Automated design of dynamic programming scheme for RNA folding with pseudoknots

Bertrand Marchand[1,2], Sebastian Will[1], Sarah Berkemer[1], Laurent Bulteau[2], Yann Ponty[1]

[1]LIX, Polytechnique, ∼Paris, France

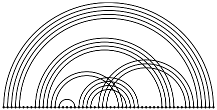
[2]LIGM, Université Gustave Eiffel, ∼Paris, France

BENASQUE 2022

work to be presented at WABI 2022

Problem: MFE folding with pseudoknots, from sequence

...AUGCAAAUGCC... $\xrightarrow[?]{}$



- textbook problem: **folding from sequence**
- without **pseudo-knots**: RnaFold, mfold, RNAstructure...
- with PK and a **general energy model**: **NP-hard** [Sheikh et al., 2012, Lyngsø, 2004]
- But a variety of polynomial DP algorithms developped for specific cases: PKnots, NUPACK, gfold, CCJ, Knotty...

State of the art: DP algorithms for tractable cases

Tool	Reference	space comp.	time comp.	restriction
`Pknots-RE`	[Rivas and Eddy, 1999]	$O(n^4)$	$O(n^6)$	“one-hole structures”
`NUPACK`	[Dirks and Pierce, 2003]	$O(n^4)$	$O(n^5)$	“2 interleaved helices”
`gfold`	[Reidys et al., 2011]	$O(n^4)$	$O(n^6)$	genus ≤ 1
`CCJ`	[Chen et al., 2009]	$O(n^4)$	$O(n^5)$	“3 groups of bands”
`Knotty`	[Jabbari et al., 2018]	$O(n^3 + Z)$	$O(n^5)$	“CCJ-type + optims”
`Pknots-RG`	[Reeder and Giegerich, 2004]	$O(n^2)$	$O(n^4)$	“simple recursive PK”

- all based on **DP tables indexed by positions on the sequence**
- designed either with a **specific target structure family** or a **complexity constraint** in mind

Example of recursive diagram and overall idea

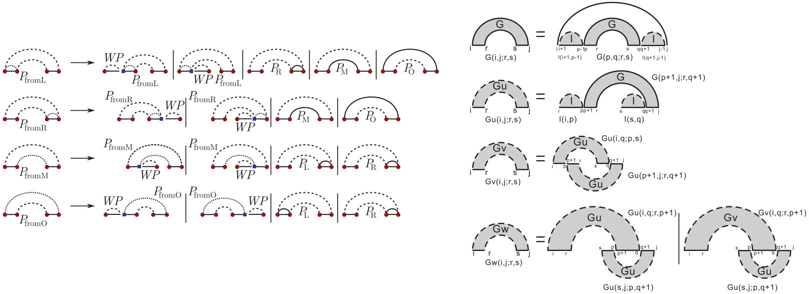
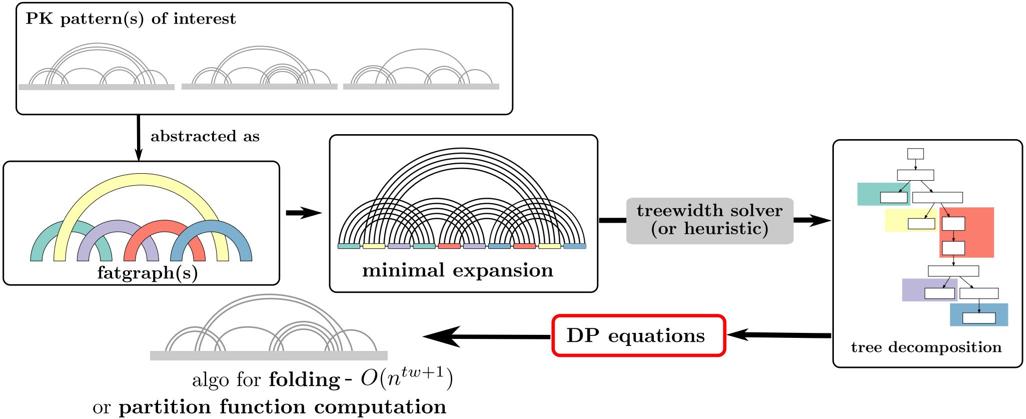


Figure: Examples of DP recursion rules from [Jabbari et al., 2018] and [Reidys et al., 2011]

- Our contribution: a method for, given an **input PK pattern**, **automatically deriving such rules while minimizing the number of used indices**

Overall pipeline



PK pattern(s) of interest

abstracted as

fatgraph(s)

minimal expansion

treewidth solver (or heuristic)

tree decomposition

DP equations

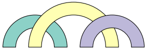
algo for **folding** - $O(n^{tw+1})$

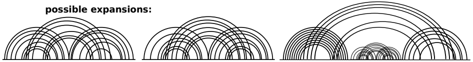
or **partition function computation**

- **fatgraph:** describes a family of structures following a PK pattern
- **1 band = 1 helix with arbitrary length/bulges**

Example: kissing hairpins

- **Input**, this fatgraph:





possible expansions:

- **Output** of our program, these equations:

$$A = \min_{a,d,g} \left(B\,[a,d|d,g] \right)$$

$$B'\,[a,d|d',g] = \min \begin{cases} B'[a,d-1|d',g], & \text{if } d-1, \notin \{a,d',g\} \\ B[a+1,d-1|d',g] + \Delta G(a,d) & \text{if } \{a+1,d-1\} \cap \{d',g\} = \emptyset \end{cases}$$

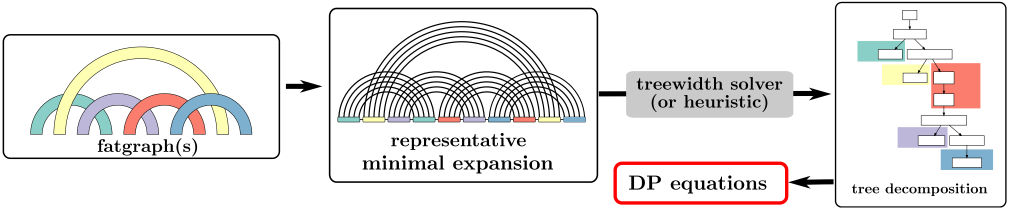
$$B\,[a,d|d',g] = \min \begin{cases} B[a+1,d|d',g], & \text{if } a+1 \notin \{d,d',g\} \\ B'[a,d-1|d',g], & \text{if } d-1, \notin \{a,d',g\} \\ B[a+1,d-1|d',g] + \Delta G(a,d) & \text{if } \{a+1,d-1\} \cap \{d',g\} = \emptyset, \\ C'[d',g|a,d] & \end{cases}$$

$$C'\,[d,g|b,c] = \min \begin{cases} C'[d,g-1|b,c], & \text{if } g-1, \notin \{d,b,c\} \\ C[d+1,g-1|b,c] + \Delta G(d,g) & \text{if } \{d+1,g-1\} \cap \{b,c\} = \emptyset \end{cases}$$

$$C\,[d,g|b,c] = \min \begin{cases} C[d+1,g|b,c], & \text{if } d+1 \notin \{g,b,c\} \\ C'[d,g-1|b,c], & \text{if } g-1, \notin \{d,b,c\} \\ C[d+1,g-1|b,c] + \Delta G(d,g) & \text{if } \{d+1,g-1\} \cap \{b,c\} = \emptyset, \\ C_{\boxtimes}'[b,c-1,d,g+1-1] & \end{cases}$$

- Output equations **solve folding problem** restricted to the **family of structures** specified by the fatgraph
- can support **stacking** and **interior loop/bulge energy terms**
- allow for **recursive substructures**

Inner engine: tree decompositions



fatgraph(s)

representative minimal expansion

treewidth solver (or heuristic)

tree decomposition

DP equations

- **treewidth**: integer quantifying **tree-likeness** of a graph
- **tree decomposition**: gives you the tree structure
- we apply it to a **representative** fatgraph expansion
- essentially gives the **parse tree** of the DP

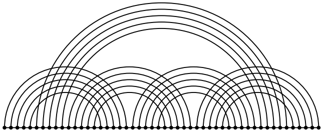
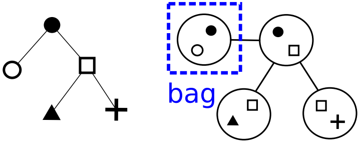


Figure: RNA structure **graph**

Tree decomposition: $\sim$ graph parsing tree

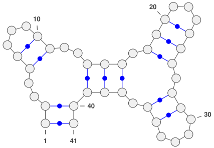
Given a graph, **tree of bags** of vertices following:

- for each **vertex**: **represented** in **connected** set of bags
- for each **edge**, there is a bag containing both ends
- **width**: size of **biggest bag** minus one

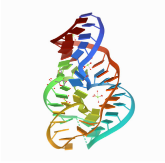


best possible width of a valid tree dec.

tree
tw=1



secondary structure
tw=2



twister ribozyme
tw=5

- hard to compute in general but good solvers/heuristics
- **Small** on RNA structures

Parenthesis: treewidth values of RNAs

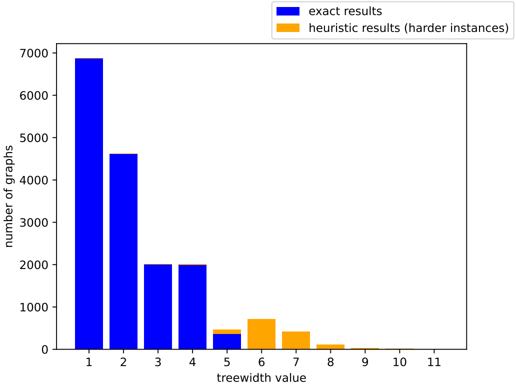


Figure: Canonical interactions only

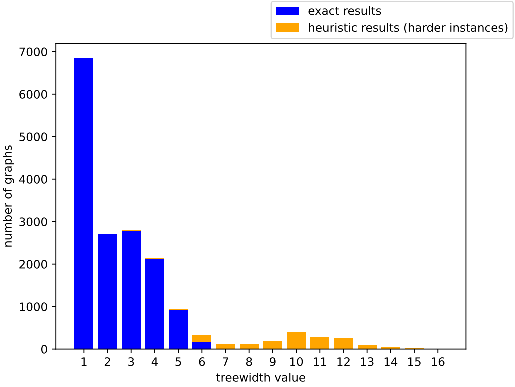


Figure: Inclucing non-canonical interactions

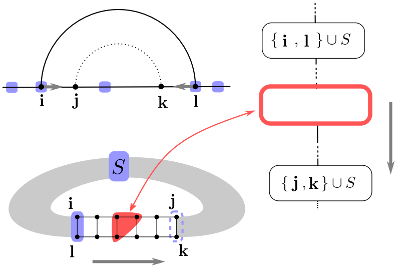
- Histograms of treewidth values over the PDB database (graph extraction with DSSR)

Structural results: recovering typical recursion strategies

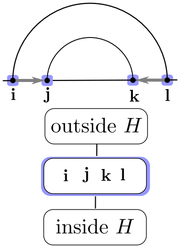
Main theorem

Give an helix H of **length** ≥ 5 in G, **any tree decomposition** of G can be modified to represent H in one of two **canonical ways**

“diagonal” case


i
j
k
l
$\{\mathbf{i}\ ,\mathbf{l}\ \} \cup S$
$\{\mathbf{j}\ ,\mathbf{k}\} \cup S$
S
i
j
l
k

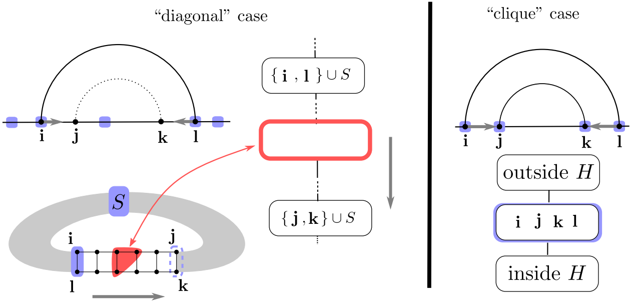
“clique” case



Structural results: recovering typical recursion strategies

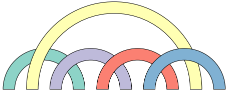
Main theorem

Give an helix H of **length** ≥ 5 in G, **any tree decomposition** of G can be modified to represent H in one of two **canonical ways**

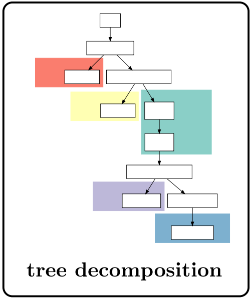


- + (in our paper) an **algorithm** to re-write tree decompositions for canonical representation

DP equations from tree decomposition

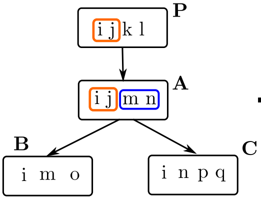


example fatgraph



canonical tree dec.

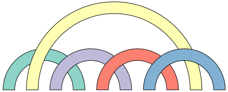
- One DP table per bag/helix
- Indices of the table: intersection with parent bag
- Indices not in parent: marginalization



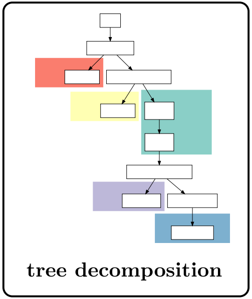
$$\longrightarrow \quad A[\boxed{i,j}] = \min_{\boxed{m,n}} (B[i,m] + C[i,n])$$

for each table → **number of indices $\leq$ treewidth**

DP equations from tree decomposition



example fatgraph



tree decomposition

canonical tree dec.

$$A = \min_{a,g,h,j,k} \left(B\,[a,g,h,j] + C_{\boxtimes}[g,h-1,j,k-1]\right)$$

$$B\,[a,g,h,j] = \min_{e,f,i} \left(C_{\boxtimes}[e,f-1,h,i-1] + C\,[a,e|f,g,i,j]\right)$$

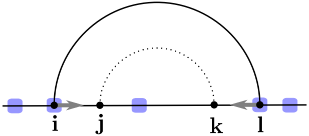
$$C\,[a,e|f,g,i,j] = \min \begin{cases} C[a+1,e|f,g,i,j], \\ C[a,e-1|f,g,i,j], \\ C[a+1,e-1|f,g,i,j] + \Delta G(a,e), \\ D[a,e+1,f,g,i,j] \end{cases}$$

$$D\,[b,d,f,g,i,j] = \min_{c} \left(C_{\boxtimes}[c,d-1,f,g-1] + C_{\boxtimes}[b,c-1,i,j-1]\right)$$

Helix equations (for simplicity: ambiguous, Nussinov)

diag case: only one end given:

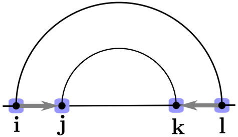
$$D[i,l|S] = \min \begin{cases} D[i+1,l|S] \\ D[i,l-1|S] \\ D[i+1,l-1|S] + score(i,l) \\ \sum_{c\in \text{ children}} M_c[I_c \subset \{i,l\} \cup S] \end{cases}$$



“diagonal” case

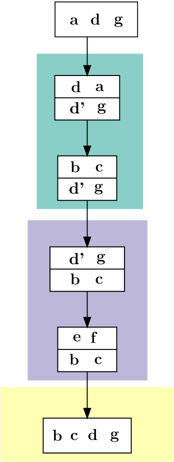
clique case: when all 4 extremities are constrained:

$$C_{\boxtimes}[i,j,k,l] = \min \begin{cases} C_{\boxtimes}[i+1,j,k,l] \\ C_{\boxtimes}[i,j,k,l-1] \\ C_{\boxtimes}[i+1,j,k,l-1] + score(i,j) \\ 0 \text{ if } (i,l) = (j,k) \end{cases}$$



“clique” case

Example: kissing hairpins - treewidth=4



a d g

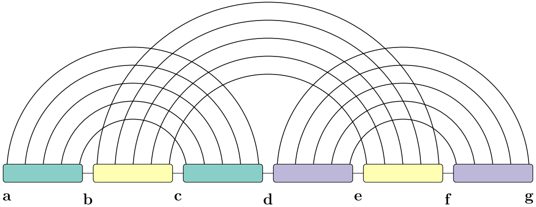
d a
d' g

b c
d' g

d' g
b c

e f
b c

b c d g



$$A = \min_{a,d,g}\left(\boxed{B}\,[a,d|d,g]\right)$$

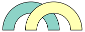
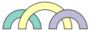
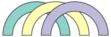
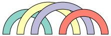
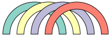
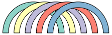
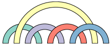
$$\boxed{B}'\,[a,d|d',g] = \min\begin{cases}\boxed{B}'[a,d-1|d',g], & \text{if } d-1, \notin \{a,d',g\}\\ \boxed{B}[a+1,d-1|d',g]+\Delta G(a,d) & \text{if } \{a+1,d-1\}\cap\{d',g\}=\emptyset\end{cases}$$

$$\boxed{B}\,[a,d|d',g] = \min\begin{cases}\boxed{B}[a+1,d|d',g], & \text{if } a+1 \notin \{d,d',g\}\\ \boxed{B}'[a,d-1|d',g], & \text{if } d-1, \notin \{a,d',g\}\\ \boxed{B}[a+1,d-1|d',g]+\Delta G(a,d) & \text{if } \{a+1,d-1\}\cap\{d',g\}=\emptyset,\\ \boxed{C}'[d',g|a,d] & \end{cases}$$

$$\boxed{C}'\,[d,g|b,c] = \min\begin{cases}\boxed{C}'[d,g-1|b,c], & \text{if } g-1, \notin \{d,b,c\}\\ \boxed{C}[d+1,g-1|b,c]+\Delta G(d,g) & \text{if } \{d+1,g-1\}\cap\{b,c\}=\emptyset\end{cases}$$

$$\boxed{C}\,[d,g|b,c] = \min\begin{cases}\boxed{C}[d+1,g|b,c], & \text{if } d+1 \notin \{g,b,c\}\\ \boxed{C}'[d,g-1|b,c], & \text{if } g-1, \notin \{d,b,c\}\\ \boxed{C}[d+1,g-1|b,c]+\Delta G(d,g) & \text{if } \{d+1,g-1\}\cap\{b,c\}=\emptyset,\\ \boxed{C_{\boxtimes}}'[b,c-1,d,g+1-1] & \end{cases}$$

More examples

Name	fatgraph	treewidth	non-Turner, non-recursive	Turner recursive
H-type		4	$O(n^5)$	$O(n^5)$
kissing hairpins		4	$O(n^4)$	$O(n^5)$
“L”		5	$O(n^6)$	$O(n^6)$
“M”		5	$O(n^6)$	$O(n^6)$
4-clique		5	$O(n^6)$	$O(n^6)$
5-clique		5	$O(n^6)$	$O(n^6)$
5-chain		6	$O(n^7)$	$O(n^7)$

- first 4 examples: the 4 “shadows” used in `gfold` [Reidys et al., 2011]
  → we recover the same complexity automatically

Features and limitations

- Can take as input a *finite* number of fatgraphs, with expansions of these fatgraphs recursively inserted.
- Regular secondary structure can also be inserted recursively
- Energy model: depends on what is put in the equations of the two helix cases. → stacking terms and bulges/interior-loop with same complexity cost [Lyngsøet al., 1999].
- Non-ambiguous: partition function computations

Limitations:

- Conformational space of some algorithms ([Rivas and Eddy, 1999], [Dirks and Pierce, 2003]) cannot be described with finite number of fatgraphs

Conclusion and next steps

- Interestingly → we recover typical DP strategies from graph theory analysis
- Algorithm generation: 20 seconds on my laptop to generate all examples shown

Future steps

- Generate **code** directly (and not just latex)
- Complexity is "minimized" but could we prove it is optimal in some sense?

- In general: my PhD → using treewidth to include pseudoknots into algorithms

Conclusion and next steps

- Interestingly → we recover typical DP strategies from graph theory analysis
- Algorithm generation: 20 seconds on my laptop to generate all examples shown

Future steps

- Generate **code** directly (and not just latex)
- Complexity is “minimized” but could we prove it is optimal in some sense?

- In general: my PhD → using treewidth to include pseudoknots into algorithms

Thank you

Chen, H.-L., Condon, A., and Jabbari, H. (2009).
An o (n 5) algorithm for mfe prediction of kissing hairpins and 4-chains in nucleic acids.
Journal of Computational Biology, 16(6):803–815.

Dirks, R. M. and Pierce, N. A. (2003).
A partition function algorithm for nucleic acid secondary structure including pseudoknots.
Journal of computational chemistry, 24(13):1664–1677.

Jabbari, H., Wark, I., Montemagno, C., and Will, S. (2018).
Knotty: efficient and accurate prediction of complex rna pseudoknot structures.
Bioinformatics, 34(22):3849–3856.

Lyngsø, R. B. (2004).
Complexity of pseudoknot prediction in simple models.
In *International Colloquium on Automata, Languages, and Programming*, pages 919–931. Springer.

Lyngsø, R. B., Zuker, M., and Pedersen, C. (1999).
Fast evaluation of internal loops in rna secondary structure prediction.
Bioinformatics (Oxford, England), 15(6):440–445.

Reeder, J. and Giegerich, R. (2004).
Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics.
BMC bioinformatics, 5(1):1–12.

Reidys, C. M., Huang, F. W., Andersen, J. E., Penner, R. C., Stadler, P. F., and Nebel, M. E. (2011).
Topology and prediction of rna pseudoknots.
Bioinformatics, 27(8):1076–1085.

Rivas, E. and Eddy, S. R. (1999).
A dynamic programming algorithm for rna structure prediction including pseudoknots.
Journal of molecular biology, 285(5):2053–2068.

Sheikh, S., Backofen, R., and Ponty, Y. (2012).
Impact of the energy model on the complexity of rna folding with pseudoknots.
In *Annual Symposium on Combinatorial Pattern Matching*, pages 321–333.
Springer.