

# A Simple Protocol for the Inference of RNA Global Pairwise Alignments

---

Felix Karg

22. Januar 2019

University of Freiburg



# Content

Recap

Tree-Based

Sankoff

LocARNA

Conclusion

Sources

**Calm down.**

# Content

Recap

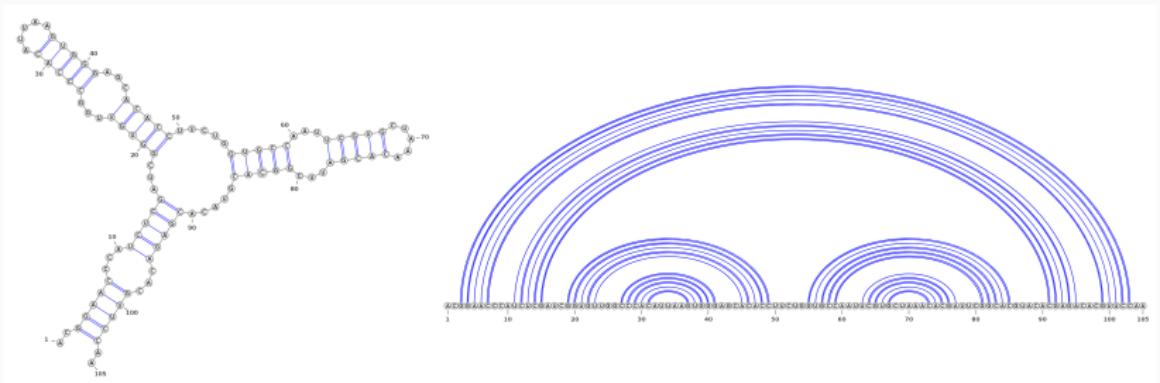
Tree-Based

Sankoff

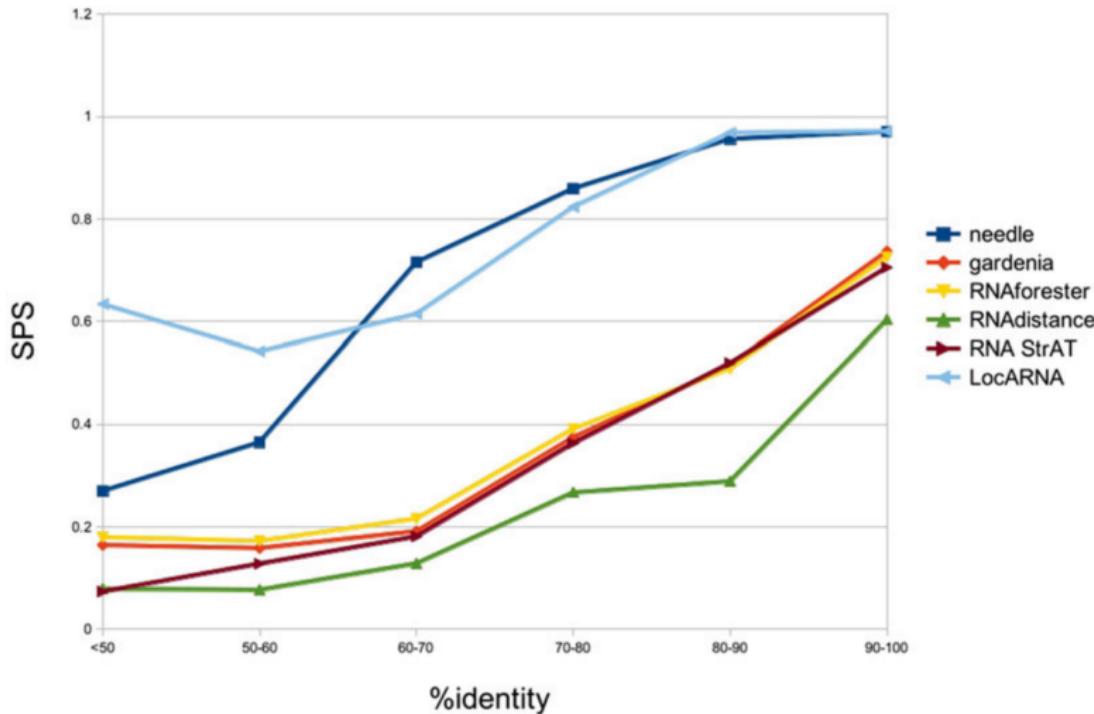
LocARNA

Conclusion

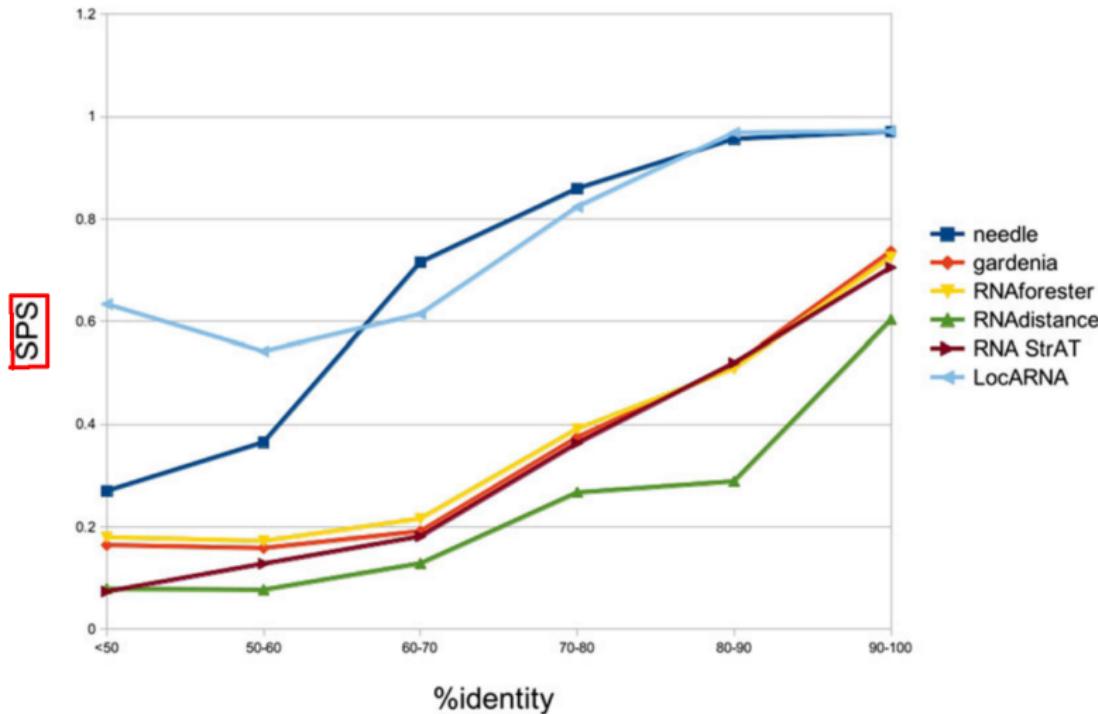
Sources



## Predicted Secondary Structures



## Predicted Secondary Structures



# SPS - introduction

---

Sum of Pairs Score

## Sum of Pairs Score

Used to measure the alignment of two RNA sequences

## Sum of Pairs Score

Used to measure the similarity of two RNA sequences

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:

1 - (edit distance / unaligned length of shorter sequence)

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:

1 - (edit distance / unaligned length of shorter sequence)

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:  $60\% = 1 - (2 / 5)$

$1 - (\text{edit distance} / \text{unaligned length of shorter sequence})$

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:

1 - (edit distance / unaligned length of shorter sequence)

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:  $60\% = 1 - (2 / 5)$

$1 - (\text{edit distance} / \text{unaligned length of shorter sequence})$

## Sequence Similarity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Similarity:

1 - (edit distance / unaligned length of shorter sequence)

## Sequence Similarity - Example

A: AAGGCTT

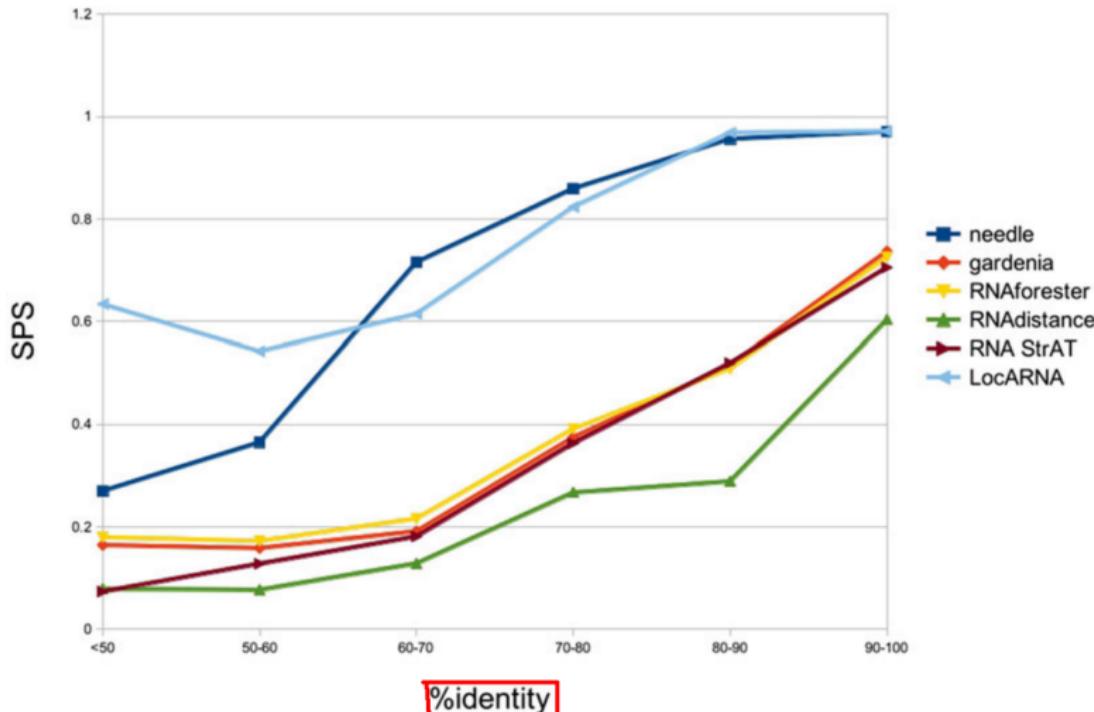
B: AAGGC

C: AAGGCAT

Similarity:  $86\% = 1 - (1 / 7)$

$1 - (\text{edit distance} / \text{unaligned length of shorter sequence})$

## Predicted Secondary Structures



## Sequence Identity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity:

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity:

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity: 100%

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity:

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity: 100%

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

---

A: AAGGCTT

B: AAGGC

C: AAGGCAT

Identity:

Identical nucleotides / shorter sequence length

## Sequence Identity - Example

A: AAGGCTT

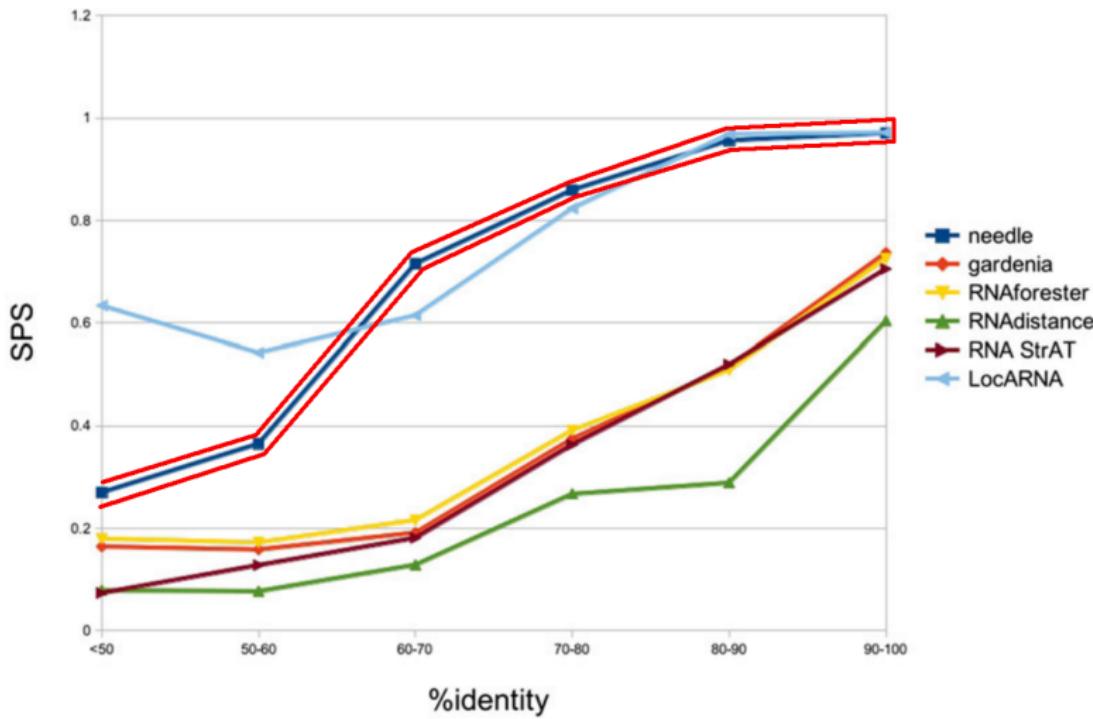
B: AAGGC

C: AAGGCAT

Identity:  $85\% = 6 / 7$

Identical nucleotides / shorter sequence length

## Predicted Secondary Structures



# Needleman-Wunsch-Algorithm

Needleman-Wunsch

match = 1

mismatch = -1

gap = -1

	G	C	A	T	G	C	U	
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1	1	1	0	-1
C	-5	-3	-3	-1	0	0	0	-1
A	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

The diagram illustrates the Needleman-Wunsch algorithm for global sequence alignment. The grid shows scores for matches, mismatches, and gaps. Blue arrows indicate local alignments, while red arrows indicate global alignments extending beyond the local ones.

# Content

---

Recap

Tree-Based

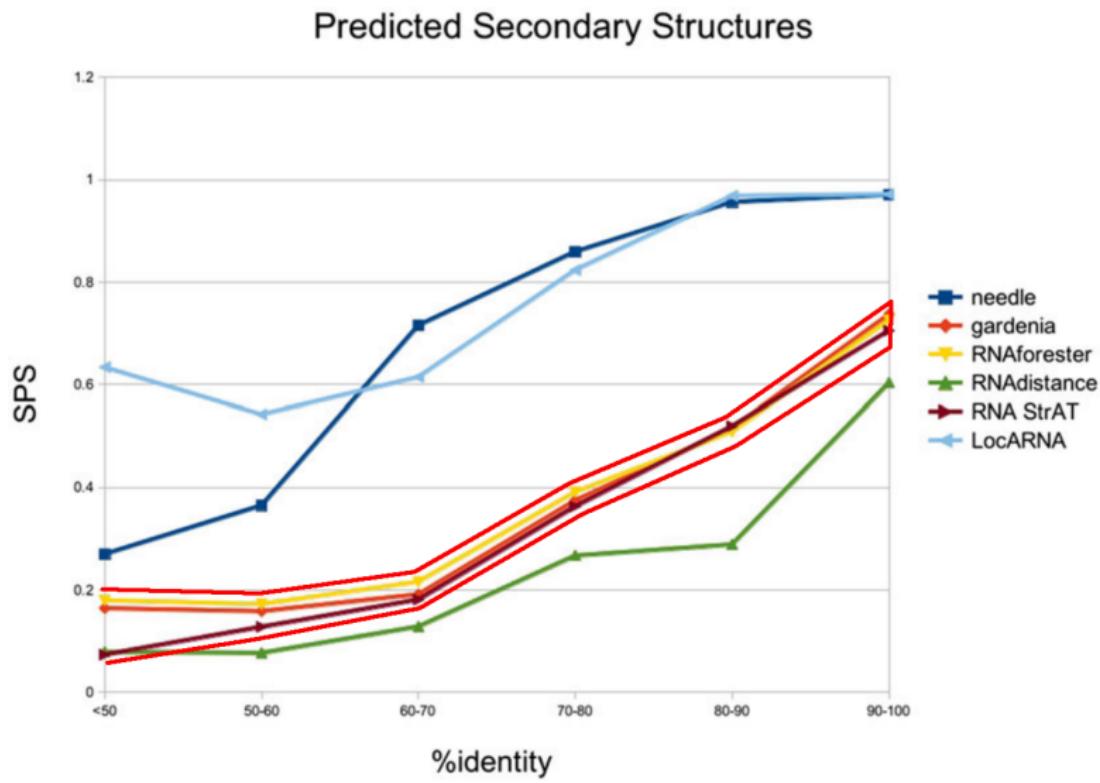
Sankoff

LocARNA

Conclusion

Sources

# Tree-based



## Tree-based

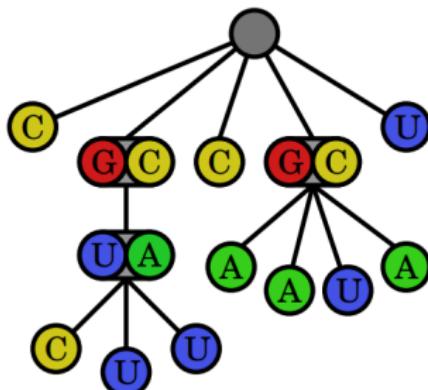
---

Using the secondary structure:

# Tree-based

Using the secondary structure:

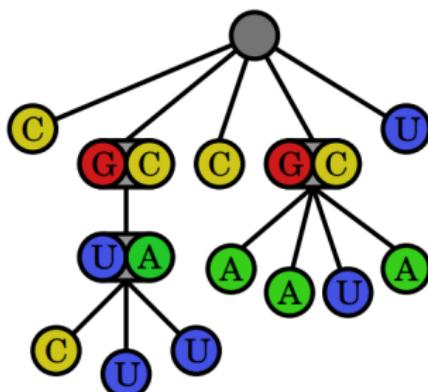
CGUCUUACCGAAUACU  
.((....)).(.....).



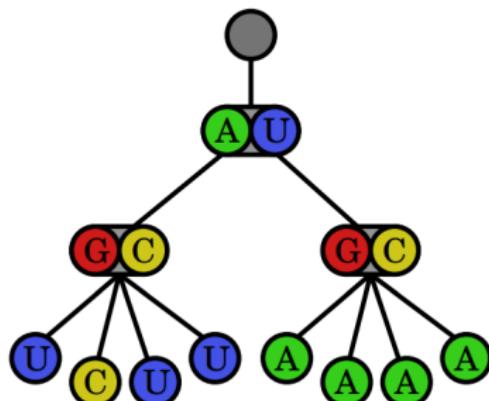
# Tree-based

Using the secondary structure:

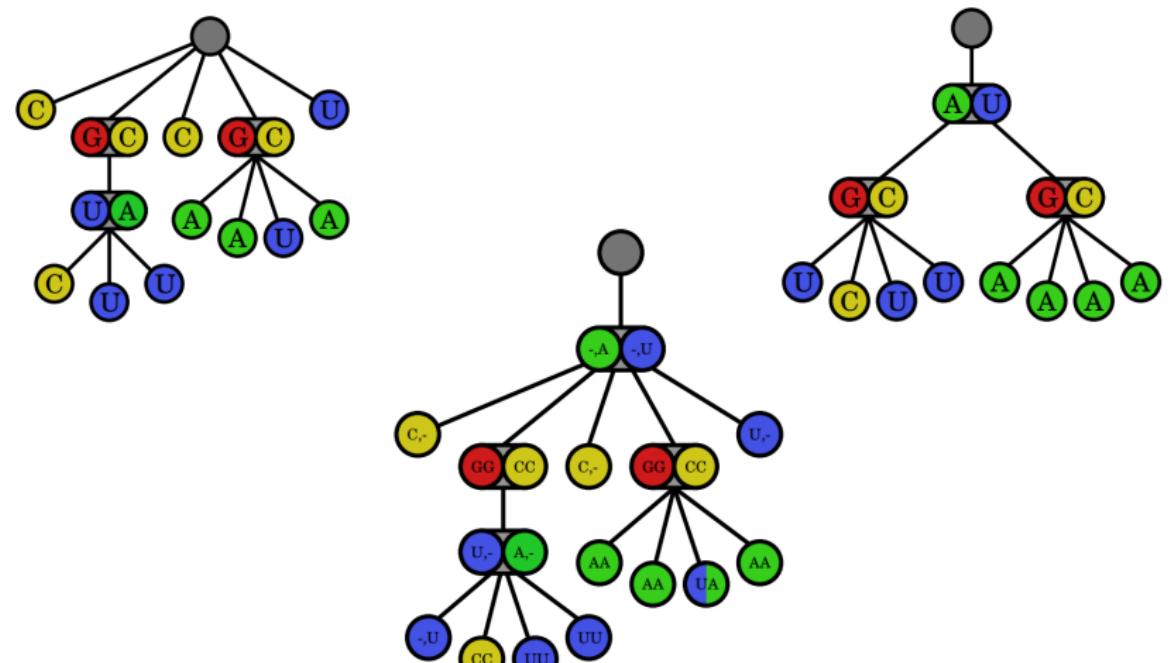
CGUCUUACCGAAUACU  
.((....)).(.....).



AGUCUUCGAAAACU  
((....))(.....)



# Tree-Alignment



# Tree-Alignment Problems

---

# Tree-Alignment Problems

---

- This is not always possible.

## Tree-Alignment Problems

---

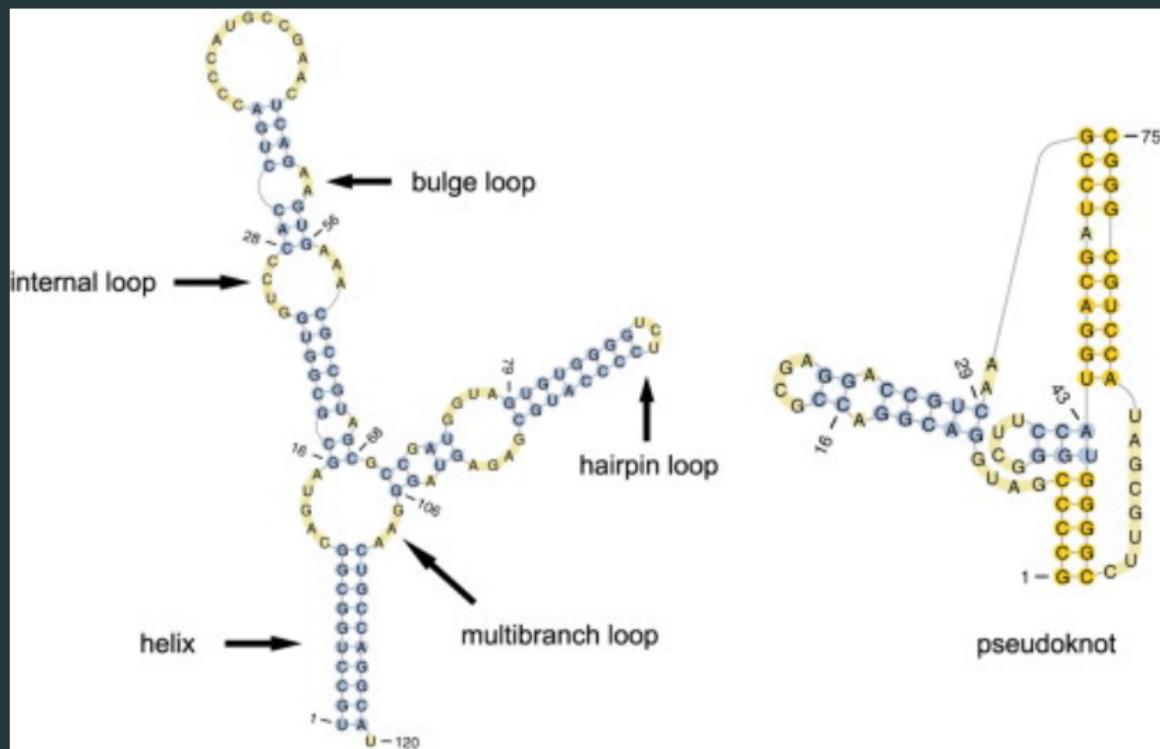
- This is not always possible.
- There's structures that cannot be represented in trees.

# Tree-Alignment Problems

---

- This is not always possible.
- There's structures that cannot be represented in trees.

Example: ( . [ . ] . )



# Tree-Editing

---

Edit operations on Trees ...

# Tree-Editing

---

Edit operations on Trees ...

... are edit operations on arc-annotated  
sequences!

# Tree-Editing

Edit operations on Trees ...

... are edit operations on arc-annotated  
sequences!  $(.((\ldots(\ldots)\ldots)))$

# Tree-Editing Possibilities

1) ACGUUGACUGACAACAC

.. (((.....))) .....

2) ACGAUCACGUACUAGCCUGAC

..... (((.((.....)).)))) .



----.----.(((.....)))....--..

---A---CGUUGACUGACAAC--AC

ACGAUCACGU--ACUAGC--CUGAC

.....(((.((--.....))--.))).

# Tree-Editing Possibilities

- 1) ACGUUGACUGACAACAC  
.. (( (.....) ) .....  
2) ACGAUCACGUACUAGCCUGAC  
.....(( (. ( (.....) ) .) ) ).



----- . (( (.....) ) ..... -- ..  
--- A --- CGUUGACUGACAAC -- AC  
ACGAUCACGU -- ACUAGC -- CUGAC  
.....(( (. ( (.....) ) -- .. ) ) ).

↑  
base match

- base match

# Tree-Editing Possibilities

- 1) ACGUUGACUGACAACAC  
.. (( (.....) ) .....)
- 2) ACGAUCACGUACUAGCCUGAC  
..... (( ( ( (.....) ) .) ) .)



---.----.(( (.....) ))....--...  
---A---CGUUGACUGACAAC--AC  
ACGAUCACGU--ACUAGC--CUGAC  
....(( ( ( ( (.....) ) --. ) ) .))

↑                   ↑  
base deletion      base match

- base match
- base indel

# Tree-Editing Possibilities

1) ACGUUGACUGACAAACAC  
.. (( (.....))) .. ....  
2) ACGAUCACGUACUAGCCUGAC  
.... (( ( ( (.....)) .)))) .



----.----.(( (.....)))....--...  
---A---CGUUGACUGACAAAC--AC  
ACGAUCACGU--ACUAGC--CUGAC  
....(( ( ( ( (.....)) .)) .)) .

↑      ↑      ↑      ↑  
base deletion base match arc match

- base match
- base indel
- arc match

# Tree-Editing Possibilities

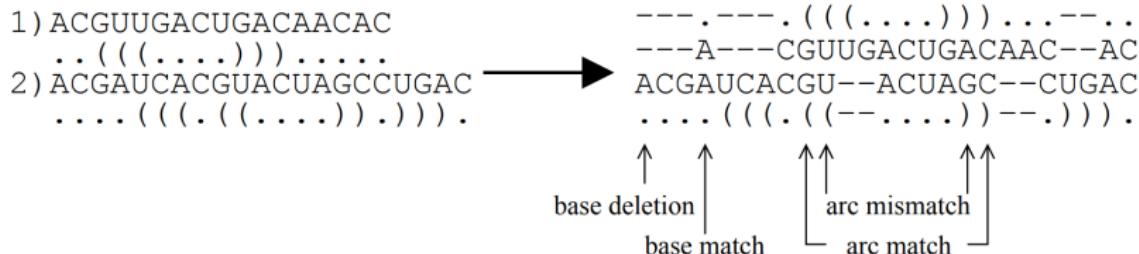
1) ACGUUGACUGACAAACAC  
.. (( (.....) ) .....  
2) ACGAUCACGUACUAGCCUGAC  
..... ((( ( (.....) ) .) ) .)



----.----.((( (.....) )) .....--...  
---A---CGUUGACUGACAAAC--AC  
ACGAUCACGU--ACUAGC--CUGAC  
..... ((( ( ( (.....) ) --. ) ) .)  
  
↑      ↑      ↑      ↑  
base deletion   base match   arc mismatch   arc match

- base match
- base indel
- arc match
- arc mismatch

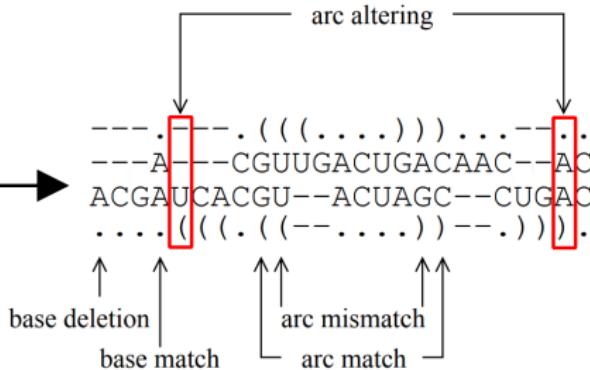
# Tree-Editing Possibilities



- base match
- base indel
- arc match
- arc mismatch
- arc breaking (missing)

# Tree-Editing Possibilities

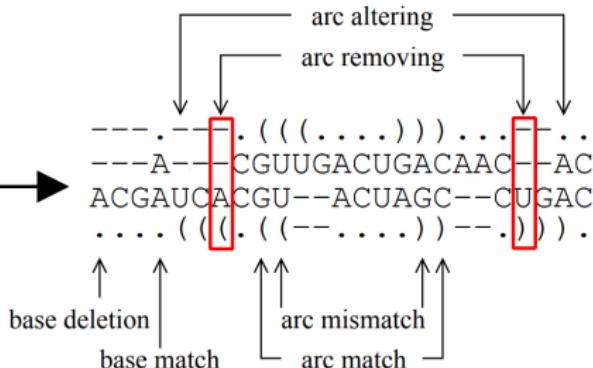
- 1) ACGUUGACUGACAACAC  
.. (( (.....) ) .....)
- 2) ACGAUCACGUACUAGCCUGAC  
..... (( ( ( (.....) ) .) ) .)



- base match
- base indel
- arc match
- arc mismatch
- arc breaking (missing)
- arc altering

# Tree-Editing Possibilities

1) ACGUUGACUGACAAACAC  
.. (( (.....)) .....  
2) ACGAUCACGUACUAGCCUGAC  
..... (( ( ( (.....)) .)))) .

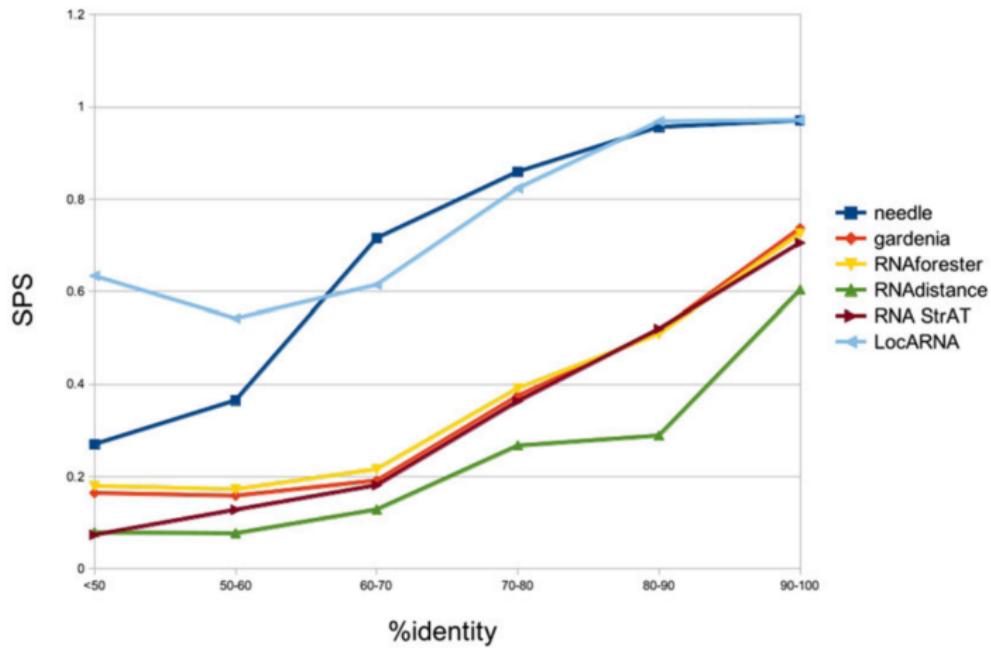


- base match
- base indel
- arc match
- arc mismatch

- arc breaking  
(missing)
- arc altering
- arc removing

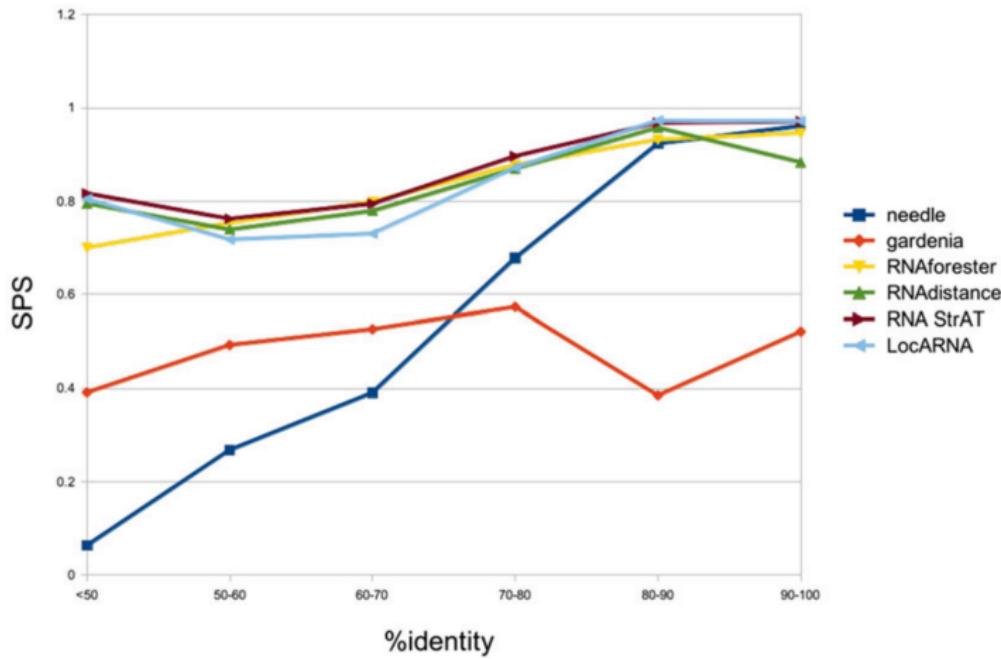
# Performance comparison

Predicted Secondary Structures



# Performance comparison

Curated Secondary Structures



# Tree-based Alignment

---

# Tree-based Alignment

---

Basically a structure-using edit distance.

# Content

---

Recap

Tree-Based

Sankoff

LocARNA

Conclusion

Sources

# Sankoff-Algorithm

---

# Sankoff-Algorithm

---

- Dynamic Programming

# Sankoff-Algorithm

---

- Dynamic Programming
- **Space** needed:  $O(n^4)$

# Sankoff-Algorithm

---

- Dynamic Programming
- **Space** needed:  $O(n^4)$
- **Runtime**  $O(n^6)$

# Sankoff-Algorithm

---

- Dynamic Programming
- **Space** needed:  $O(n^4)$
- **Runtime**  $O(n^6)$

What does it do ... ?

# Sankoff-Algorithm

---

Global free energy minimisation.

# Sankoff-Algorithm

---

Global free energy minimisation.  
Considering every possible combination.

# Content

---

Recap

Tree-Based

Sankoff

LocARNA

Conclusion

Sources

# LocARNA - basic information

---

# LocARNA - basic information

---

- Extensive Caching (Dynamic Programming) + Pruning

# LocARNA - basic information

---

- Extensive Caching (Dynamic Programming) + Pruning
- **Runtime**  $O(n^4)$

# LocARNA - basic information

---

- Extensive Caching (Dynamic Programming) + Pruning
- **Runtime**  $O(n^4)$
- **Space**  $O(n^4)$

# LocARNA - basic information

---

- Extensive Caching (Dynamic Programming) + Pruning
- **Runtime**  $O(n^4)$
- **Space**  $O(n^4)$
- Web Interface

# LocARNA - basic information

---

- Extensive Caching (Dynamic Programming) + Pruning
- **Runtime**  $O(n^4)$
- **Space**  $O(n^4)$
- Web Interface
- Developed at the University of Freiburg

# LocARNA - Procedure

---

# LocARNA - Procedure

---

- Where can arcs happen? And how likely are they?

# LocARNA - Procedure

---

- Where can arcs happen? And how likely are they?
- Building own likely secondary structures

# LocARNA - Procedure

---

- Where can arcs happen? And how likely are they?
- Building own likely secondary structures
- Aligning between secondary structures

# LocARNA - Procedure

---

- Where can arcs happen? And how likely are they?
- Building own likely secondary structures
- Aligning between secondary structures
- Using **one** relative path Matrix

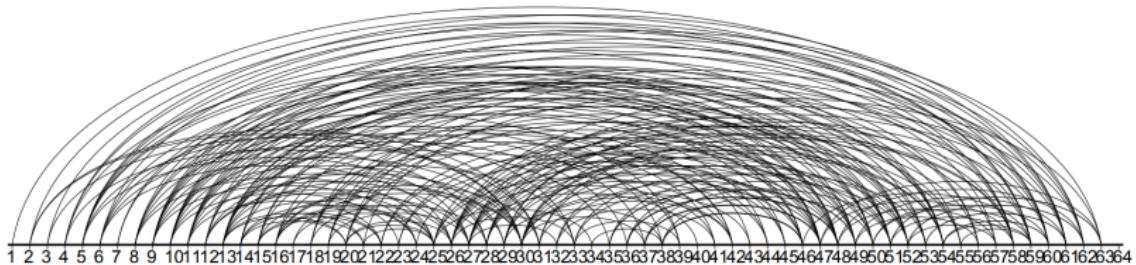
# LocARNA - Procedure

---

- Where can arcs happen? And how likely are they?
- Building own likely secondary structures
- Aligning between secondary structures
- Using **one** relative path Matrix
- Last: 'looking up' optimal alignment

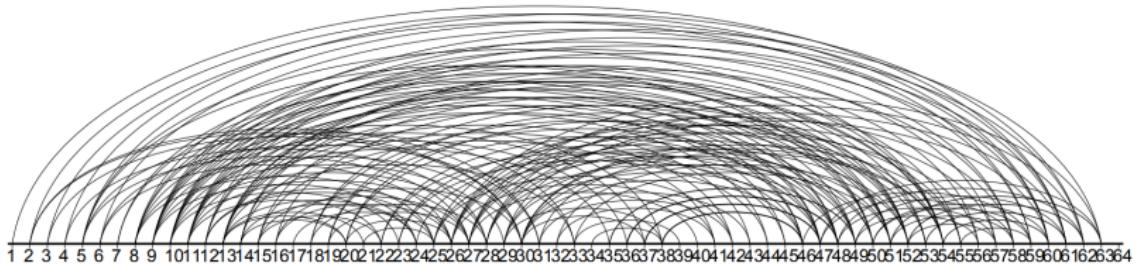
# LocARNA - Where can arcs happen? And how likely are they?

$$p_{\text{cutoff}} = 0.005$$



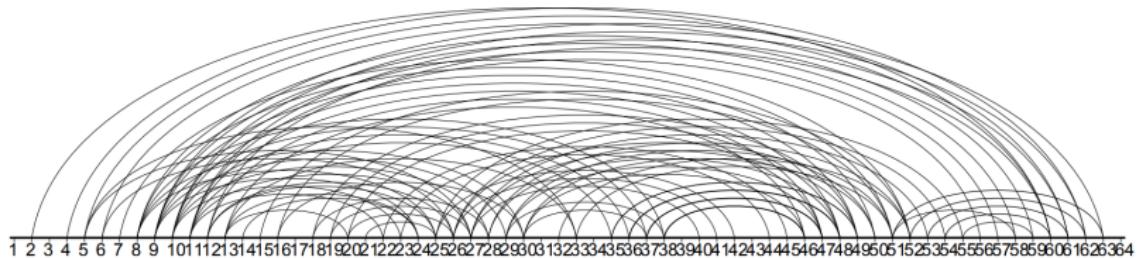
# LocARNA - Where can arcs happen? And how likely are they?

$$p_{\text{cutoff}} = 0.01$$



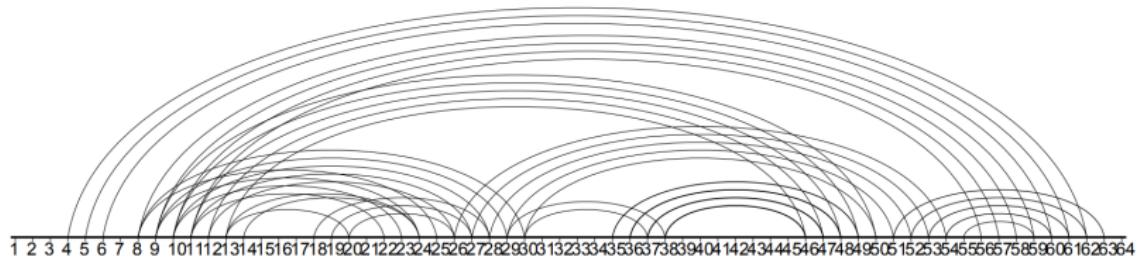
# LocARNA - Where can arcs happen? And how likely are they?

$$p_{\text{cutoff}} = 0.05$$

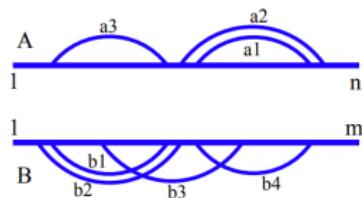


# LocARNA - Where can arcs happen? And how likely are they?

$$p_{\text{cutoff}} = 0.1$$



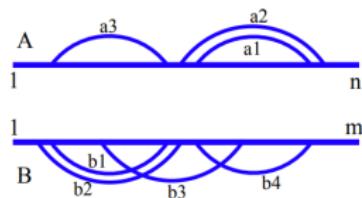
# LocARNA - Matrices



D

	$b_1$	$b_2$	$b_3$	$b_4$
$a_1$				
$a_2$				
$a_3$				

# LocARNA - Matrices



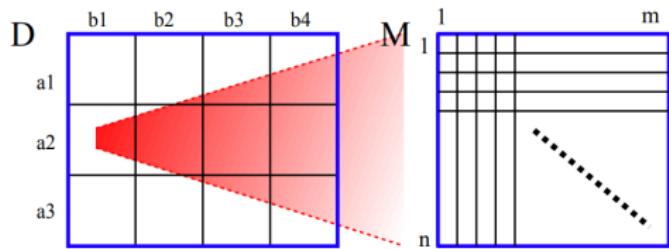
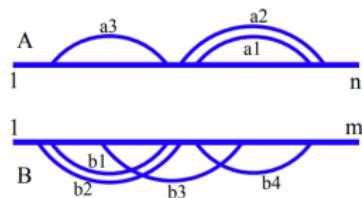
D

	b1	b2	b3	b4
a1	■			
a2				
a3				

M<sub>1</sub>

	1								m
	■								
		■							
			■						
				■					
					■				
						■			
							■		
								■	
									■

# LocARNA - Matrices



# LocARNA - Aligning between secondary structures

---

Cases:

# LocARNA - Aligning between secondary structures

---

Cases:

- Base match

# LocARNA - Aligning between secondary structures

---

Cases:

- Base match
- Base insertion

# LocARNA - Aligning between secondary structures

---

Cases:

- Base match
- Base insertion
- Base deletion

# LocARNA - Aligning between secondary structures

---

Cases:

- Base match
- Base insertion
- Base deletion
- Base **pair** match

# LocARNA - Aligning between secondary structures

Needleman-Wunsch

match = 1

mismatch = -1

gap = -1

	G	C	A	T	G	C	U	
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

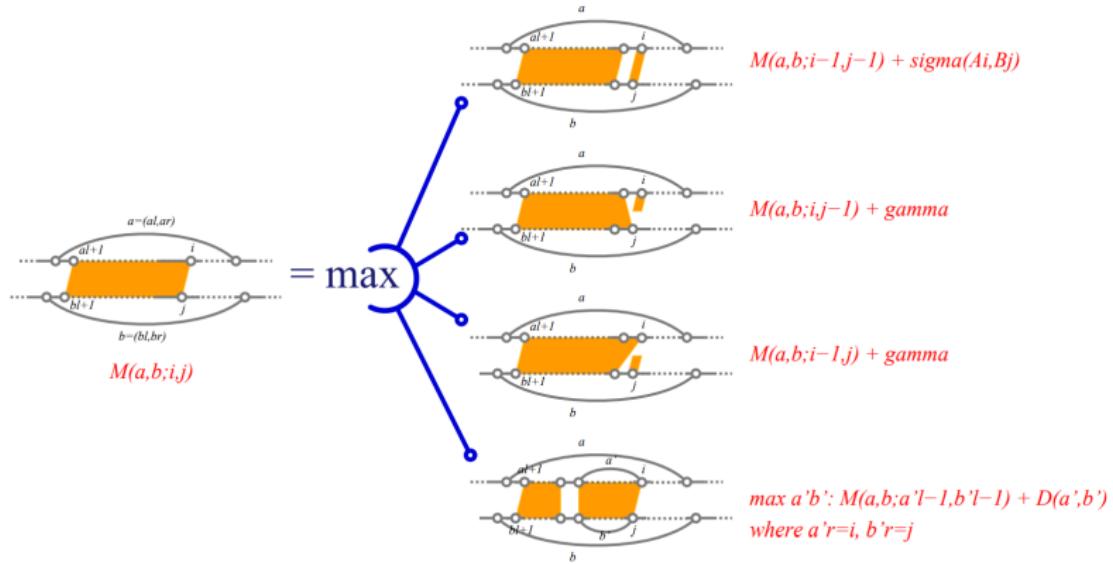
The diagram shows a Needleman-Wunsch alignment matrix for two RNA sequences. The matrix is a grid where rows represent the first sequence and columns represent the second sequence. The scoring parameters are: match = 1, mismatch = -1, and gap = -1. The matrix values are filled with scores from -7 to 0. Blue arrows indicate matches (G-G, C-C, A-A, T-T), blue arrows with a dot indicate mismatches (A-T, T-A, C-G, G-C), and red arrows indicate gaps (A-deletions, T-deletions, G-deletions, C-deletions). The highest-scoring path (blue arrows) starts at the top-left (0,0) and ends at approximately (4, 4), indicating a local alignment between the two sequences.

# LocARNA - Demonstration



# Demo Time!

# LocARNA - Final actual Alignment



# LocARNA - Final actual Alignment

Needleman-Wunsch

match = 1

mismatch = -1

gap = -1

	G	C	A	T	G	C	U	
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1	1	1	0	-1
A	-5	-3	-3	-1	0	0	0	-1
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

# Content

---

Recap

Tree-Based

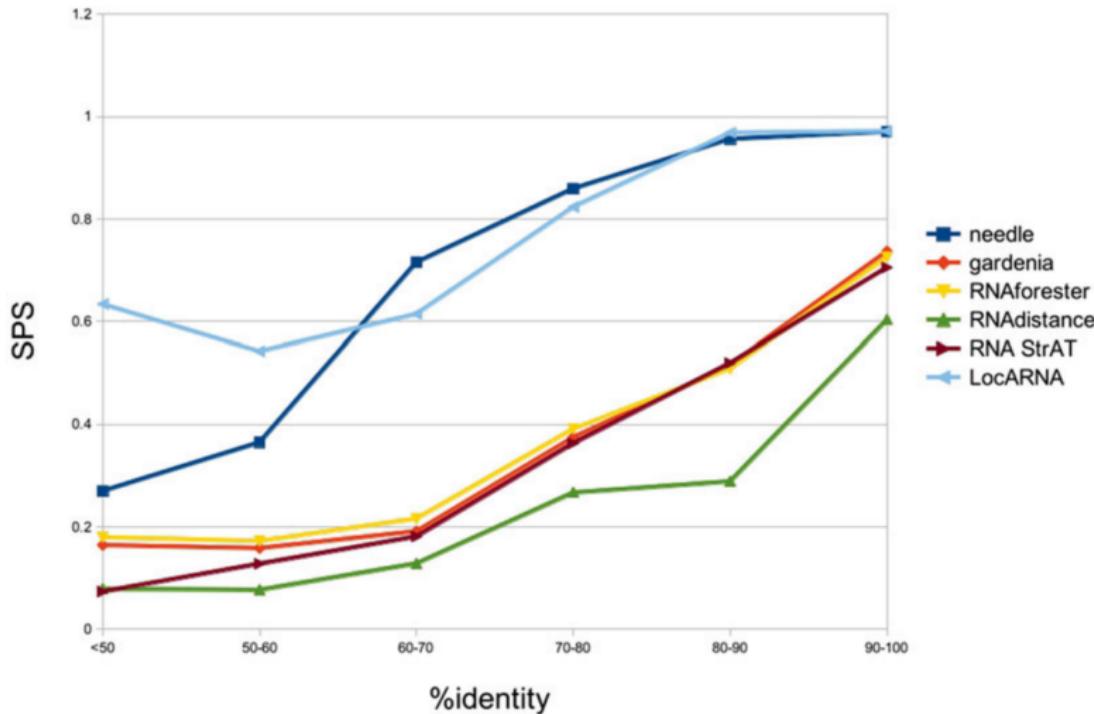
Sankoff

LocARNA

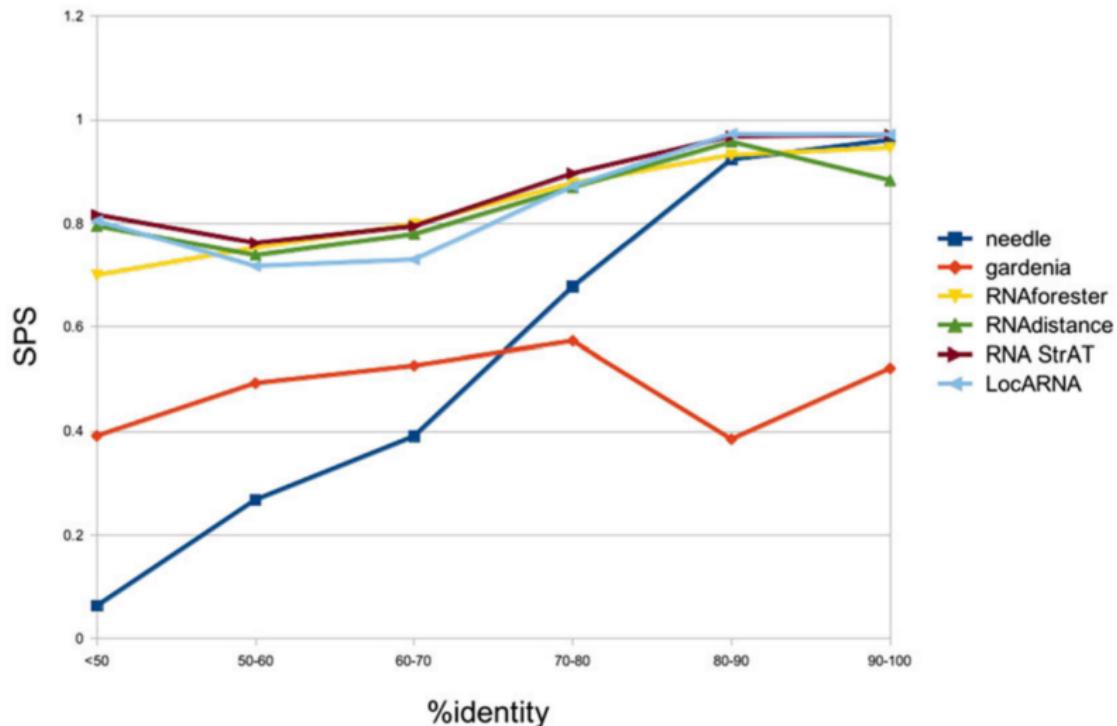
Conclusion

Sources

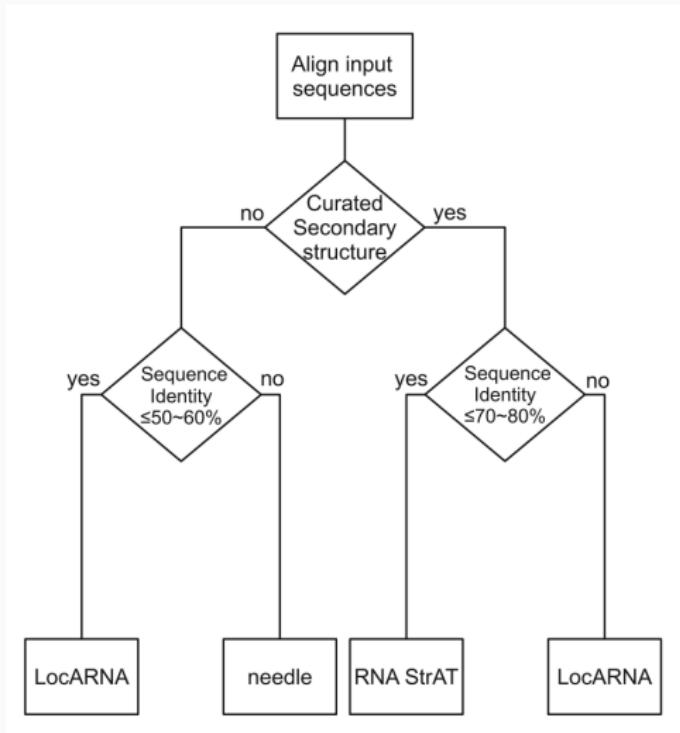
## Predicted Secondary Structures



## Curated Secondary Structures



# Proposed Workflow



# Content

---

Recap

Tree-Based

Sankoff

LocARNA

Conclusion

Sources

# Sources i

---

The slides can be found at:



## Github

[https://github.com/fkarg/things-to-talk-about/  
tree/master/bioinfII](https://github.com/fkarg/things-to-talk-about/tree/master/bioinfII)

## Sources ii

---

### Needleman-Wunsch Image

[https://upload.wikimedia.org/wikipedia/commons/3/3f/Needleman-Wunsch\\_pairwise\\_sequence\\_alignment.png](https://upload.wikimedia.org/wikipedia/commons/3/3f/Needleman-Wunsch_pairwise_sequence_alignment.png)

### Secondary structures Image

<https://www.sciencedirect.com/science/article/pii/B9780124200371000014>

# Sources iii

---

## RNA-Bioinformatics Lecture

[https://ilias.uni-freiburg.de/ilias.php?ref\\_id=1009368&obj\\_id=1&cmd=layout&cmdClass=illmpresentationgui&cmdNode=fu&baseClass=illMPresentationGUI](https://ilias.uni-freiburg.de/ilias.php?ref_id=1009368&obj_id=1&cmd=layout&cmdClass=illmpresentationgui&cmdNode=fu&baseClass=illMPresentationGUI)

## RNA-Comparison Lecture MIT

<https://math.mit.edu/classes/18.417/Slides/RNA-comparisonIV.pdf>