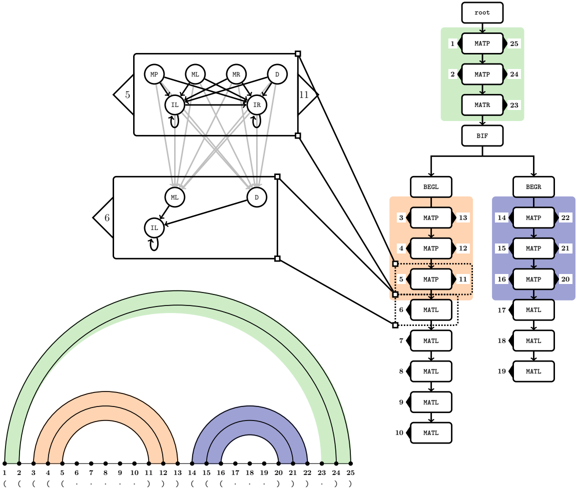
Formulation of Pseudoknotted Covariance Models

Bertrand Marchand
Post-doc at University of Sherbrooke (Québec, Canada)
with Manuel Lafond and Aïda Ouangraoua

Former PhD student of Yann Ponty and Laurent Bulteau
this work: with Amibio team, in $\sim$Paris

August 2, 2024

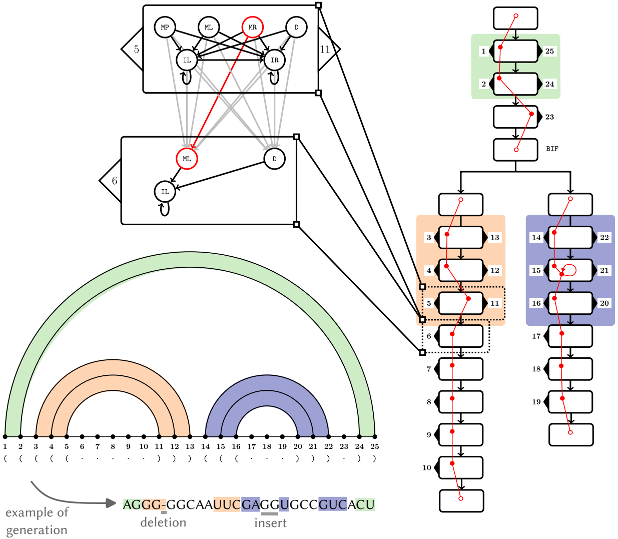
Covariance Models, reminder 1: basics



- Statistical model for **homology families** [Eddy and Durbin, 1994]
- built from **structure-annotated** MSA
- Can use it to **scan databases** for new homologs

  → at the base of RFAM [Kalvari et al., 2021] through `InfeRNAl` [Nawrocki and Eddy, 2013]
- Can produce MSA of input sequences

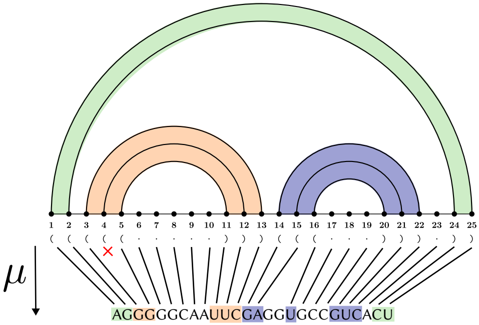
Covariance Models, reminder 2: sampling

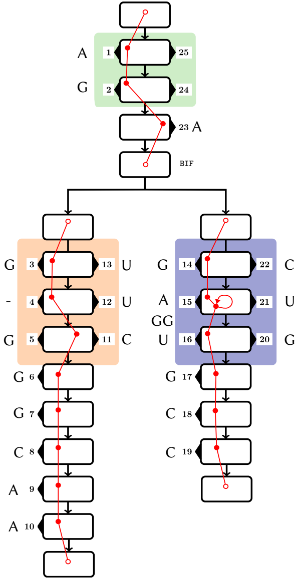


- probability space = **aligned sequences**

$$
\begin{aligned}
\log P(\text{aligned seq})) = \\
&\log P(\text{state sequence}) + \\
&\log P(\text{symbol emissions}) \\
&= \sum_{\text{transition } u \to v} \log P(u \to v) \\
&+ \sum_{\text{emitting state } u} \log P(\text{emission})
\end{aligned}
$$

Covariance Models, reminder 3: alignment of a sequence to the model



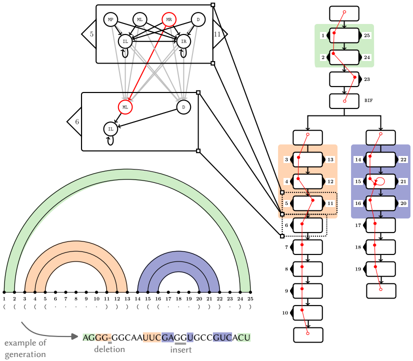


dynamic progaming. Example for MP state:

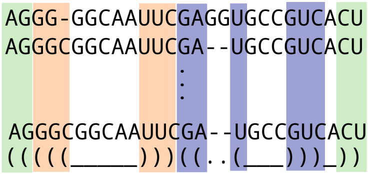
$$M[v,i,j] = \log P(v \text{ emits } S[i],S[j]) + \max_{w\in\texttt{succ}(v)} [M[w,i+1,j-1] + \log p(v \to w)]$$

$M[v,i,j]$: most likely state sequence +
emission scenario starting at v and generating S[i:j]

Covariance Models: highlight on some features



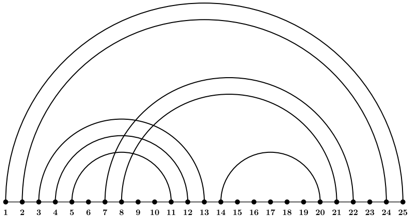
- **stacked** base-pairs are not independent
- no dependence accross helices though
- scores are **position-dependent**, learned from alignment (counting)



```
AGGG-GGCAAUUCGAGGUGCCGUCACU
AGGGCGGCAAUUCGA--UGCCGUCACU
              :
 AGGGCGGCAAUUCGA--UGCCGUCACU
(((((______)))((..(___)))_))
```

structure-annotated MSA

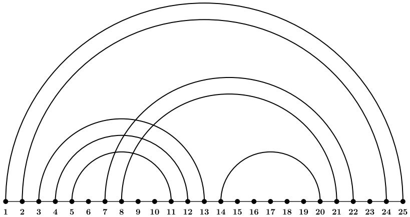
Formulating a Pseudoknotted version: the task



Problems:

- Lose **inside/outside** separation
- No notions of “left/right” anymore
- **alignment**: a close problem is NP-hard [Jiang et al., 2002].

Formulating a Pseudoknotted version: the task



Problems:

- Lose **inside/outside** separation
- No notions of “left/right” anymore
- **alignment**: a close problem is NP-hard [Jiang et al., 2002].

Need equivalents of

- **cmbuild**
- **cmemit**

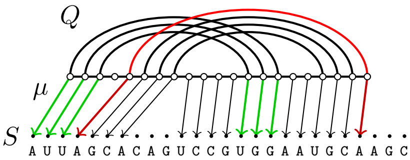
What probabilistic model ?

- **cmalign** → a parameterized algorithm ?

  i.e. runtime of the form $f(k)n^{g(k)}$

A related precedent: LiCoRNA [Rinaudo et al., 2012]

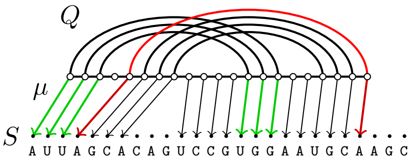
- **input:** structure-annotated seq. Q and seq. S
- **output:** best mapping $\mu \sim$ alignment
- **complexity** $|Q| \cdot |S|^{tw+1}$



$$\texttt{cost}(\mu) = \sum_{i,j \in \texttt{bps}} \texttt{bp\_cost}(Q[i], Q[j], S[\mu(i)], S[\mu(j)])$$

$$+ \sum_{i\,\texttt{unpaired}} \texttt{unpaired\_cost}(Q[i], S[\mu(i)]) + \texttt{affine\_gap\_costs}(\mu)$$

A related precedent: LiCoRNA [Rinaudo et al., 2012]



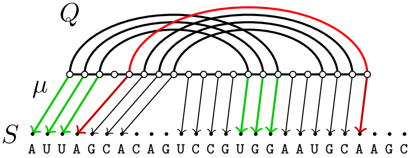
- **input:** structure-annotated seq. Q and seq. S
- **output:** best mapping $\mu \sim$ alignment
- **complexity** $|Q| \cdot |S|^{tw+1}$

$$\texttt{cost}(\mu) = \sum_{i,j \in \texttt{bps}} \texttt{bp\_cost}(Q[i], Q[j], S[\mu(i)], S[\mu(j)])$$
$$+ \sum_{i\,\texttt{unpaired}} \texttt{unpaired\_cost}(Q[i], S[\mu(i)]) + \texttt{affine\_gap\_costs}(\mu)$$

- tw: **treewidth**
- no pseudoknots → tw=2
- RFAM cons. str. (with pk): $tw \leq 5$
  ($tw = 3$: 110, $tw = 4$: 52, $tw = 5$: 3)

A related precedent: LiCoRNA [Rinaudo et al., 2012]

- **input:** structure-annotated seq. Q and seq. S
- **output:** best mapping $\mu \sim$ alignment
- **complexity** $|Q| \cdot |S|^{tw+1}$

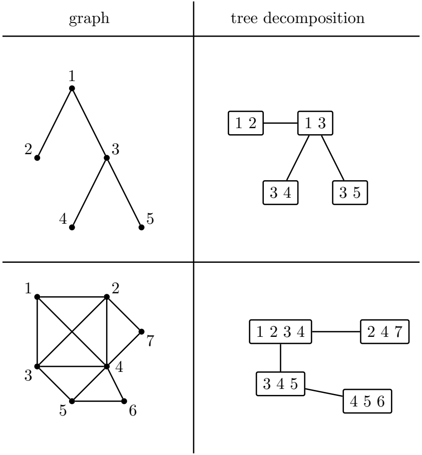


$$\texttt{cost}(\mu) = \sum_{i,j \in \texttt{bps}} \texttt{bp\_cost}(Q[i], Q[j], S[\mu(i)], S[\mu(j)])$$
$$+ \sum_{i\,\texttt{unpaired}} \texttt{unpaired\_cost}(Q[i], S[\mu(i)]) + \texttt{affine\_gap\_costs}(\mu)$$

- tw: **treewidth**
- no pseudoknots → tw=2
- RFAM cons. str. (with pk): $tw \leq 5$ ($tw = 3$: 110, $tw = 4$: 52, $tw = 5$: 3)

	PK	stacking	positional scores
InfeRNAl	✗	✓	✓
LiCoRNA	✓	✗	✗
we want	✓	✓	✓

Treewidth



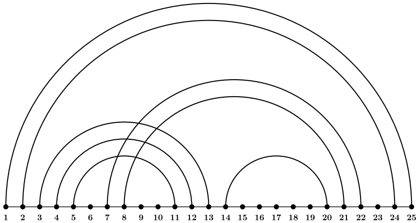
tree decomposition: tree of bags of vertices $\mathcal{T} = (T, \{X_t\}_{t \in T})$ s.t

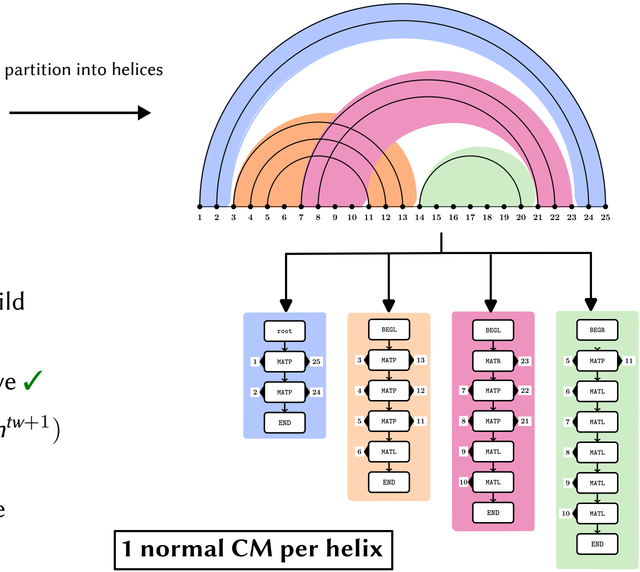
- every vertex is represented in a non-empty connected set of bags
- for each edge (u, v), there is a bag containing u and v

$$tw(G) = \min_{\mathcal{T} \text{ tree dec.}} \max_{t \in T} |X_t| - 1$$

- NP-hard to compute but good heuristic and solvers [Tamaki, 2019]

Proposal: one covariance model per helix



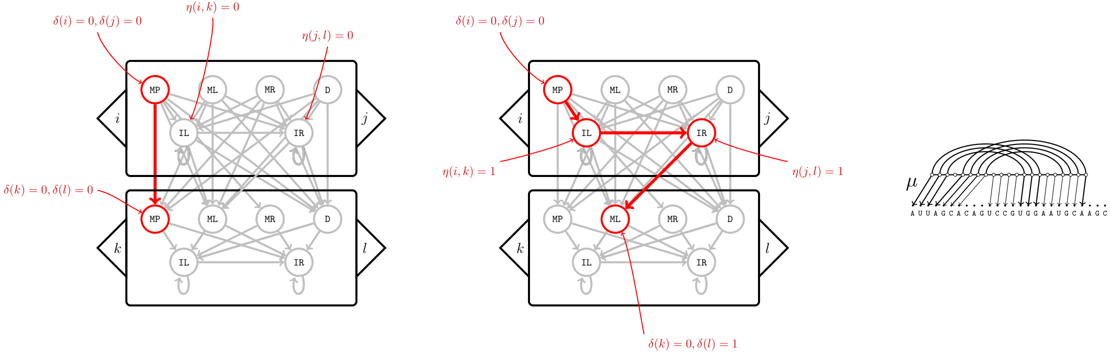


1 normal CM per helix

- building: split seed alignment and build normal CMs ✓
- samping: sample each CM + interleave ✓
- aligning: the hard part, in $O(2^{c \cdot tw} m \cdot n^{tw+1})$ (latest: $O(m \cdot n^{tw+1})$)
- m =consensus size, n = sequence size
- no PK → $tw = 2$ → recover n^3

How do we solve alignment ? $\rightarrow$ state variable encoding

$\delta(i) \in \{0,1\}$: whether consensus position *i* is deleted, $\eta(i) \in \{0,1\}$: whether there is an insert between *i* and $i+1$, $\mu(i)$: where *i* is mapped in the input sequence


$\delta(i)=0, \delta(j)=0$
$\eta(i,k)=0$
$\eta(j,l)=0$
MP
ML
MR
D
IL
IR
i
j
$\delta(k)=0, \delta(l)=0$
MP
ML
MR
D
IL
IR
k
l
$\delta(i)=0, \delta(j)=0$
MP
ML
MR
D
IL
IR
i
j
$\eta(i,k)=1$
$\eta(j,l)=1$
MP
ML
MR
D
IL
IR
k
l
$\delta(k)=0, \delta(l)=1$
μ
A U U A G C A C A G U C C G U G G A A U G C A A G C

- **state sequence + emission scenario** $\Leftrightarrow$ **assigning** δ, η, μ variables.

Cost function network

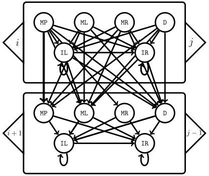
The cost function is :

$$\sum_{\text{helices}} \sum_{\text{node}_1 \to \text{node}_2} f_{\text{node}_1 \to \text{node}_2}(\text{delta\_vars}_1, \text{insert\_vars}_1, \text{delta\_vars}_2)$$

$$+ \text{emission\_term}_1(\text{mu\_vars}_1) + \text{emission\_term}_2(\text{mu\_vars}_2)$$

e.g. for two stacked bps:

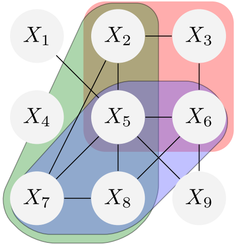
δ_i δ_j δ_{i+1} δ_{j-1} η_i η_{j-1}	score contribution
0 0 0 0 0 0	$\log(P(MP_{ij} \to MP_{i+1,j-1}))$ + emission
1 0 0 0 0 0	$\log(P(MR_{ij} \to MP_{i+1,j-1}))$ + emission
1 0 0 1 0 0	$\log(P(MR_{ij} \to ML_{i+1,j-1}))$ + emission

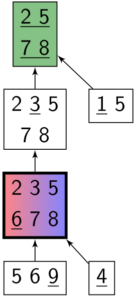


$\vdots$

yields $\to$ $f_{\texttt{MATP}_{ij} \to \texttt{MATP}_{i+1 j-1}}(\delta_i, \delta_j, \eta_i, \eta_{j-1}, \delta_{i+1}, \delta_{j-1})$ term in the costfunction

Treewidth and cost function networks





- X_i: variables with domain D_i
- purpose: minimize some $\sum_{i=k}^{m} f_k(S_k \subset \{X_i\})$
- network: variables scored together → connected
- example on the left: $f_1(X_2, X_5, X_7, X_8) + f_2(X_2, X_3, X_5, X_6) + \dots$

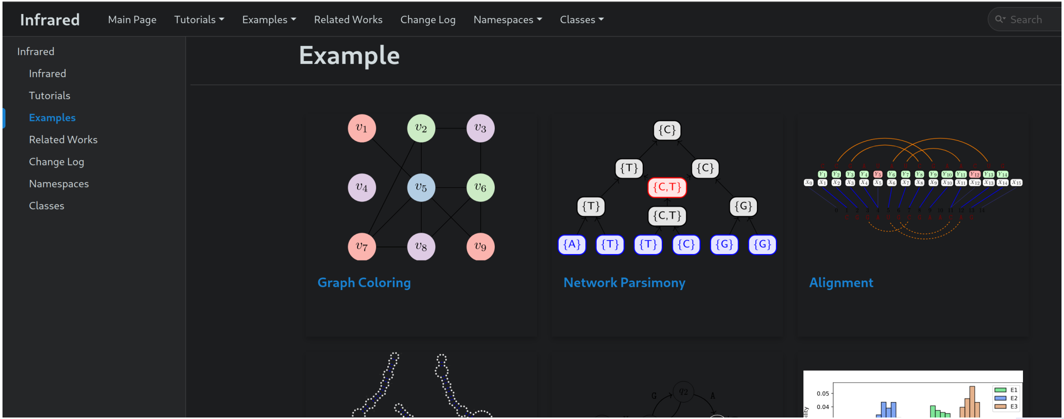
May encode **many** problems, and solvable in $O(m \cdot D^{tw+1})$ with $D = \max_i D_i$

$$T[\texttt{bag}, \texttt{assignment}] = \min_{x \in D_{\text{new}}} \left[\texttt{lcost}(\texttt{full\_assignment}) + \sum_{\texttt{child}} T[\texttt{child}, \texttt{full\_assignment} \cap \texttt{child} \;] \right]$$

`assignment`: on variables both in bag and its parent.

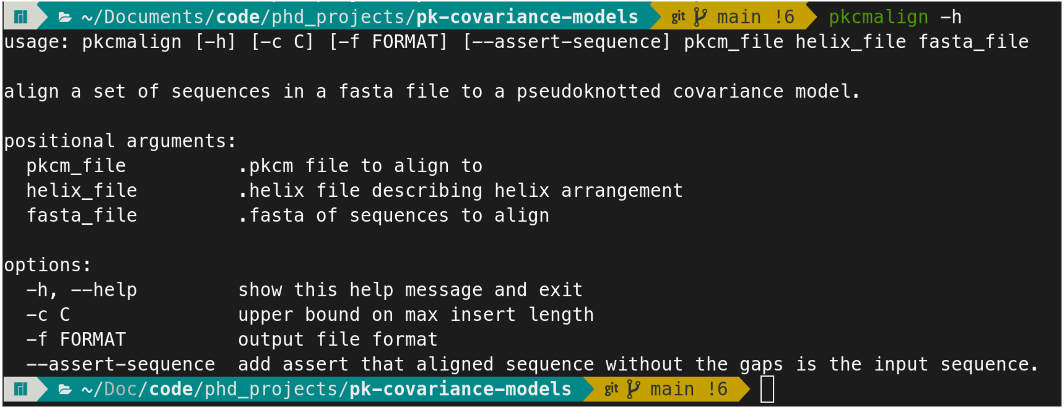
Infrared and prototype implementation

- `Infrared`: generic framework for cost function network optimization [Yao et al., 2024]
- `https://gitlab.inria.fr/amibio/Infrared`



Infrared and prototype implementation

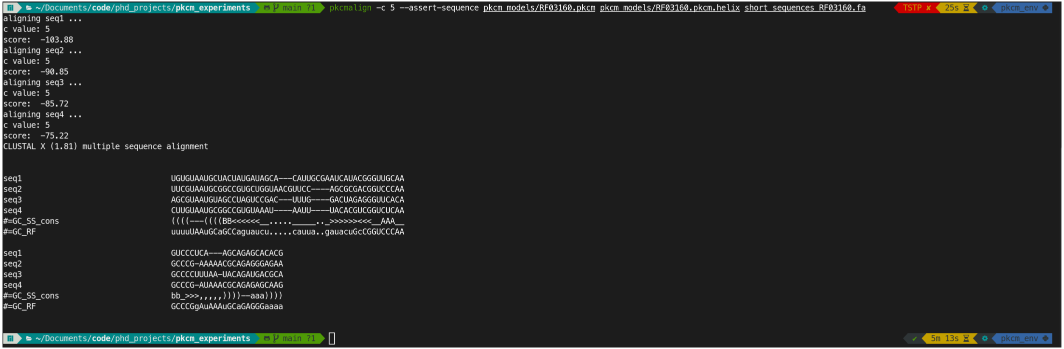
- Infrared: generic framework for cost function network optimization [Yao et al., 2024]
- https://gitlab.inria.fr/amibio/Infrared



```
~/Documents/code/phd_projects/pk-covariance-models  git main !6  pkcmalign -h
usage: pkcmalign [-h] [-c C] [-f FORMAT] [--assert-sequence] pkcm_file helix_file fasta_file

align a set of sequences in a fasta file to a pseudoknotted covariance model.

positional arguments:
  pkcm_file          .pkcm file to align to
  helix_file         .helix file describing helix arrangement
  fasta_file         .fasta of sequences to align

options:
  -h, --help         show this help message and exit
  -c C               upper bound on max insert length
  -f FORMAT          output file format
  --assert-sequence  add assert that aligned sequence without the gaps is the input sequence.
~/Doc/code/phd_projects/pk-covariance-models  git main !6
```

Example

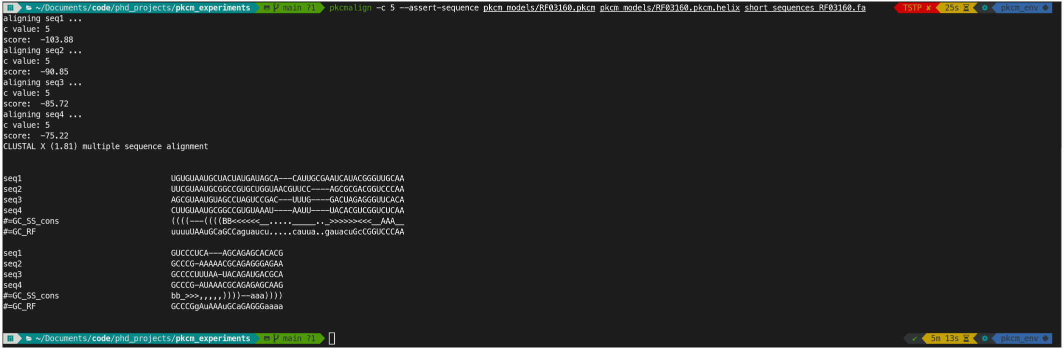
- With **domain banding** $(i - c \leq \mu(i) \leq i + c) \rightarrow O(m \cdot c^{tw+1})$
- Example below: $c = 5$, 4 sequences, $tw = 5$ (RF03160, twister ribozyme), 5 minutes



```
~/Documents/code/phd_projects/pkcm_experiments  main ?1  pkcmalign -c 5 --assert-sequence pkcm_models/RF03160.pkcm pkcm_models/RF03160.pkcm.helix short_sequences_RF03160.fa      TSTP ✘  25s ⌛   pkcm_env
aligning seq1 ...
c value: 5
score:  -103.88
aligning seq2 ...
c value: 5
score:  -90.85
aligning seq3 ...
c value: 5
score:  -85.72
aligning seq4 ...
c value: 5
score:  -75.22
CLUSTAL X (1.81) multiple sequence alignment


seq1                                UGUGUAAUGCUACUAUGAUAGCA---CAUUGCGAAUCAUACGGGUUGCAA
seq2                                UUCGUAAUGCGGCCGUGCUGGUAACGUUCC----AGCGCGACGGUCCCAA
seq3                                AGCGUAAUGUAGCCUAGUCCGAC---UUUG----GACUAGAGGGUUCACA
seq4                                CUUGUAAUGCGGCCGUGUAAAU----AAUU----UACACGUCGGUCUCAA
#=GC_SS_cons                        ((((---((((BB<<<<<<__....._____.._>>>>>><<<__AAA__
#=GC_RF                             uuuuUAAuGCaGCCaguaucu.....cauua..gauacuGcCGGUCCCAA

seq1                                GUCCCUCA---AGCAGAGCACACG
seq2                                GCCCG-AAAAACGCAGAGGGAGAA
seq3                                GCCCCUUUAA-UACAGAUGACGCA
seq4                                GCCCG-AUAAACGCAGAGAGCAAG
#=GC_SS_cons                        bb_>>>,,,,,))))--aaa))))
#=GC_RF                             GCCCGgAuAAAuGCaGAGGGaaaa

~/Documents/code/phd_projects/pkcm_experiments  main ?1                                   ✓  5m 13s ⌛   pkcm_env
```

Example

- With **domain banding** $(i - c \leq \mu(i) \leq i + c) \rightarrow O(m \cdot c^{tw+1})$
- Example below: $c = 5$, 4 sequences, $tw = 5$ (RF03160, twister ribozyme), 5 minutes

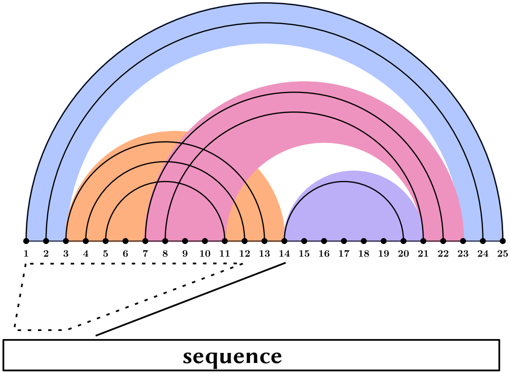


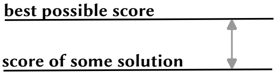
```
 ~/Documents/code/phd_projects/pkcm_experiments   main ?1  pkcmalign -c 5 --assert-sequence pkcm_models/RF03160.pkcm pkcm_models/RF03160.pkcm.helix short_sequences_RF03160.fa      TSTP ✘  25s ⌛    pkcm_env
aligning seq1 ...
c value: 5
score:  -103.88
aligning seq2 ...
c value: 5
score:  -90.85
aligning seq3 ...
c value: 5
score:  -85.72
aligning seq4 ...
c value: 5
score:  -75.22
CLUSTAL X (1.81) multiple sequence alignment


seq1                                UGUGUAAUGCUACUAUGAUAGCA---CAUUGCGAAUCAUACGGGUUGCAA
seq2                                UUCGUAAUGCGGCCGUGCUGGUAACGUUCC----AGCGCGACGGUCCCAA
seq3                                AGCGUAAUGUAGCCUAGUCCGAC---UUUG----GACUAGAGGGUUCACA
seq4                                CUUGUAAUGCGGCCGUGUAAAU----AAUU----UACACGUCGGUCUCAA
#=GC_SS_cons                        ((((---((((BB<<<<<<__.....______.._>>>>>><<<__AAA__
#=GC_RF                             uuuuUAAuGCaGCCaguaucu.....cauua..gauacuGcCGGUCCCAA

seq1                                GUCCCUCA---AGCAGAGCACACG
seq2                                GCCCG-AAAAACGCAGAGGGAGAA
seq3                                GCCCCUUUAA-UACAGAUGACGCA
seq4                                GCCCG-AUAAACGCAGAGAGCAAG
#=GC_SS_cons                        bb_>>>,,,,,))))--aaa))))
#=GC_RF                             GCCCGgAuAAAuGCaGAGGGaaaa


 ~/Documents/code/phd_projects/pkcm_experiments   main ?1  ▯                                      ✓  5m 13s ⌛    pkcm_env
```

- needs **speeding-up**
- Room for improvement in enumeration of variables, in tree decomposition computation. . .

Idea for speeding up 1: analyze weights + some solution

Use *some* solution (e.g. `InfeRNAl`) to give a lower bound to the best score, and use it to rule out possibilities for μ variables



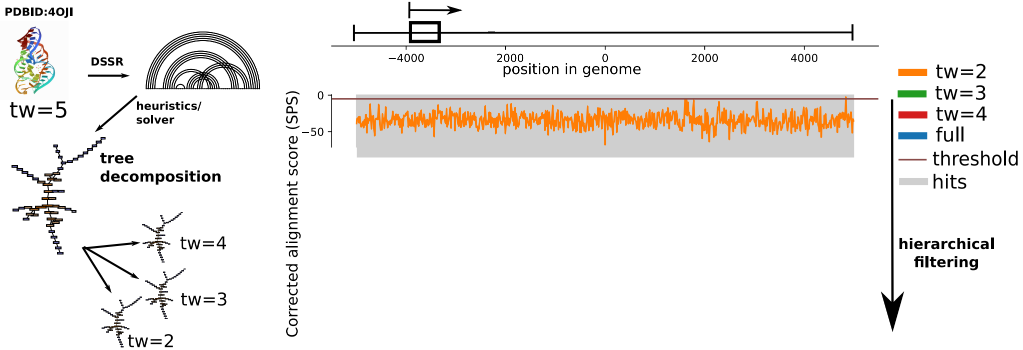


best possible score

score of some solution

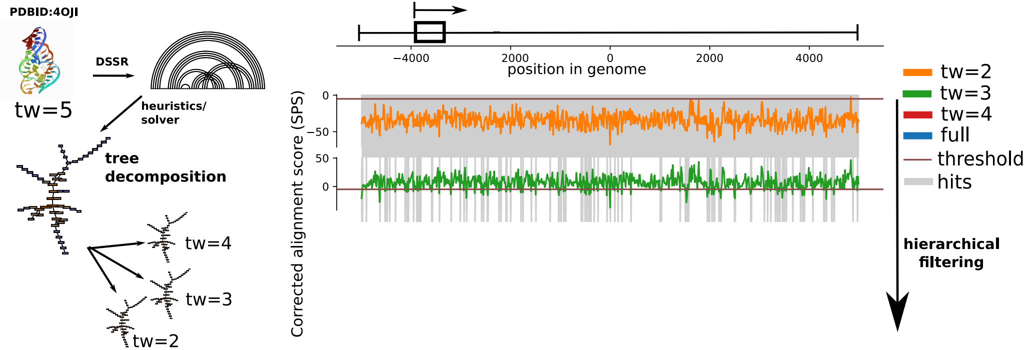
- can compute, for each helix and i, j, the best score of mapping it into $seq[i : j]$. → use it to bound μ domains

Idea for speeding up 2: hierarchy of models[Marchand et al., 2022]



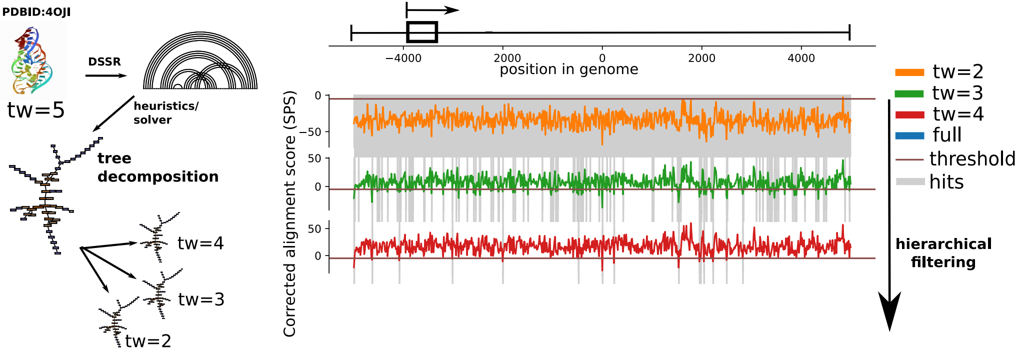
- **Guarantee:** no hits are lost → **exact process**
- Computation of the hierarchy: **a few seconds.**
- Overall speedup: **x42** (24 hours → 34 min with “LiCoRNA” [Rinaudo et al., 2012])

Idea for speeding up 2: hierarchy of models[Marchand et al., 2022]



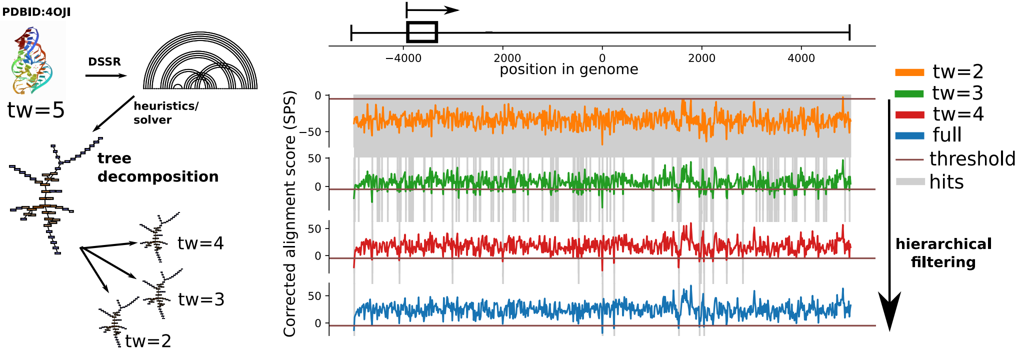
- **Guarantee:** no hits are lost → **exact process**
- Computation of the hierarchy: **a few seconds.**
- Overall speedup: **x42** (24 hours → 34 min with “LiCoRNA” [Rinaudo et al., 2012])

Idea for speeding up 2: hierarchy of models[Marchand et al., 2022]



- **Guarantee:** no hits are lost → **exact process**
- Computation of the hierarchy: **a few seconds.**
- Overall speedup: **x42** (24 hours → 34 min with “LiCoRNA” [Rinaudo et al., 2012])

Idea for speeding up 2: hierarchy of models[Marchand et al., 2022]



- **Guarantee:** no hits are lost → **exact process**
- Computation of the hierarchy: **a few seconds.**
- Overall speedup: **x42** (24 hours → 34 min with "LiCoRNA" [Rinaudo et al., 2012])

Conclusion

- Model and prototype implementation for a fully-featured generalizations of covariance models to the pseudoknotted case → **now, needs speeding up**
- there might also be an interesting middle-ground:

model	PK	stacking	position-dependent scores	multiple interactions
InfeRNAl	✗	✓	✓	✗
LiCoRNA	✓	✗	✗	✓
LiCoRNA+probas	✓	✗	✓	✓
Pseudoknotted CMs	✓	✓	✓	✗

- treewidth and tree decompositions: **automates the design of dynamic programming algorithms for pseudoknotted structures**
- Joint work with past and current members of **Amibio team**: Yann Ponty, Sebastian Will, Hua-Ting Yao, Sarah Berkemer

Conclusion

- Model and prototype implementation for a fully-featured generalizations of covariance models to the pseudoknotted case → **now, needs speeding up**
- there might also be an interesting middle-ground:

model	PK	stacking	position-dependent scores	multiple interactions
InfeRNAl	✗	✓	✓	✗
LiCoRNA	✓	✗	✗	✓
LiCoRNA+probas	✓	✗	✓	✓
Pseudoknotted CMs	✓	✓	✓	✗

- treewidth and tree decompositions: **automates the design of dynamic programming algorithms for pseudoknotted structures**
- Joint work with past and current members of **Amibio team**: Yann Ponty, Sebastian Will, Hua-Ting Yao, Sarah Berkemer

Thank you for your attention

Eddy, S. R. and Durbin, R. (1994).
Rna sequence analysis using covariance models.
Nucleic acids research, 22(11):2079–2088.

Jiang, T., Lin, G., Ma, B., and Zhang, K. (2002).
A general edit distance between RNA structures.
Journal of computational biology, 9(2):371–388.

Kalvari, I., Nawrocki, E. P., Ontiveros-Palacios, N., Argasinska, J., Lamkiewicz, K., Marz, M., Griffiths-Jones, S., Toffano-Nioche, C., Gautheret, D., Weinberg, Z., et al. (2021).
Rfam 14: expanded coverage of metagenomic, viral and microrna families.
Nucleic Acids Research, 49(D1):D192–D200.

Marchand, B., Ponty, Y., and Bulteau, L. (2022).
Tree diet: reducing the treewidth to unlock FPT algorithms in RNA bioinformatics.
Algorithms for Molecular Biology, 17(1):1–17.

Nawrocki, E. P. and Eddy, S. R. (2013).

Infernal 1.1: 100-fold faster RNA homology searches.
Bioinformatics, 29(22):2933–2935.

Rinaudo, P., Ponty, Y., Barth, D., and Denise, A. (2012).
Tree Decomposition and Parameterized Algorithms for RNA Structure-Sequence Alignment Including Tertiary Interactions and Pseudoknots.
In Raphael, B. and Tang, J., editors, *Algorithms in Bioinformatics*, pages 149–164, Ljubljana, Slovenia. Springer.

Tamaki, H. (2019).
Positive-instance driven dynamic programming for treewidth.
Journal of Combinatorial Optimization, 37(4):1283–1311.

Yao, H.-T., Marchand, B., Berkemer, S. J., Ponty, Y., and Will, S. (2024).
Infrared: a declarative tree decomposition-powered framework for bioinformatics.
Algorithms for Molecular Biology, 19(1):13.