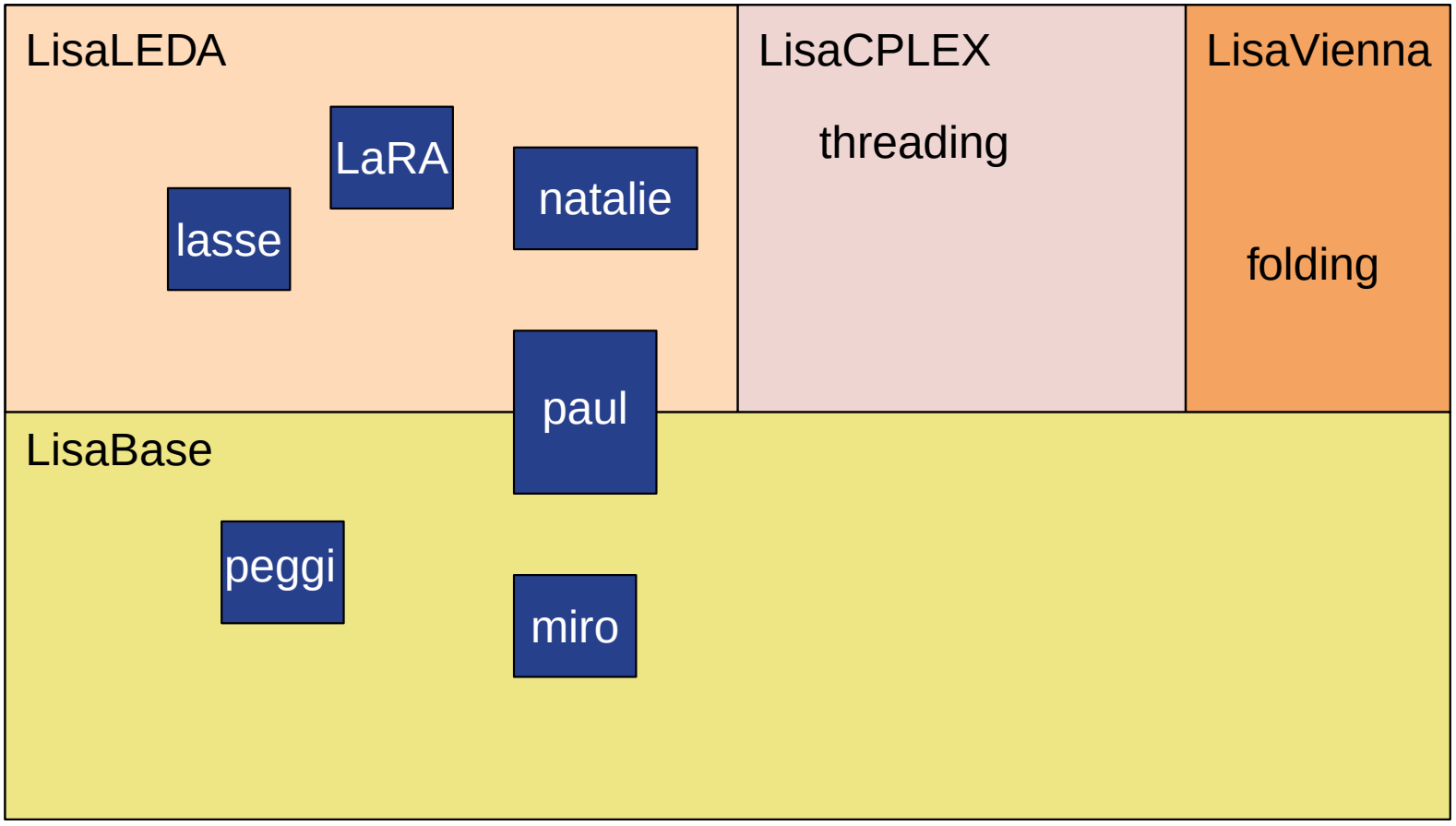


- and build a library for T-Coffee [Notredame et al., 00]
  - The library is a collection of local information contained in the $\binom{k}{2}$ pairwise alignments
  - T-Coffee is a popular progressive alignment tool that respects the local information.
- Output: Multiple alignment computed by T-Coffee

Computational Results

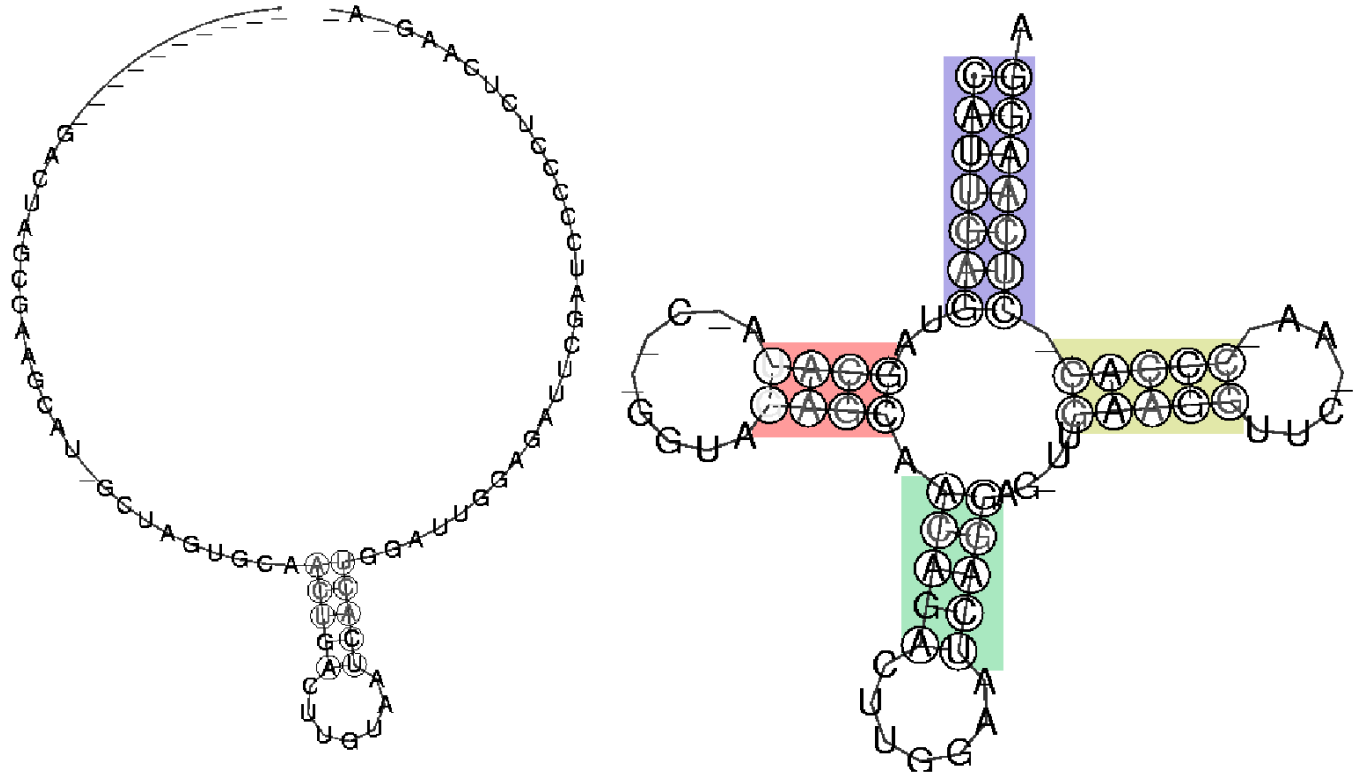
Computational Results

LiSA: Library of Structural Alignment algorithms



C++, open source, `http://www.planet-lisa.net`

Computational Results: LaRA



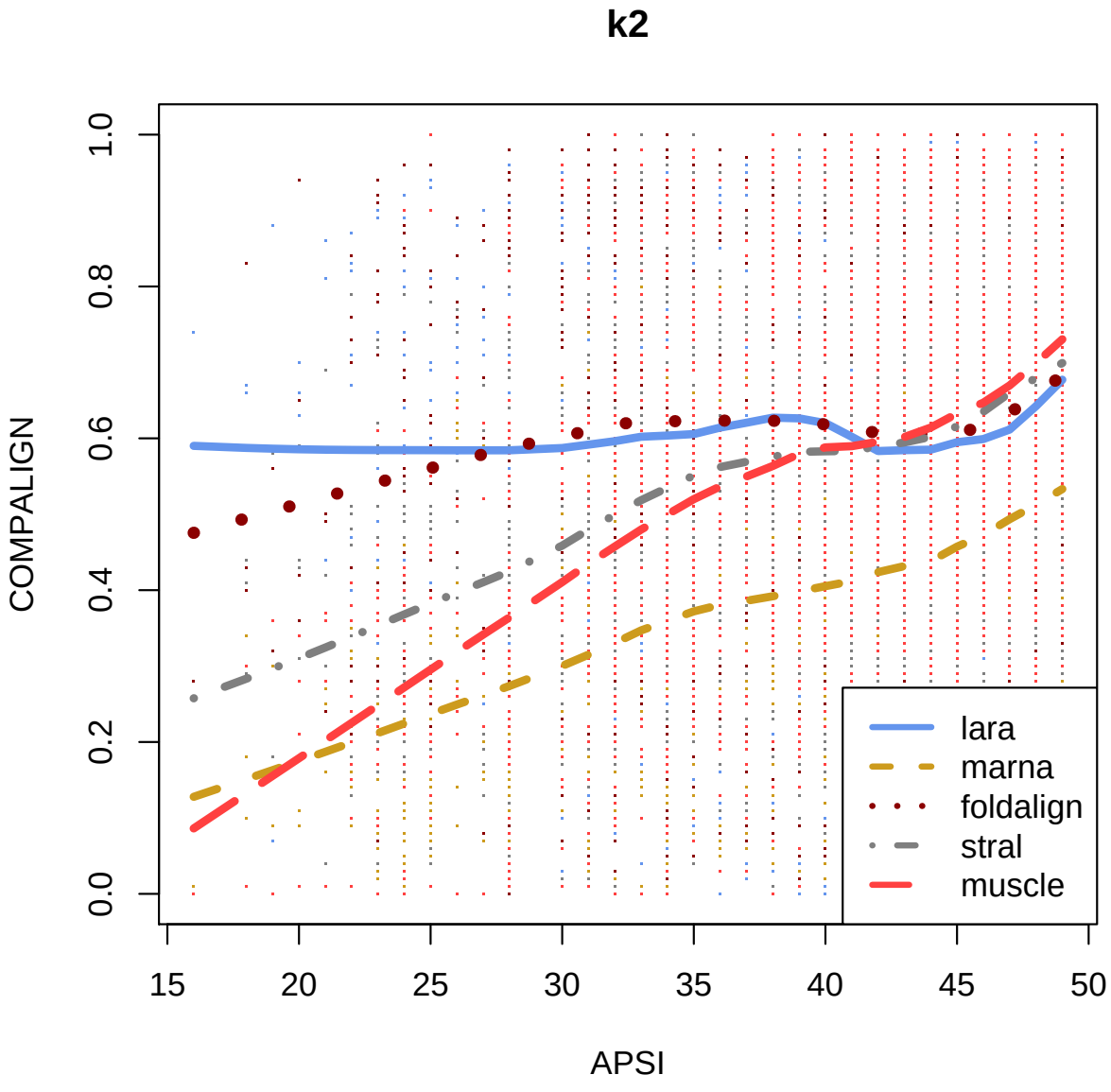
```
M93388.1_1185-1 CAGUUUAGUAGUUU-AA-UGAAGAAUGCUAGCUUUGGGGGUUGGAG-GUCUCUGG---UU-UGGAG-UUGGCUGG
AF217350.1_193- GCUUGAGUAGCAAAGC-GGUUAAUGCUUGAGAUUUAGGUUCUCACA-UCAAAGGUUCAAG-CCCUUUCUCUAGUU
L07095.1_15292- GUCUUGAUAGUAUA----AACAUUACUCUGGUCUUGUAAACCUGAA-AUGAAGAUC-UUC-UCUUC-UCAAGACA
AF233324.1_3722 CGGCGAGUAGCGCAGCUUGGUAGCGCAACUGGUUUGGGACCAGUGGGUCGGAGGUUCGAAUCCUCU-CUCGCCGA
D00558.1_58-130 GUCUGAUUAGCGCAACU-GGCAGAGCAACUGACUCUUAAUCAGUGGGUUGUGGGUUCGAUUCCCAC-AUCAGGCA
X52392.1_9016-9 CAUUAAGAAGCUAUGCA-CC---AGCACUAGCCUUUUAAGCUAGAGAGAGGGGACACCC--UCCCC-CUUAAUGA
AB042524.1_9391 ACUCCCUUAGUAUA----AUUAAUAUAACUGACUUCCAAUUAGUAGA-UUCUGAAUA-AACCCAG-AAGAGAGUA
```

Benchmark study: LaRA competitive with/outperforms alternative approaches.

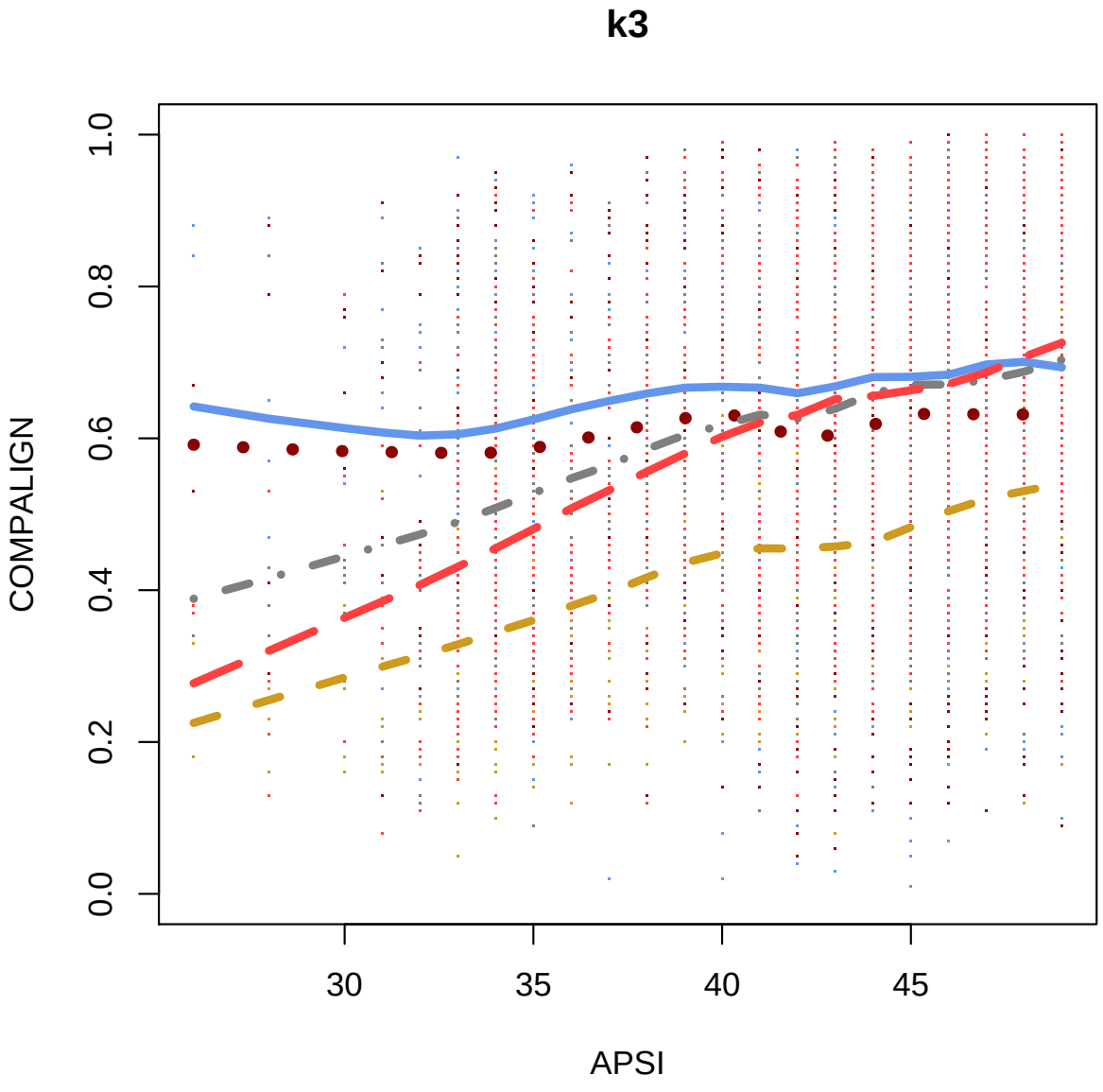
Computational Results: LaRA

- LaRA: Lagrangian RNA Alignments
- Benchmark set of manually curated structural alignments
  - BraliBase 2.1 [Wilm, 06]
  - contains alignments of 2, 3, 5, 7, 10, and 15 sequences
  - classified according to average pairwise sequence identity (APSI)
- Comparison to state-of-the-art tools: MARNA, STRAL, FoldalignM, Muscle
- Quality assessment by comparing *sum-of-pairs score* (COMPALIGN) (comparison to reference alignment)

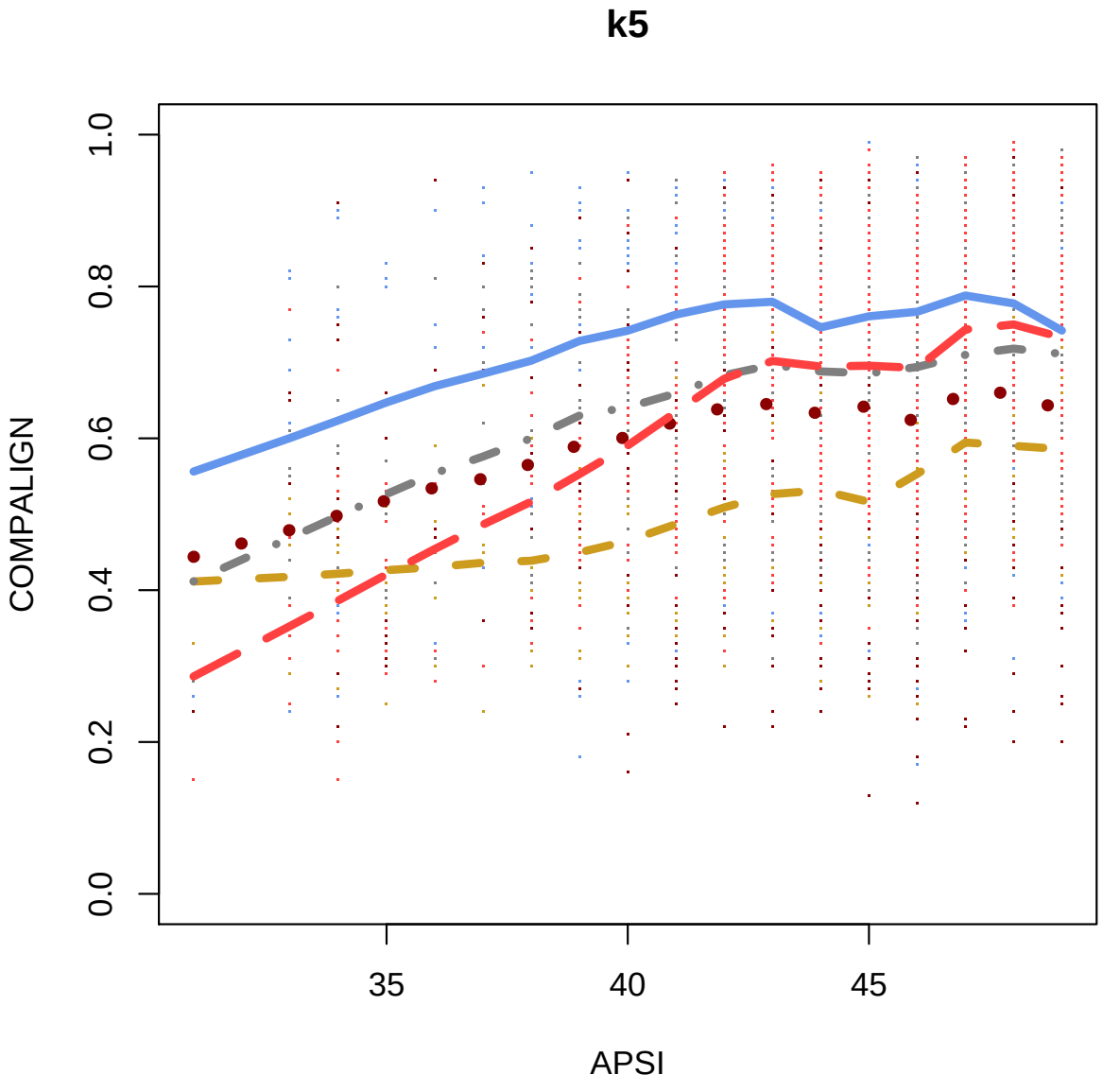
Computational Results: LaRA



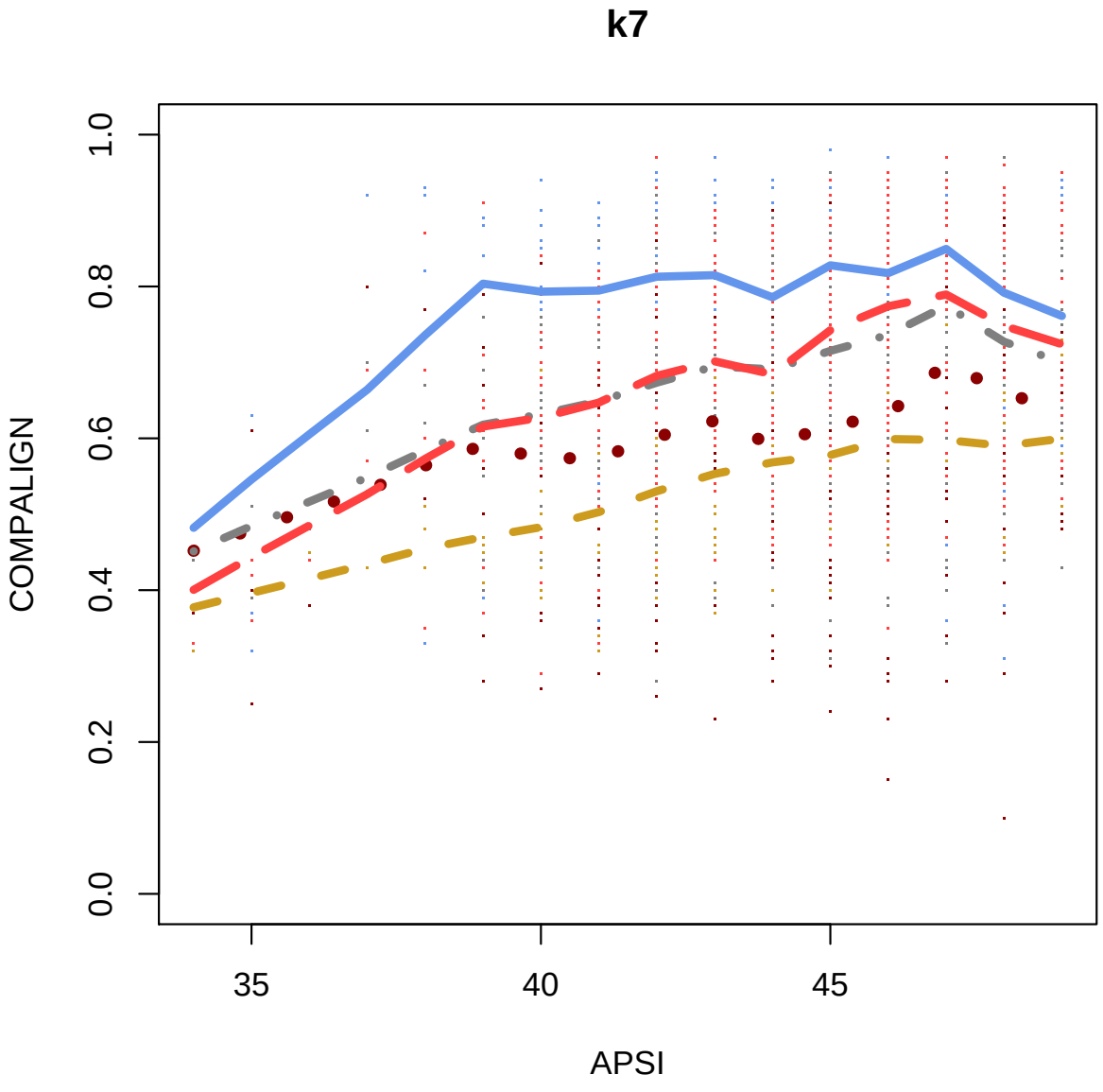
Computational Results: LaRA



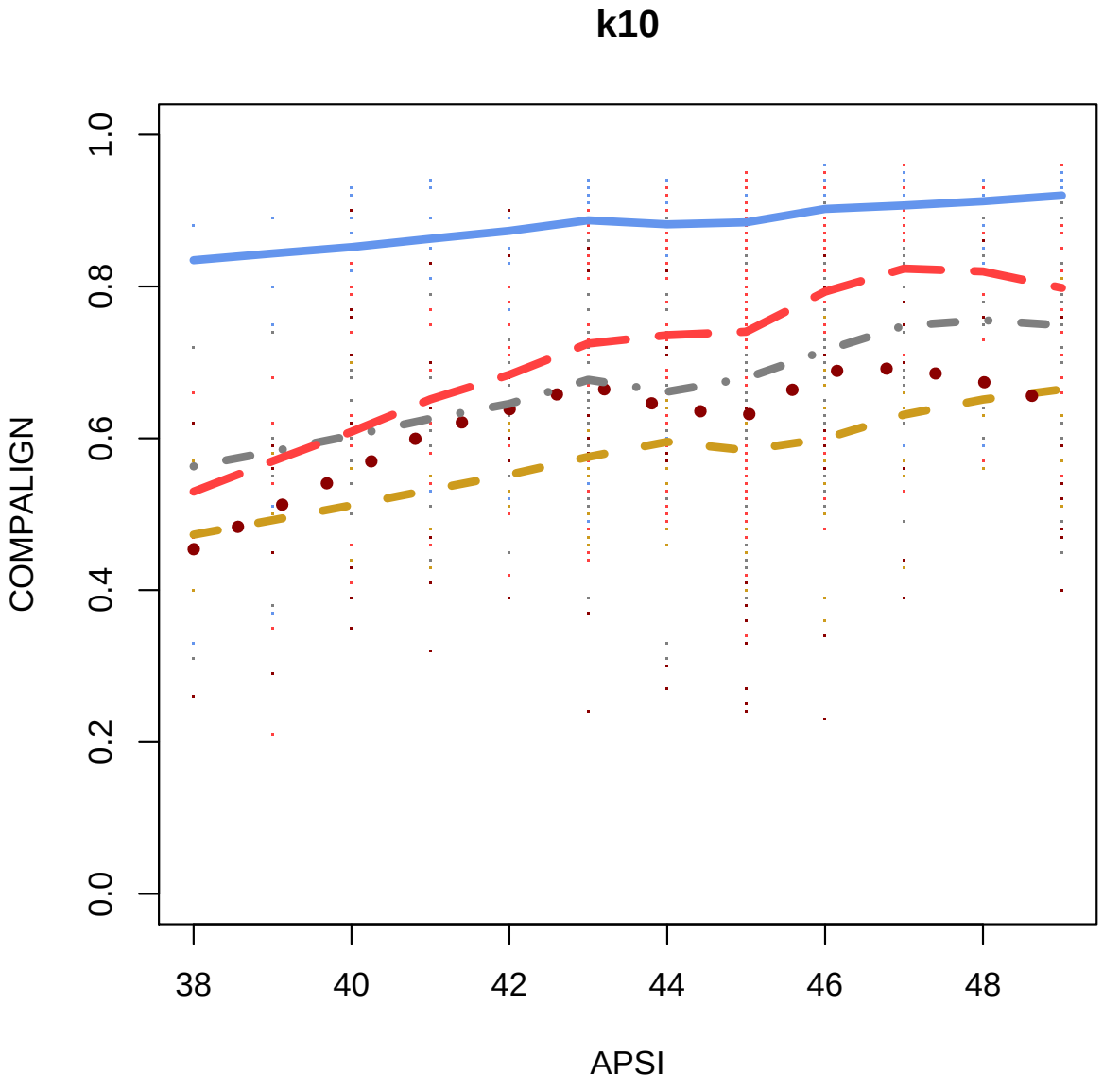
Computational Results: LaRA



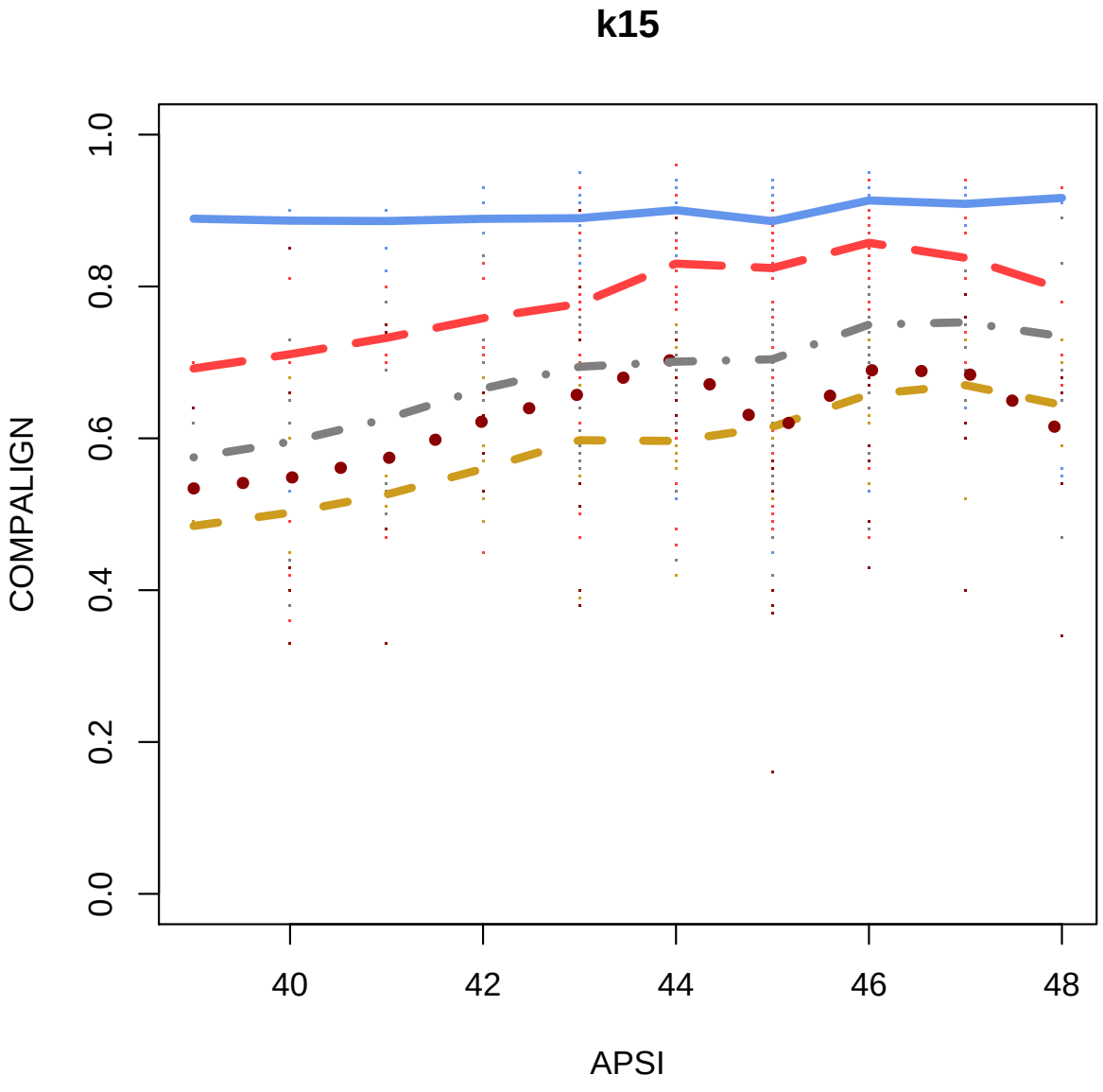
Computational Results: LaRA



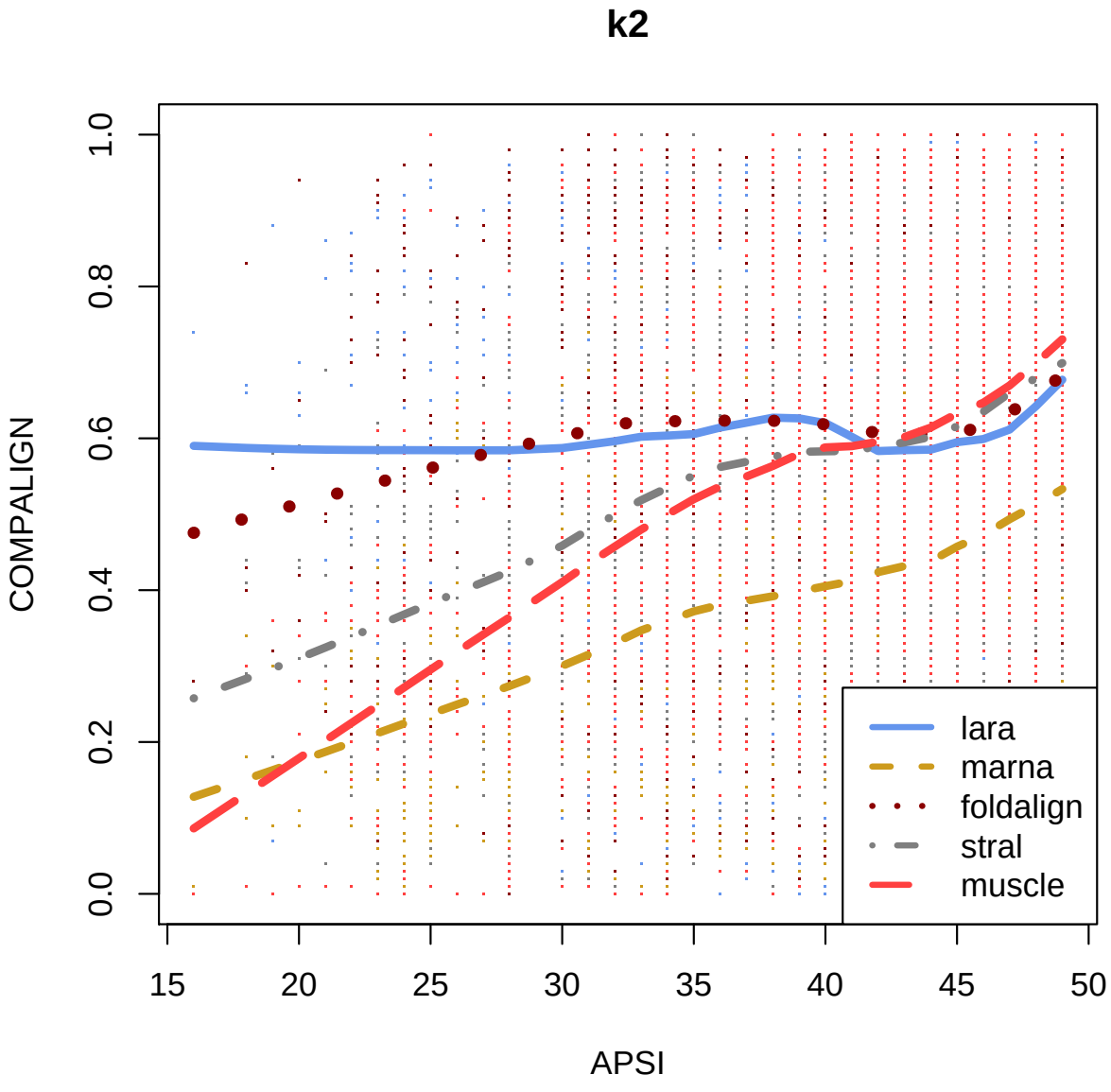
Computational Results: LaRA



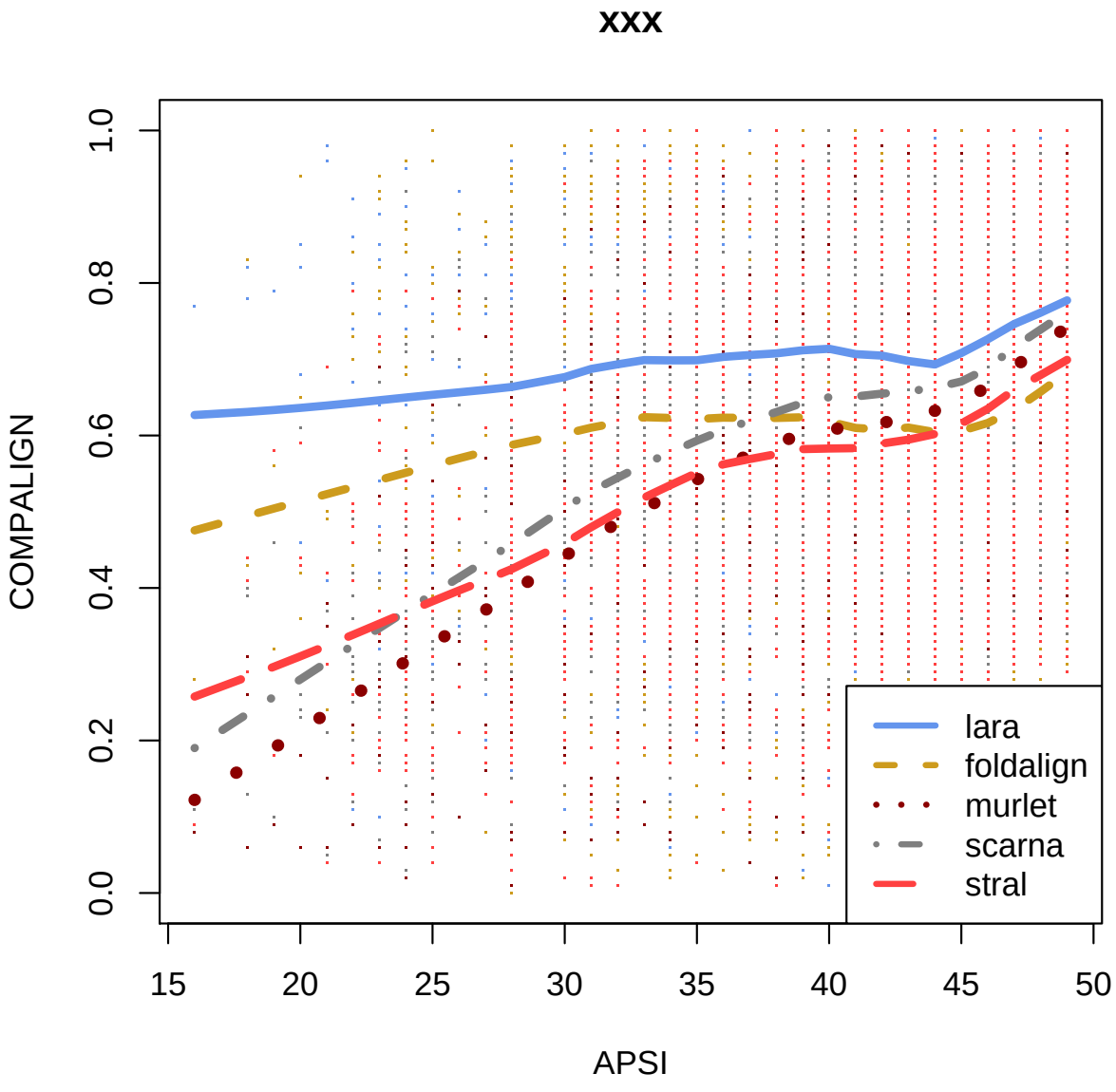
Computational Results: LaRA



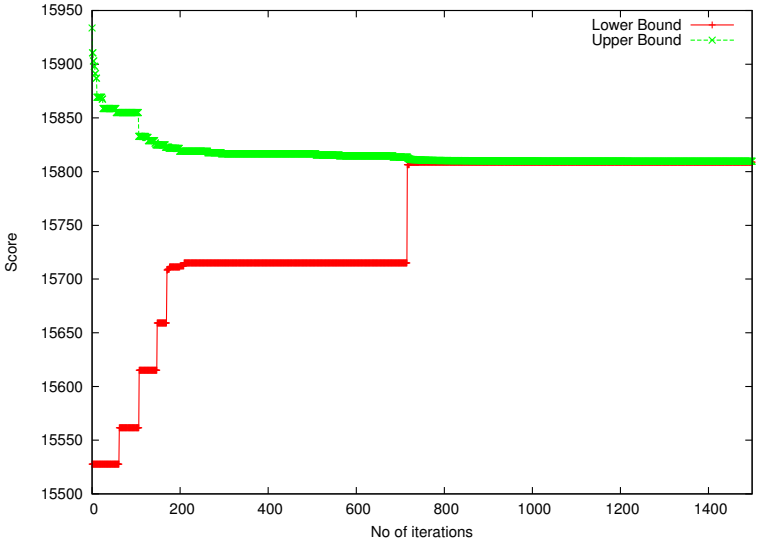
Computational Results: different parameters

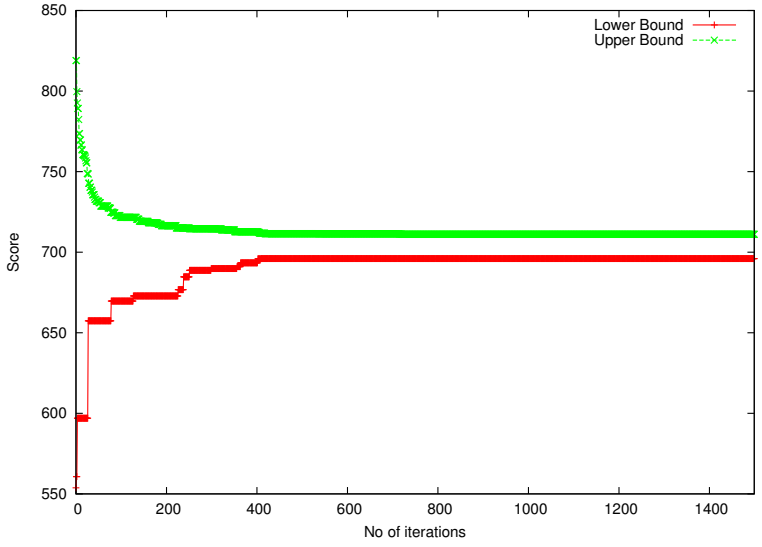


Computational Results: different parameters



Computational Results: LaRA





- LaRA is no exact method: Structures dissimilar $\longrightarrow$ large gap.

HAVE A NICE CHRISTMAS BREAK AND A HAPPY NEW YEAR!