

# PMFastR: A New Approach to Multiple RNA Structure Alignment

Dan DeBlasio

Jocelyn Braund

Shaojie Zhang

University of Central Florida, Orlando, FL 32816 USA

{deblasio,shzang}@eecs.ucf.edu

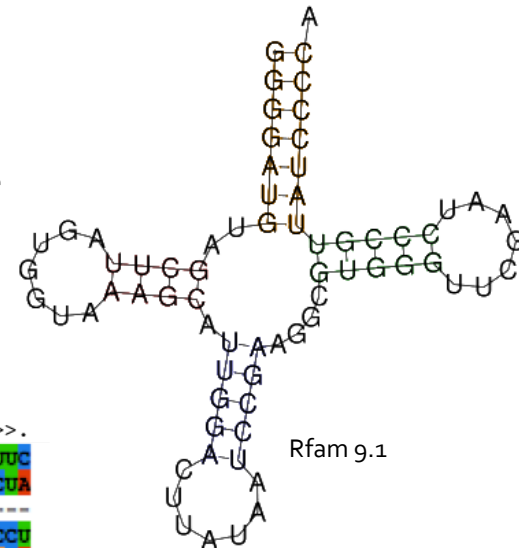
University of California, San Diego, La Jolla, CA 93093 USA

jbraund@ucsd.edu

# Background

# RNA secondary structure is more important to function

- ncRNA secondary structure is very important to function
- Structure is more conserved than sequence
- For example, tRNA (shown) conserves secondary structure between samples more than it conserves sequence
- Alignments of ncRNA should take into account this secondary structure because it is so important



M10217/7325-7260	AAGCCUGCG-GUGUUT	-----GACAUGCCAGAUUGCAAAUCUCGAGA-AGCA-A	---ACGAAGGT-UUGCCGGGCUUC
M10217/5910-5978	AGUAAAAGUC-AGCUAA	---AA---AAGCUUUUGGGCCCAUACCCCAACA-UGUU	GGUUAAC-CCCUUCCUUUACUA
M10217/13781-13846	GCUUUUAAA-GGAAAAC	-AGUC-UAUCGCGUGGUCUUAGGAACAGAAAACUUU	GGUGCAAA-UCCAAGU-----
M10217/5770-5840	GGAAAUGUG-CCCGAA	-AGU---CAGGGAUCACUUUGAUAGAGUGAAAUUAUG	GGUUCAAA-CCCCAUCUCCU
M13046/2222-2289	GCUUACGUA-GCUUAA	---GU---AAAGCACAGCACUGAAGAUGCUGAGA-UGAG	CCCUAGAA--AGCUCCGAAAGCA
Y00163/254-326	GCCCCAUA-GCUCAGUC	GGU---AGAGCAUCAGACUUUUAUCUGAGGGUCCAG	GGUUC AAG-UCCUGUUCGGGGG
M10217/7015-7083	AGAGAUUUA-AGUUAACA	-----AGACUAAGAGCCUUCAAAGCCCUAAG-CAGG	AGUUAGAA-UCUCCUAAUCUCUG
M10217/11905-11973	GAGUUGUUA-GUCUAAAC	-----AAGACAGUUGAUUUUCGGCUCAACAAA-UUAU	GGUUAAC-CCCAUAAUAAUCUCU
M10217/9797-9871	CACUAAAGAA-GCUAAAUA	GGGCAUUAGCGAGAGCCUUUUAAGCUGUAGAUUGGU	GACUCCCA-ACCACCCUUAUGA
X04821/1-73	GUUUCUGUA-GUGUAGC	---GGUU-AUCACGUUCGCCUCACAUCCGAAAGGUCCCC	GGUUCGAA-ACCGGGCAGAAACA
M10217/5909-5841	-AGGAAGUG-GUAUAGU	---GGG-AGUACGGAGGGUUUUGAUCUCUCAGG-UGCA	GGUUC AAU-UCCUGUCUUUCUA
Y00430/2429-2358	GGGGGUGUA-GCUCAGU	---GGU---AGAGCGCAUGCUUUGCAUGUAUGAGGUUUG	GGUUC AAU-CCCCAGCAUCUCCA
M10217/13648-13715	GUAGAUUUA-GUUUAAU	-----AAAACACUAGAUUGUAUUCUAGAGU-CAGA	GGUUAAC-CCCUUUUAUCAACC
M10217/17388-17457	GUCCUGAUA-GCUUAAU	---UU---AAAGCAUCGGUCUUGUAAGCCGAAGA-UGA	GGCUAAAA-CCCUCCUCAAGACU
M10217/11492-11561	ACUUUCUUA-GUAUUA	---ACC-AGUACACGUGACUUCCAAUCCAAAG-UCUU	AGUUAGAA-UCUAAGAGAAAGUA
M10217/7154-7086	AAGGCUUUAAGUUAAUU	-----AAAGUGUUUGAGUUUCAUUCAAUUGA-UGUU	GGAUAAAA-UCCUGCAAGCCUUA
K02456/141-212	AGCAGAGUG-GCGCAGC	---GGA---AGCGUGCUGGGCCCAUAAACCCAGAGGUCGAU	GGAUCGAA-ACCAUUCUCUGCUA
Y10943/948-1016	CAAAGUUAU-GCUUAAAC	-----AAAGCCUUUCGCUUACACCGAAACAA-UAUC	UGUUAAAC-CCGGAUUACUUUGA
M10217/9038-9106	GAGAUGUUA-GUAAAA	---CA---AUUAGCACGCCUUGUCAAGGCGAAGU-AGCU	GGUUAGAC-UCCGGGCACAUCUCA
M10217/2136-2204	GCUUACGUA-GCUUAA	---GU---AAAGCACAGCACUGAAGAUGCUGAGA-UGAG	CCCCUACGAA-A-GCUCCGUAAGCA
M10217/4724-4798	GCUAGCGUG-GCAGAGCCUGGCU	AAUGCAGAAAGACCUAAGCUCUUUUUAUCAGG-GGUUCAAA	-UCCCCUCGUAAACU

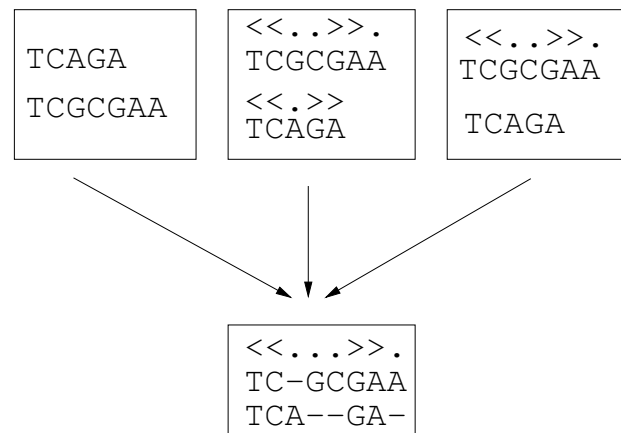
Rfam/ClustalX

# Current methods for multiple alignment

- ClustalW
  - Thomson, *et al.* (Nucleic Acids Res. 1994)
  - Uses profile construction and pairwise alignments to create a sequence based multiple alignment
  - Sequence only multiple alignment
- RNACAD
  - Brown. (ISMB 2000)
  - Uses a CM to build a multiple alignment from a seed alignment
  - Used for RDP
  - SCFG based, requires good seed alignment
- LARA
  - Bauer, *et al.* (BMC Bioinformatics 2007)
  - Using a graph and an ILP solver, they take structural probability into consideration
  - Does not output structure

# Different approaches to RNA alignment

- sequence with structure alignment is an extension of RNA structure prediction
- three major types of alignments
  - *sequence-sequence* -- two sequences with out structure, predict structure after [Sankoff , (*SIAM J. Appl. Math*, 1985)]
  - *structure-structure* – both sequences have structure
  - *sequence-structure* – only one sequence has the associated structure given



# Reasoning for using the *sequence-structure* paradigm

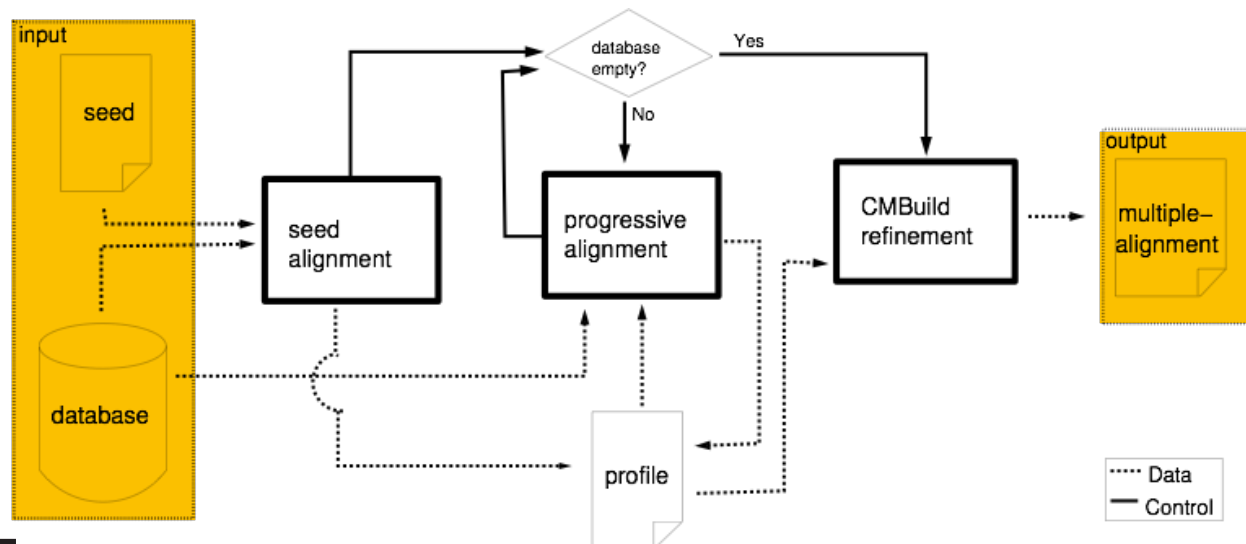
- Experimentally finding the sequence structure for RNA is expensive
- Assume only one sequence in a family has known structure
- Align all others to infer the structure



# Methods

# PMFastR

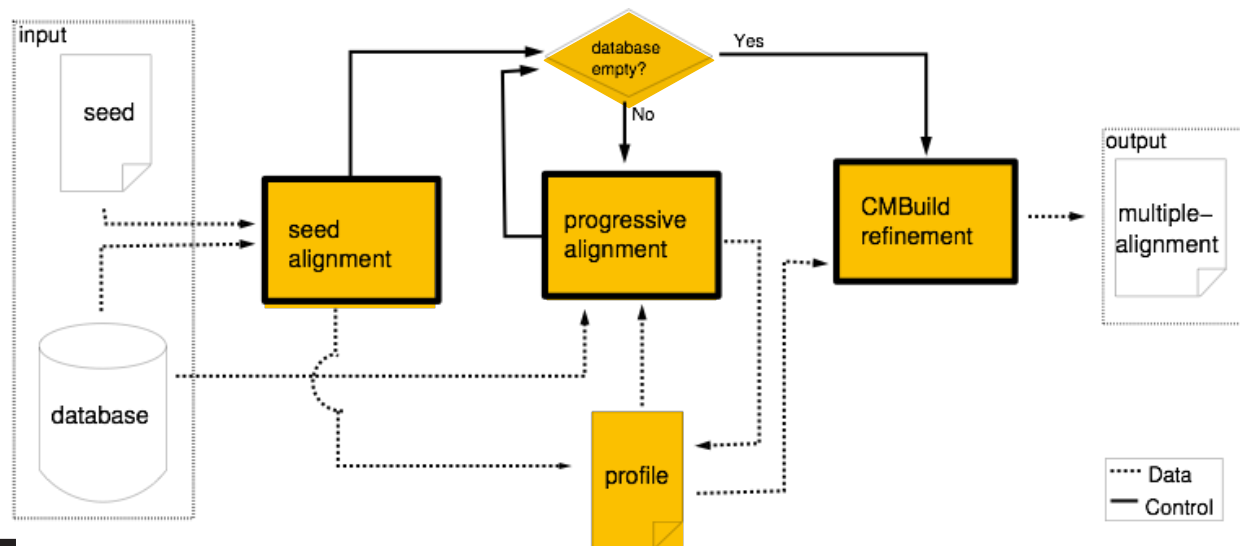
- Given:
  - One sequence with structure
  - Database of sequences without
- Output multiple alignment with structure





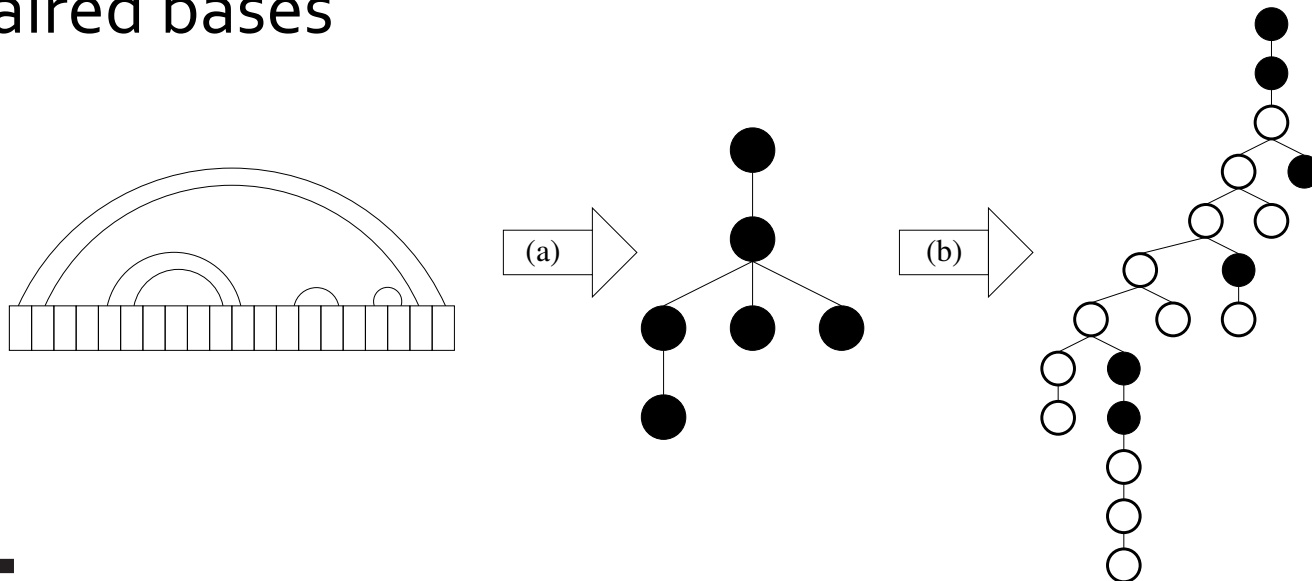
# PMFastR

- Align the sequence with structure to one sequence from the database
- This becomes the input to the next alignment
- Align this profile with another sequence from the DB
- Repeat until all sequences have been aligned
- Run *CMBuild-refine* to repair unpaired regions removed in the alignment procedure



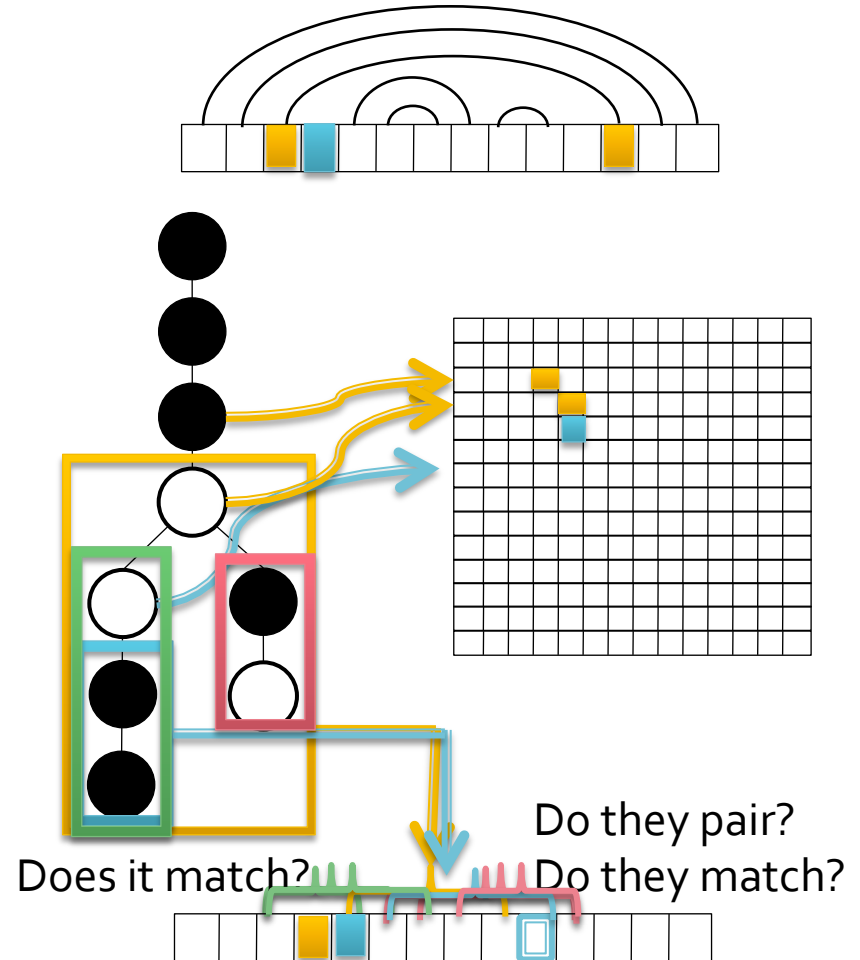
# Binarized trees encode the RNA sequence and structure

- From Bafna, *et al.* 1995(a), and Zhang, *et al.* 2004(b)
- Each base pair is a solid node
- Parental structure because no pseudoknots
- Dotted nodes indicate bifurcation points or unpaired bases



# Alignment is done by traversing the binarized tree

- Each node has its own dynamic programming table
- Each cell represents a segment (two locations) in the target that align to that particular node for solid nodes
- Uses the best calculated results from its children to find its answer
- Trace from the root of the tree to find the alignment



# Alignment algorithm

```

procedure PAln
  (*M is the set of base-pairs in RNA profile R. M' is the augmented set. *)
  for all nodes  $v \in M'$ ,
    all intervals  $(i, j)$ ,  $l_v - \text{band} \leq i \leq l_v + \text{band}$  and  $r_v - \text{band} \leq j \leq r_v + \text{band}$ 
    if  $v \in M$ 
      
$$\text{value} = \max \begin{cases} \text{mapRetrieve}(\text{child}(v), i + 1, j - 1) + \delta(l_v, r_v, t[i], t[j]), \\ \text{mapRetrieve}(v, i, j - 1) + \gamma(' - ', t[j]), \\ \text{mapRetrieve}(v, i + 1, j) + \gamma(' - ', t[i]), \\ \text{mapRetrieve}(\text{child}(v), i + 1, j) + \gamma(l_v, t[i]) + \gamma(r_v, ' - '), \\ \text{mapRetrieve}(\text{child}(v), i, j - 1) + \gamma(l_v, ' - ') + \gamma(r_v, t[j]), \\ \text{mapRetrieve}(\text{child}(v), i, j) + \gamma(l_v, ' - ') + \gamma(r_v, ' - '), \end{cases}$$

    else if  $v \in M' - M$ , and  $v$  has one child
      
$$\text{value} = \max \begin{cases} \text{mapRetrieve}(\text{child}(v), i, j - 1) + \gamma(r_v, t[j]), \\ \text{mapRetrieve}(\text{child}(v), i, j) + \gamma(r_v, ' - '), \\ \text{mapRetrieve}(v, i, j - 1) + \gamma(' - ', t[j]), \\ \text{mapRetrieve}(v, i + 1, j) + \gamma(' - ', t[i]), \end{cases}$$

    else if  $v \in M' - M$ , and  $v$  has two children
      
$$\text{value} = \max_{i \leq k \leq j} \{ \text{mapRetrieve}(\text{left\_child}(v), i, k - 1) + \text{mapRetrieve}(\text{right\_child}(v), k, j) \}$$

    end if
     $\text{mapSet}(v, i, j, \text{value})$ 
  end for

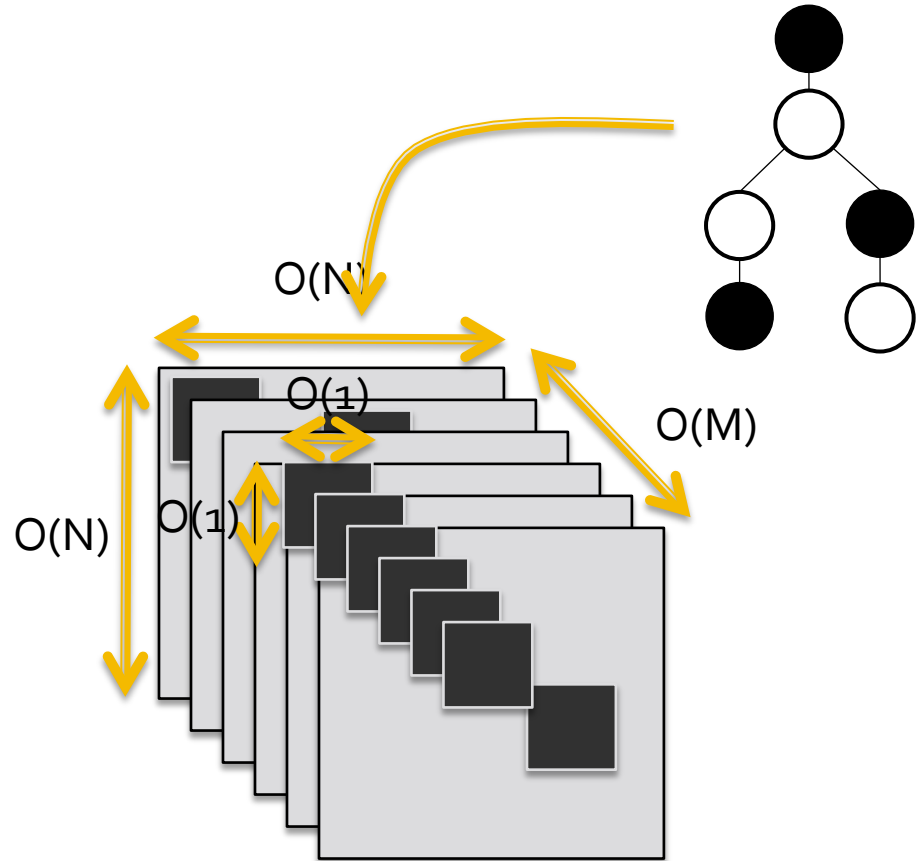
```

# Tree Based Alignment

- For solid nodes, find the max score of
  - Matching bases
  - Match left, insert right
  - Insert left, match right
  - Delete left
  - Delete right
  - Gap left and right
- For unpaired base nodes (dotted with one child), find the max score of
  - Match base
  - Delete
  - Insert left
  - Insert right
- For Bifurcation nodes (dotted with two children), find the max score of
  - For each split of the covered area, sum the score for the two children

# Banding reduces running time and memory consumption

- PMFastR is doing a global alignment
- We can assume that the location of the node  $v$  in the target will be in a similar location in the target
- Search and store only those locations
- There is still a 2-D array for each node
- This array reduces from  $n^2$  to  $band^2$



# Banding reduces running time and memory consumption

- This becomes very important for large sequences such as 16S and 23S rRNA
- Where  $b$  is the banding constant, most times set to be  $< 300$
- $b$  needs to be larger than the difference in lengths of the sequences
- Nawrocki and Eddy (PLoS, 2007) are using a similar idea to align 16S rRNA using Covariance Models

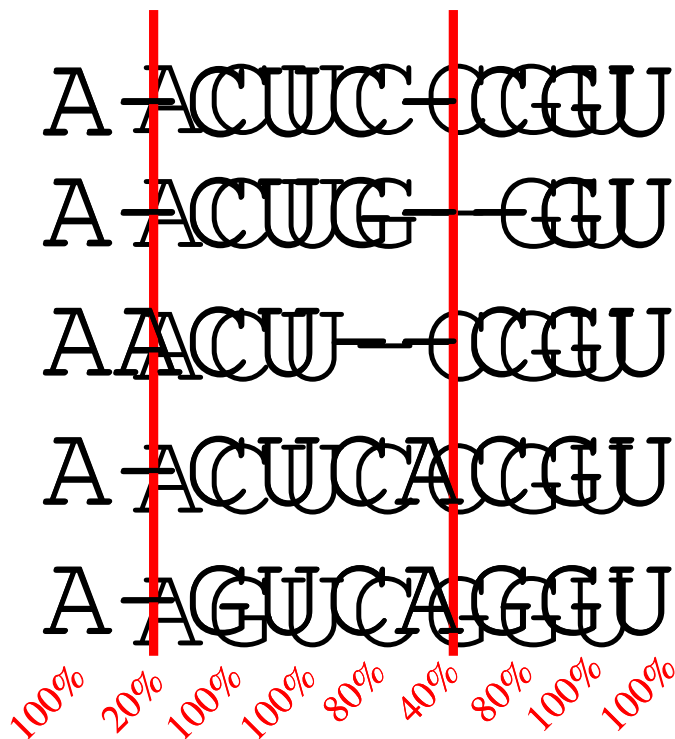
	Without Banding	Banded
Number of Bases	16K	16K
Space Consumption (Order)	$O(MN^2)$	$O(Mb^2)$
Space Consumption (Theoretical)	~3.8 GB	~137 MB (Assuming $b=300$ )

# Issues that arise using banding

- PMFastR does a global alignment
- ncRNA within the same family has a similar length
- The length can be quite large
- As the profile grows the in height (number of sequences) it also expands in length
- For banding to be effective, the sequence lengths need to be similar

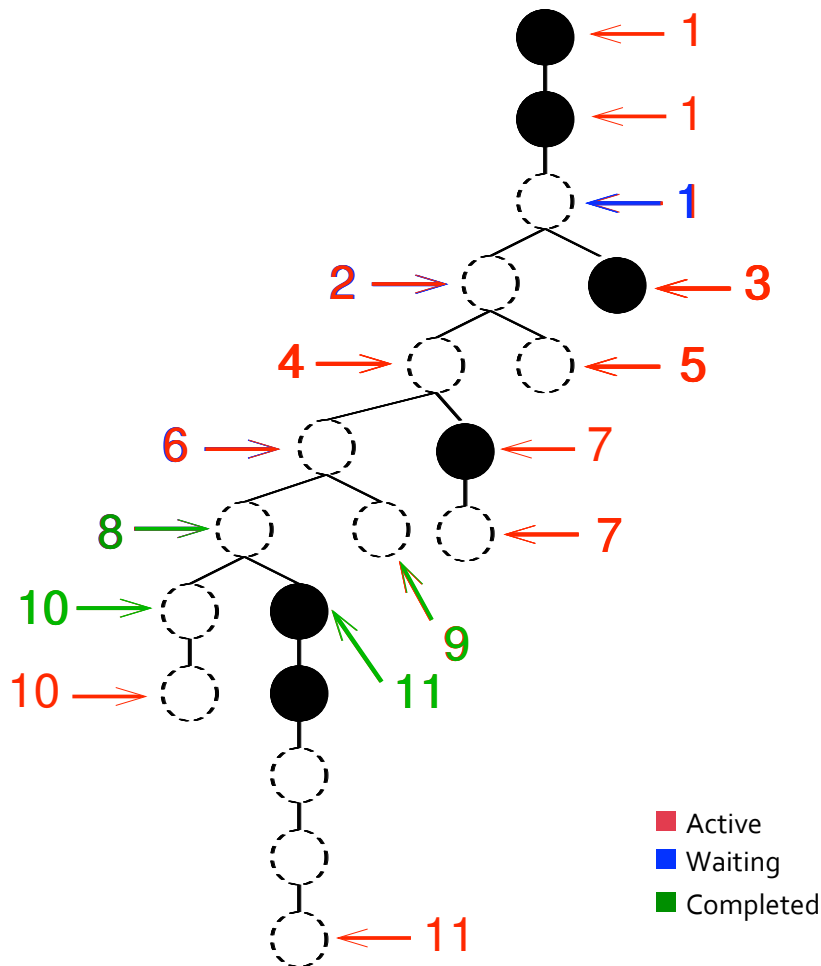


# Compacting the unpaired regions to increase quality and normalize the length



- Improve quality and reduce the size of the profile by removing columns in unpaired regions
- a predefined quality metric is the percentage of a column that must be present to not be removed
- present meaning not a blank character
- columns that encode structure are never removed

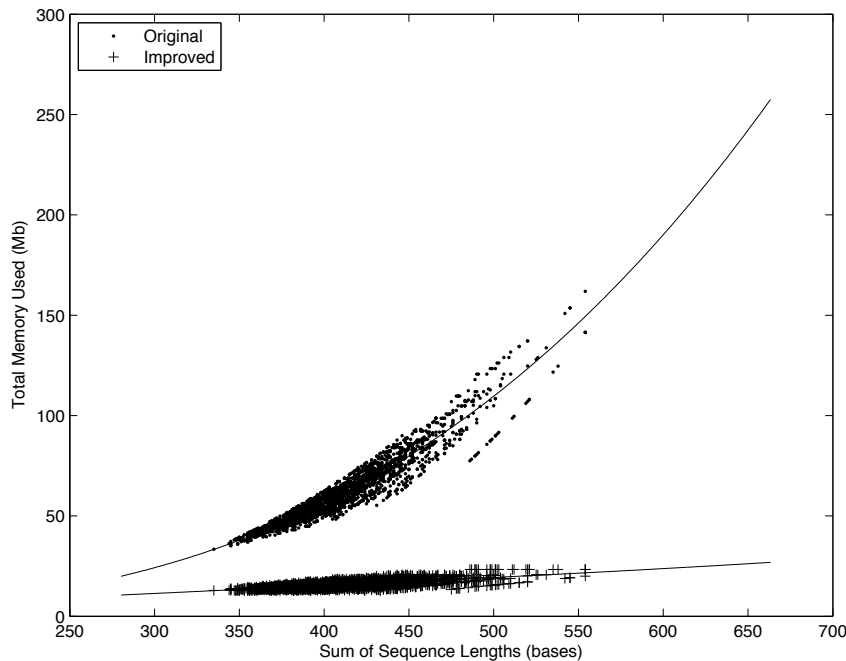
# Multithreading to increase the wall time



- each node depends on only its child nodes if they exist
- each of the children does not depend on each other
- the child nodes can then be run independently
- once both children have been computed, the parent can be computed
- this is recursive down the tree

# Results

# Memory Consumption Reduction



- Ran PMFastR and FastR on the same input set
- Cobalamin data
- 300 random sequence pairs
- X-axis: length of input
- Y-axis: memory
- Cubic regressions shown

# BRAlibase Benchmarking

[Wilm, *et al.* (*Algorithms Mol Biol.* 2006)]

- A carefully constructed set of sequence groups from the Rfam 7 release
- Groups contain 2,3,5,7,10 or 15 sequences
- Range in APSI from 39 to 50

# BRAliBase Benchmarking

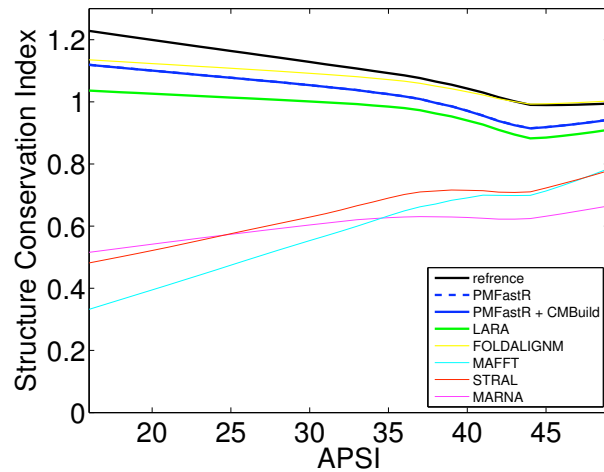
- SCI (Structure Conservation Index)
  - Measure of the percentage of bases that conserve structure
  - Uses AliFold to produce structure
- SCR (Structure Conservation Rate)
  - Similar to SCI
  - Uses input structure from the alignment
- SPS (Sum-of-Pairs Score)
  - Comparing with some reference, the ratio of the number of bases that are aligned in both the reference and the test alignment
  - Score of 1 means the alignments are exactly the same
- Compalign
  - Similar to SPS
  - Scores the bases not the locations in the sequence

# BRAliBase Benchmarking

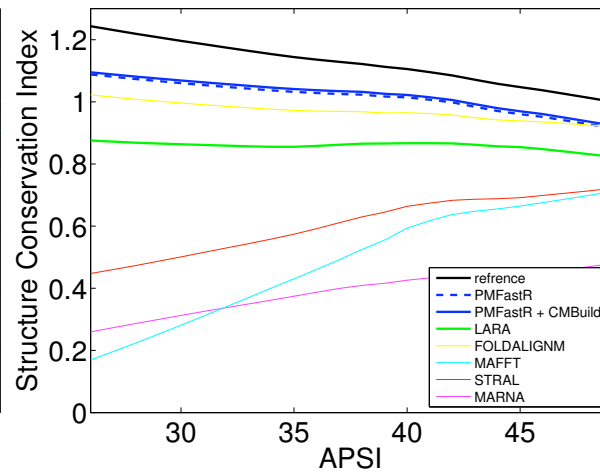
- LARA
  - Bauer, *et al.* 2007
  - Results to follow from supplemental data
- FoldAlign
  - Torarinsson, *et al.* 2007
- MAFFT
  - Katoh, *et al.* 2005
- STRAL
  - Dalli, *et al.* 2006
- MARNA
  - Siebert and Backofen, 2005

# BRAlIBase Benchmarking

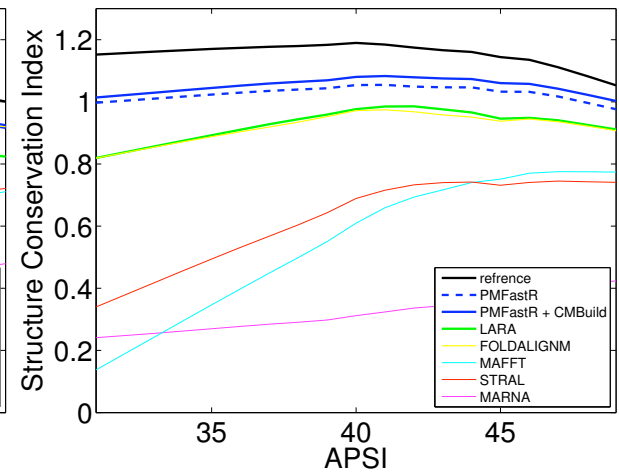
## SCI Results



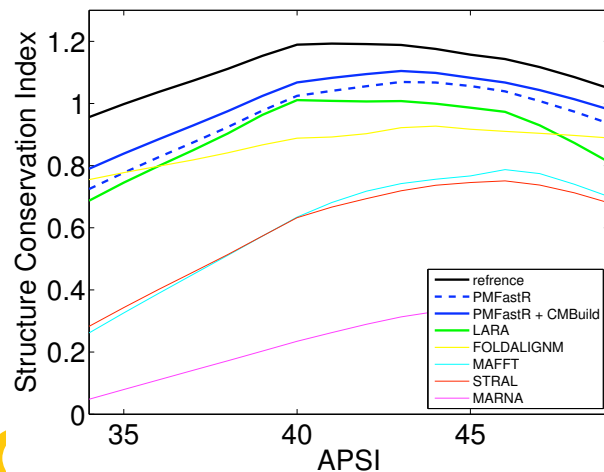
K<sub>2</sub>



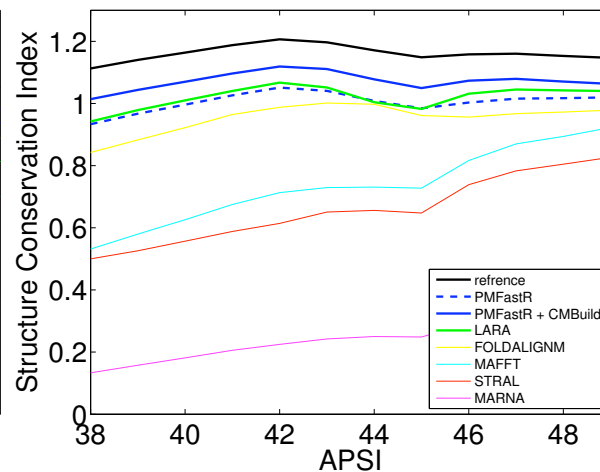
K<sub>3</sub>



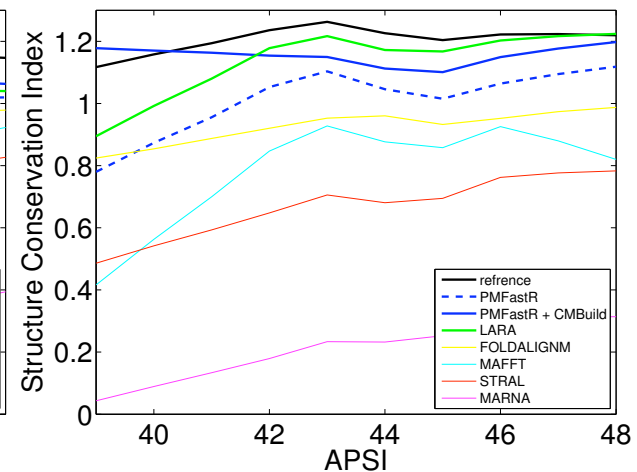
K<sub>5</sub>



K<sub>7</sub>



K<sub>10</sub>



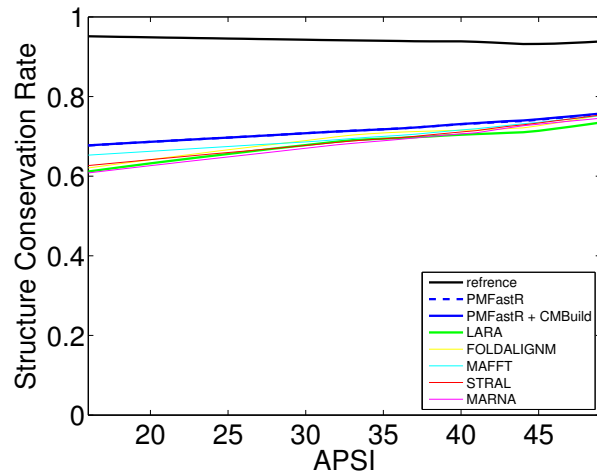
K<sub>15</sub>



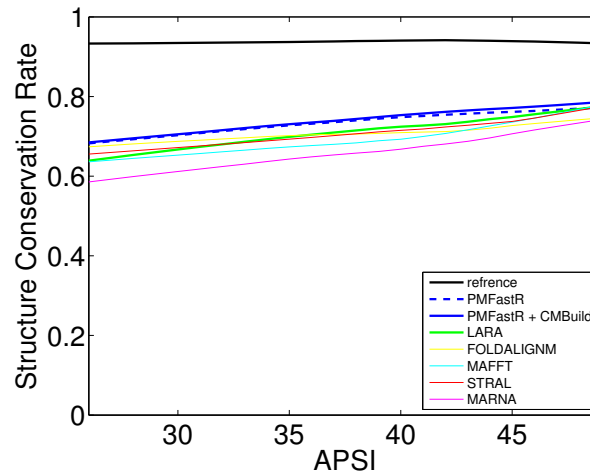


# BRAlIBase Benchmarking

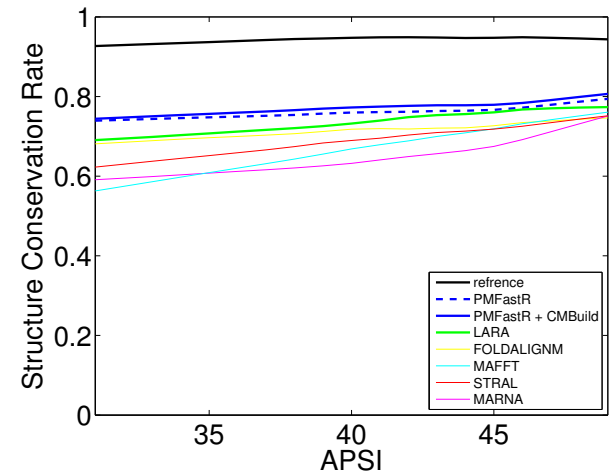
## SCR Results



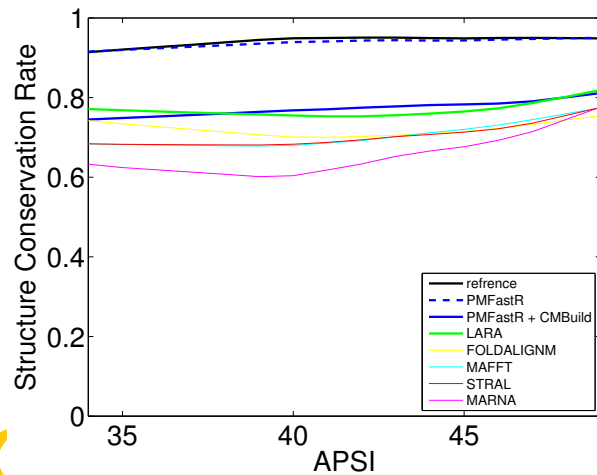
K2



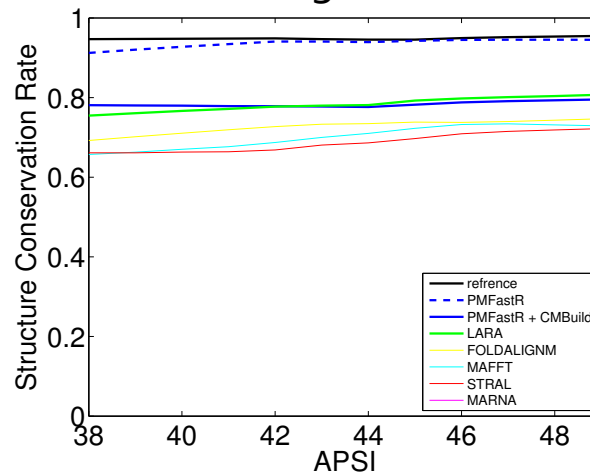
K3



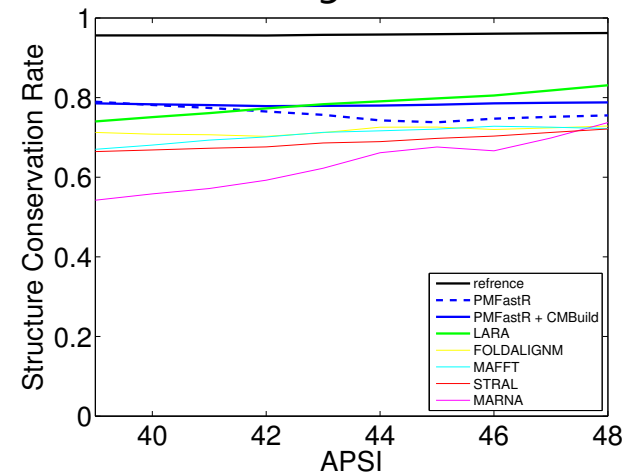
K5



K7



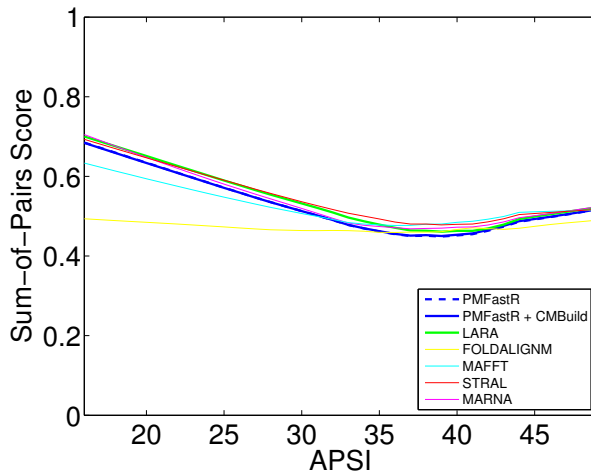
K10



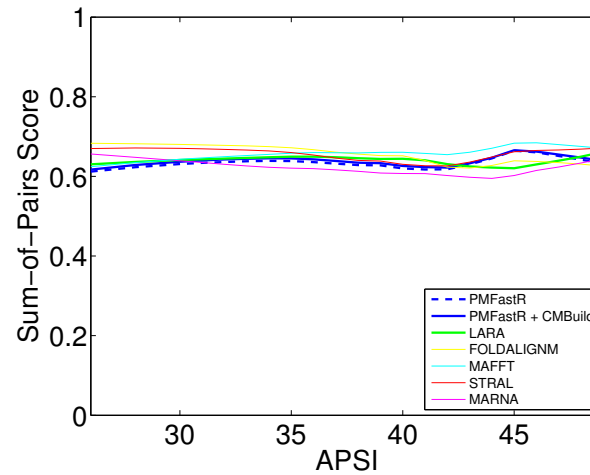
K15

# BRAlIBase Benchmarking

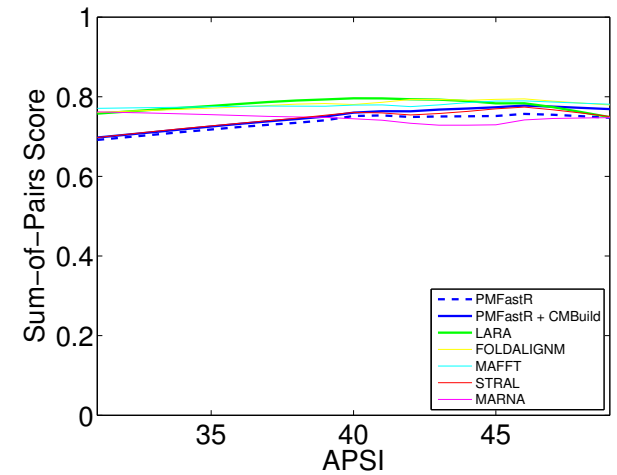
## SPS Results



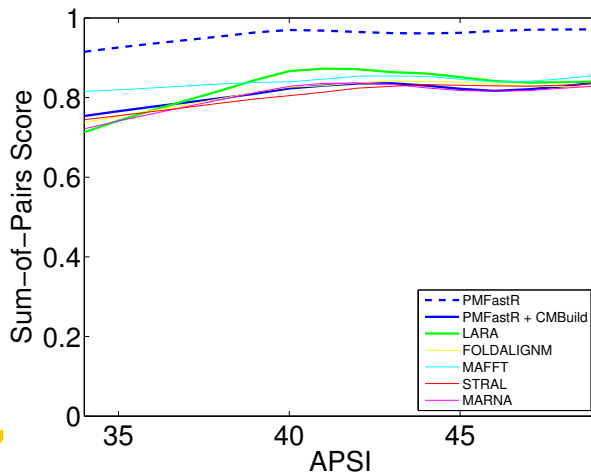
K2



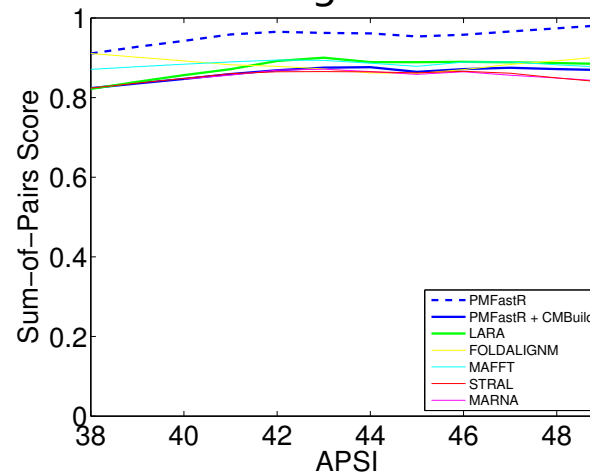
K3



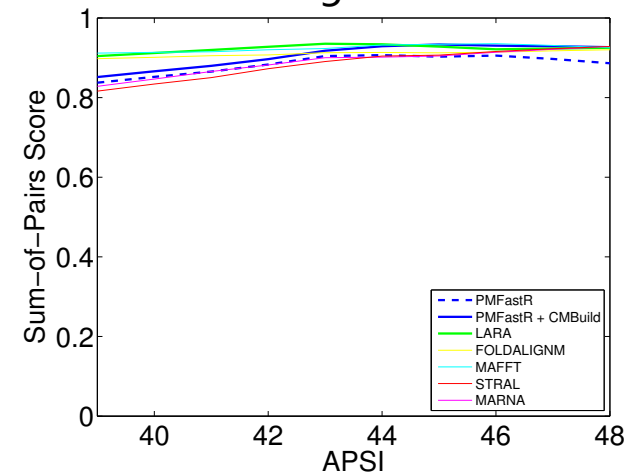
K5



K7



K10

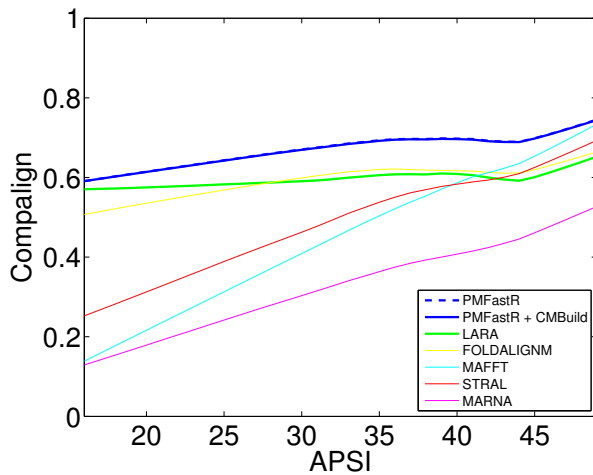


K15

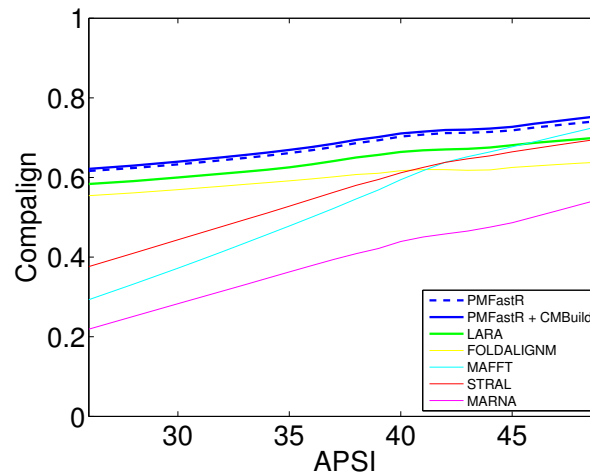


# BRAlIBase Benchmarking

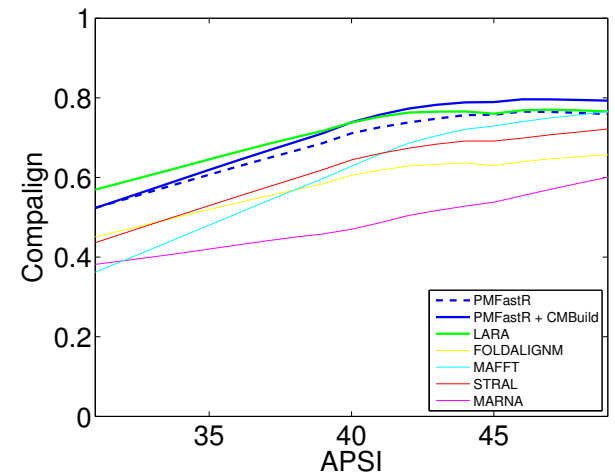
## Compalign Results



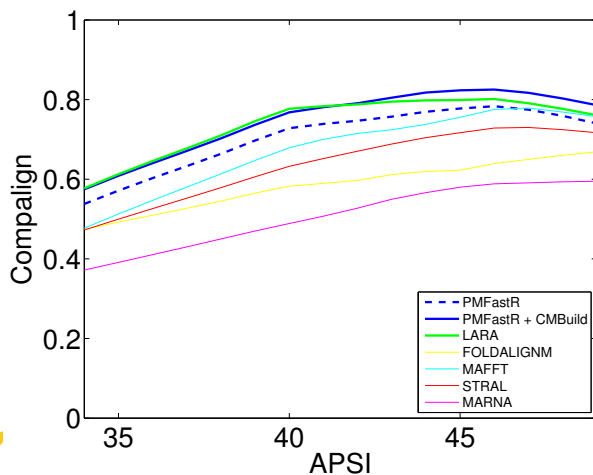
K2



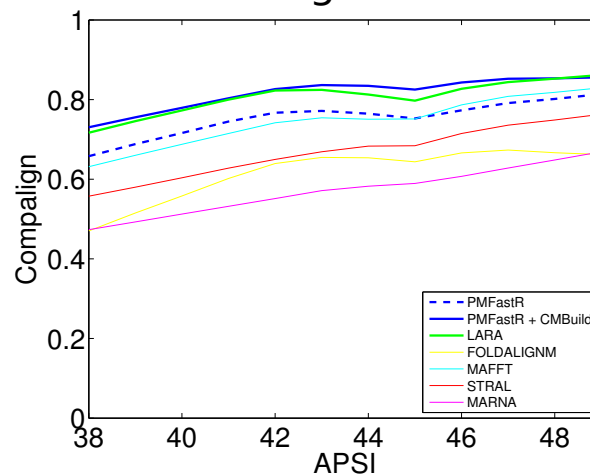
K3



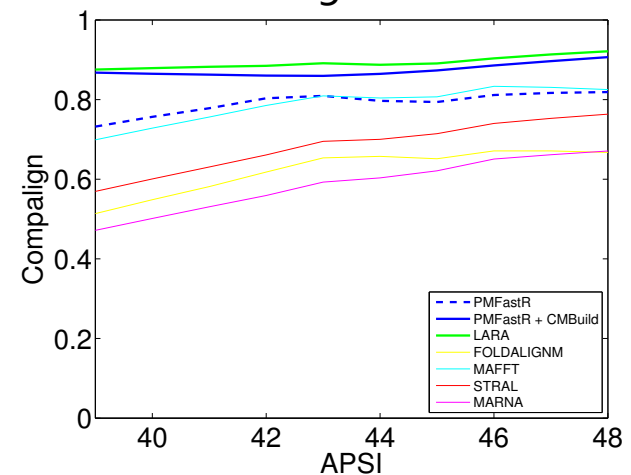
K5



K7



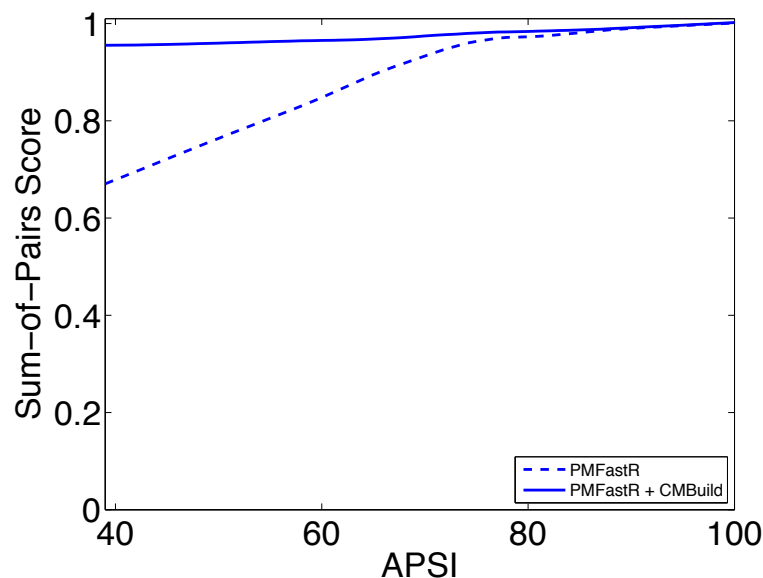
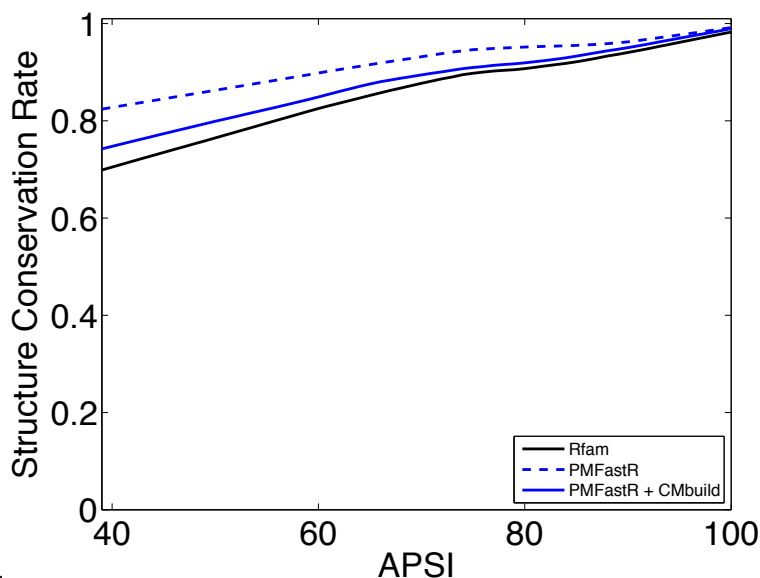
K10



K15

# Reconstructing the Rfam database to show that PMFastR produces high quality alignments

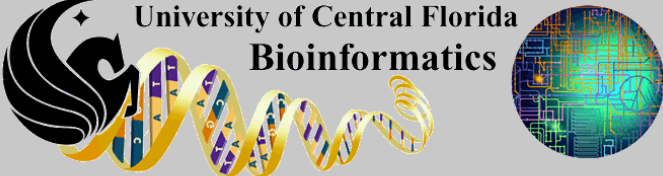
- Downloaded all families from Rfam 8.1
- Assigned the structure to one of the sequences
- Aligned remaining sequences using PMFastR
- SPS and SCR benchmarks shown



# Conclusion

# Conclusion

- Multiple alignment in a low amount of space using structure information for only one sequence
- Results comparable to hand made alignments
- Publicly available along with detailed results at <http://genome.ucf.edu/PMFastR>



University of Central Florida  
Bioinformatics

[PMFastR](#)

### PMFastR

#### Paper

**Multiple Alignment using Sequence Structure for Very Long Sequences**

**Abstract:** Many programs are available for the individual tasks of multiple alignment or sequence structure alignments. Here we present an algorithm based on FastR that not only does a multiple alignment using sequence structure, but it is done in such a way that the memory consumption is low enough for large sequences such as 16S and 23S rRNA. The algorithm also provides a method to utilize a multicore environment. We provide results with an empirical and real world comparisons to commonly used alignment references.

[\(pdf\)](#)

#### Supplemental Data

Rfam Comparison [\(link\)](#) (with CMBuild-refine)  
Overlap Comparison [\(link\)](#)  
BRAliBase Analysis [\(link\)](#)  
Source Code [\(tar.gz\)](#)

# Future Work

- Remove the Refinement step and work that into each iteration
- Apply the PMFastR algorithm to a database search
- What if the input could be a multiple alignment rather than a single sequence?