Multiple alignment by aligning alignments

Travis Wheeler
John Kececioglu

Department of Computer Science
University of Arizona

July 23, 2007
Intelligent Systems for Molecular Biology

Multiple sequence alignment

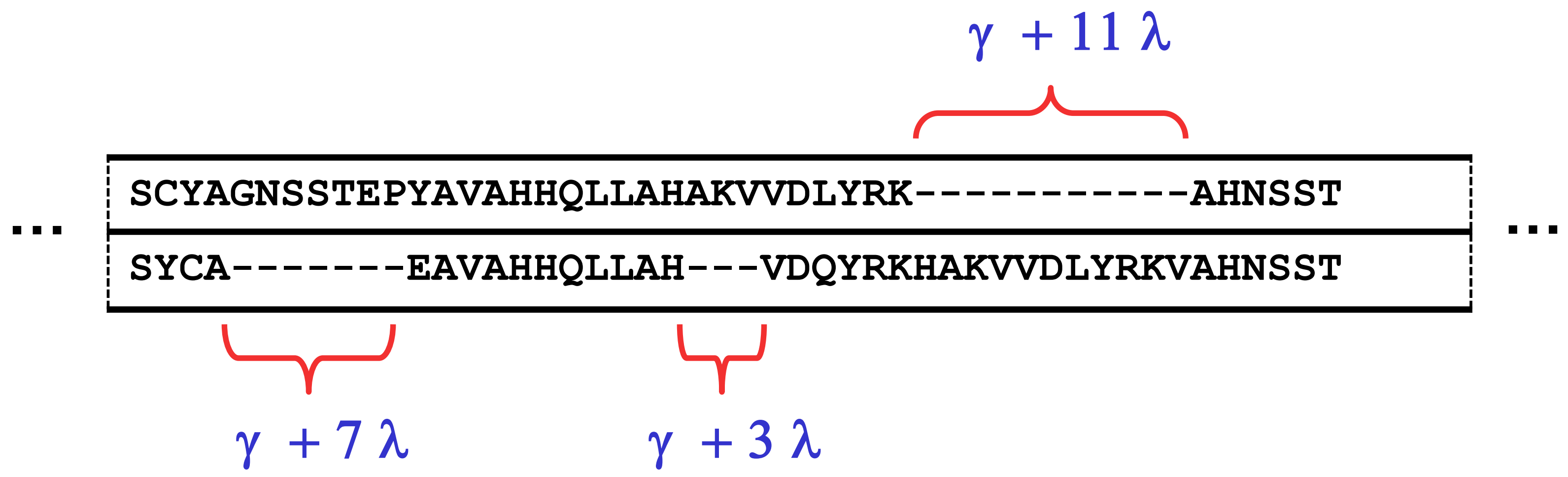
Sequence alignment central to computational biology

- Functional conservation
- Phylogenetic analysis
- Signals of selection
- Prediction of structure
- Comparative genomics
- and many others …

Aligning two sequences

A *two-sequence* alignment, with affine gap-costs, is scored,

$$\sum_{\text{columns}} \left(\begin{matrix}\text{substitution}\\ \text{score}\end{matrix}\right) + \lambda \left(\begin{matrix}\text{total gap}\\ \text{length}\end{matrix}\right) + \gamma \left(\begin{matrix}\text{number}\\ \text{of gaps}\end{matrix}\right)$$

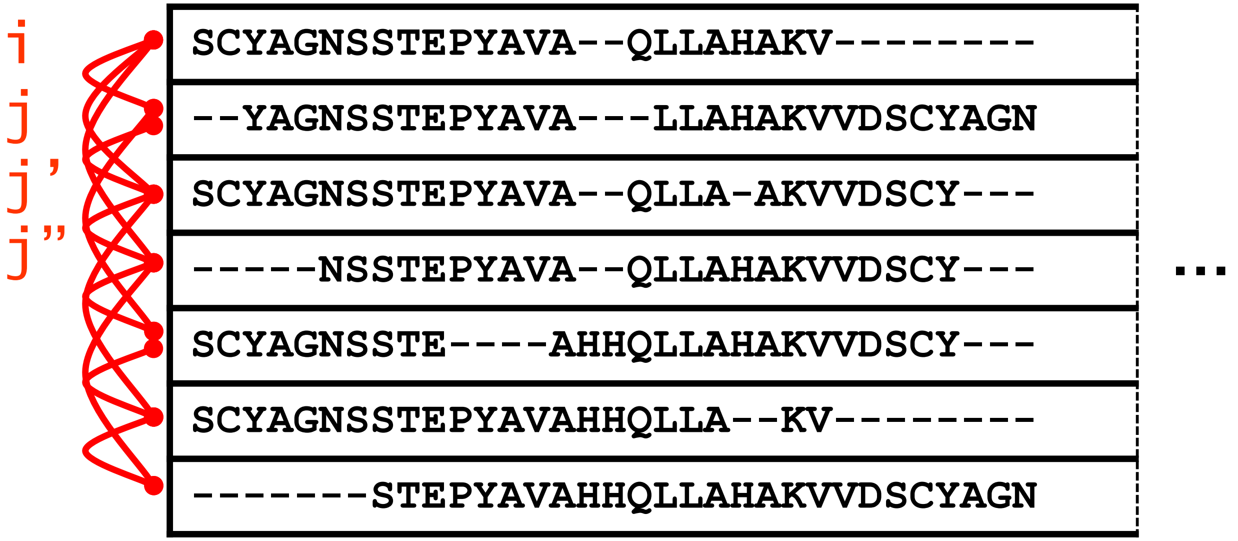


```
...  SCYAGNSSTEPYAVAHHQLLAHAKVVDLYRK-----------AHNSST  ...
     SYCA-------EAVAHHQLLAH---VDQYRKHAKVVDLYRKVAHNSST
```

Scoring a multiple alignment

Sum-of-pairs:

$$\sum_{i,j} w_{i,j} \;\; \text{score}(i, j”))$$



```
SCYAGNSSTEPYAVA--QLLAHAKV--------
--YAGNSSTEPYAVA---LLAHAKVVDSCYAGN
SCYAGNSSTEPYAVA--QLLA-AKVVDSCY---
-----NSSTEPYAVA--QLLAHAKVVDSCY---
SCYAGNSSTE----AHHQLLAHAKVVDSCY---
SCYAGNSSTEPYAVAHHQLLA--KV--------
-------STEPYAVAHHQLLAHAKVVDSCYAGN
```

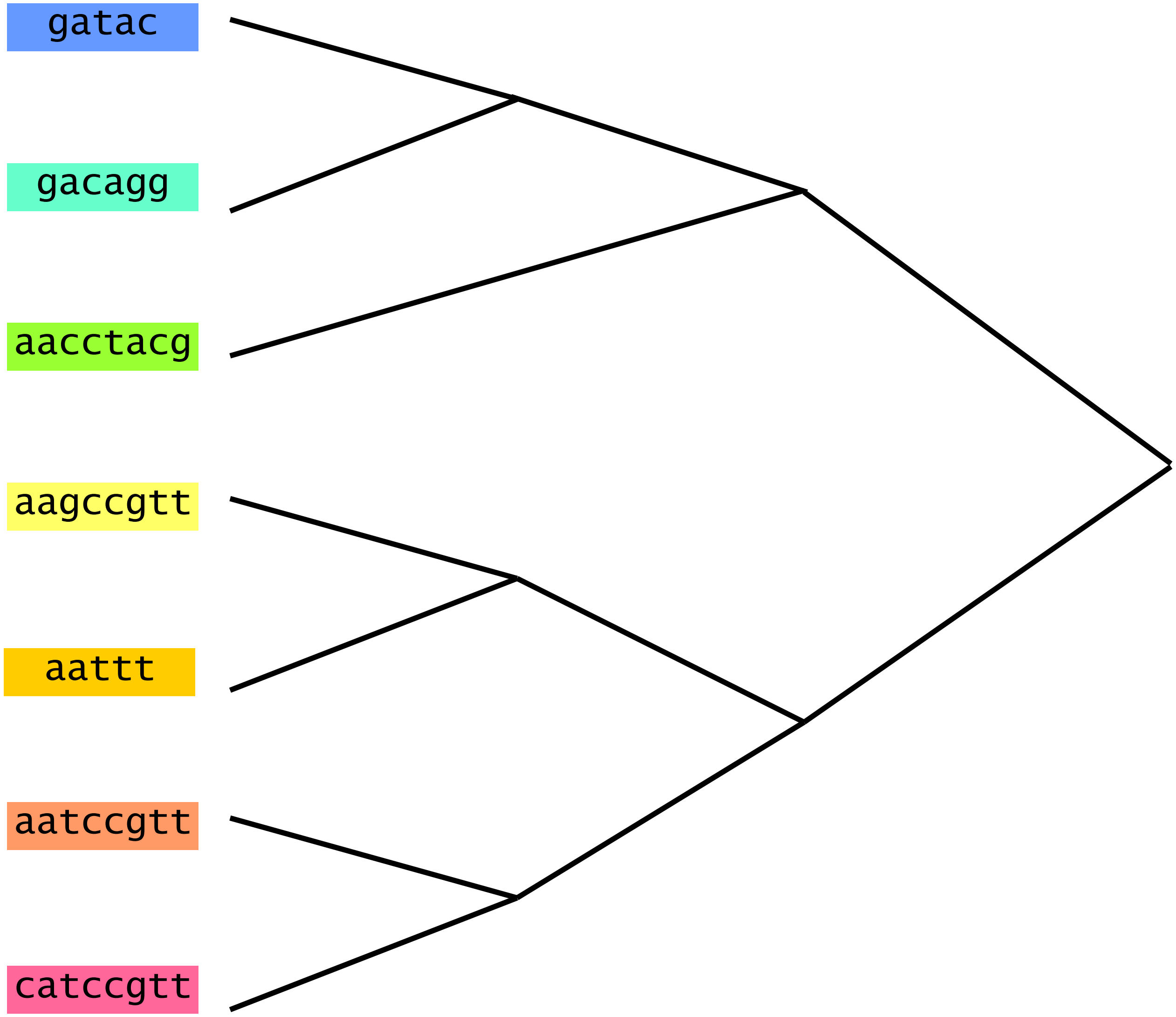
Optimal alignment of multiple sequences is NP-hard

Carrillo, Lipman 1988

Form-and-polish strategy

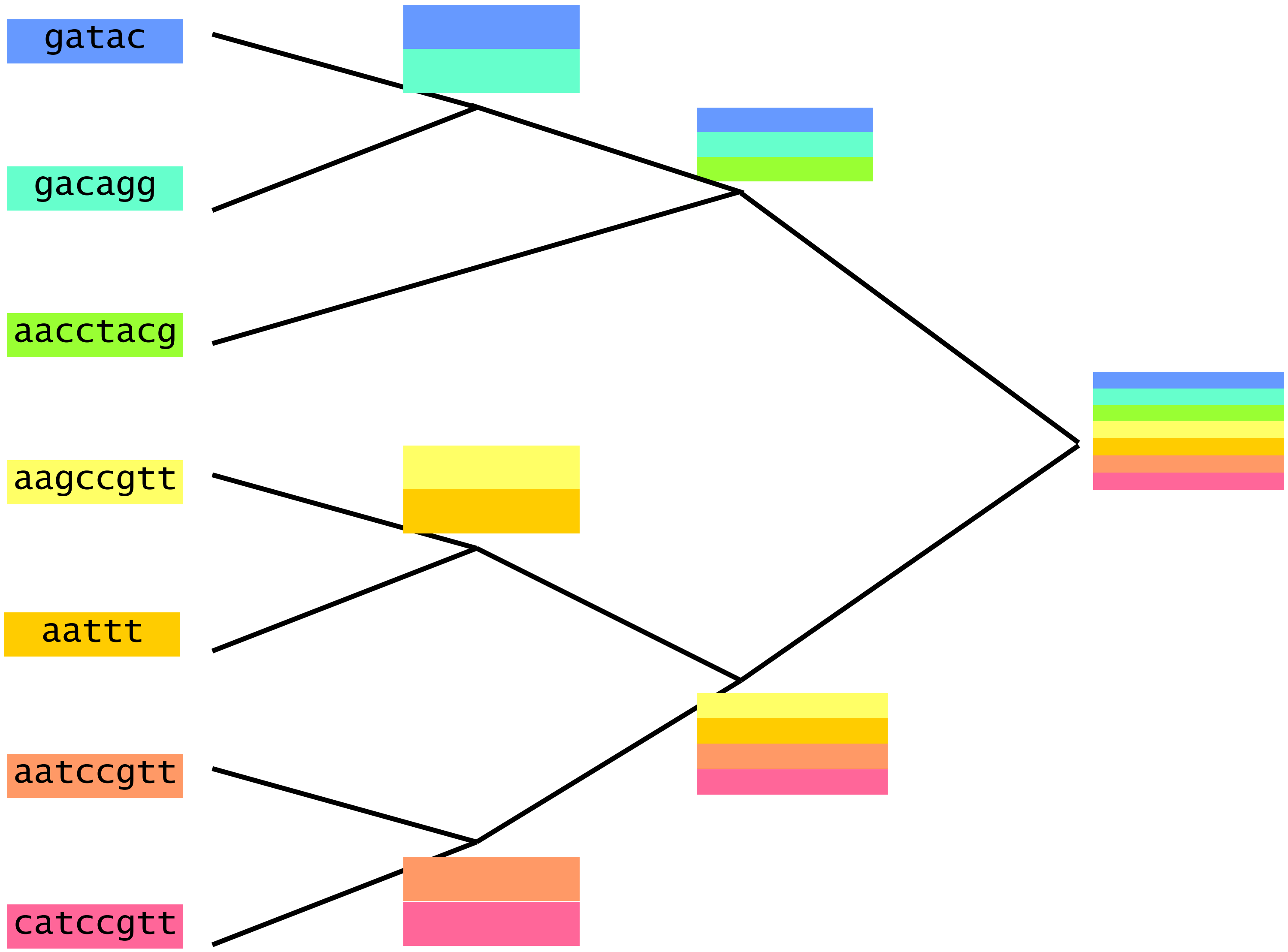
1. Choosing parameters
2. Constructing the merge tree
    a. Grouping sequences
    b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Constructing merge tree



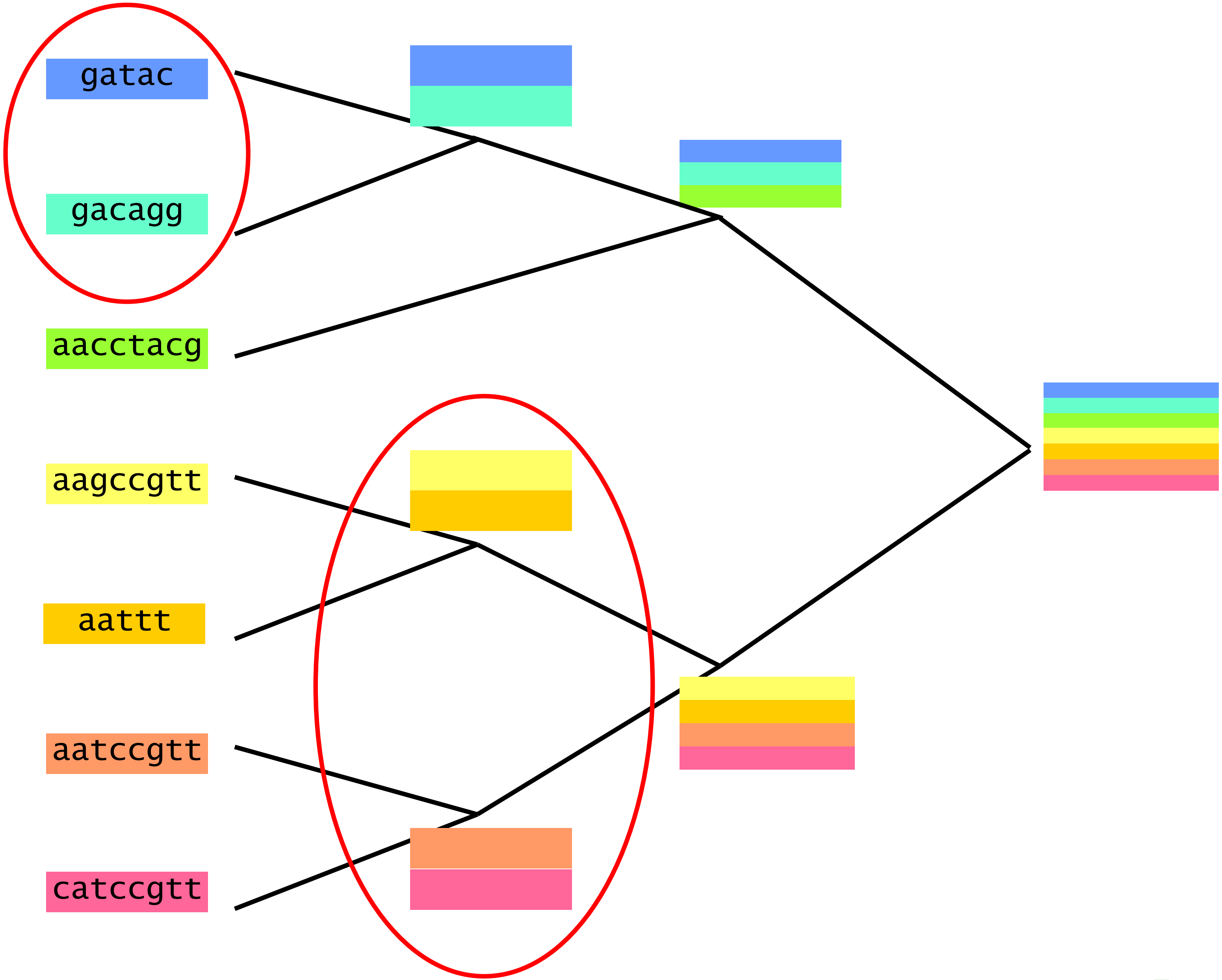
Feng, Doolittle 1987

Merging alignments



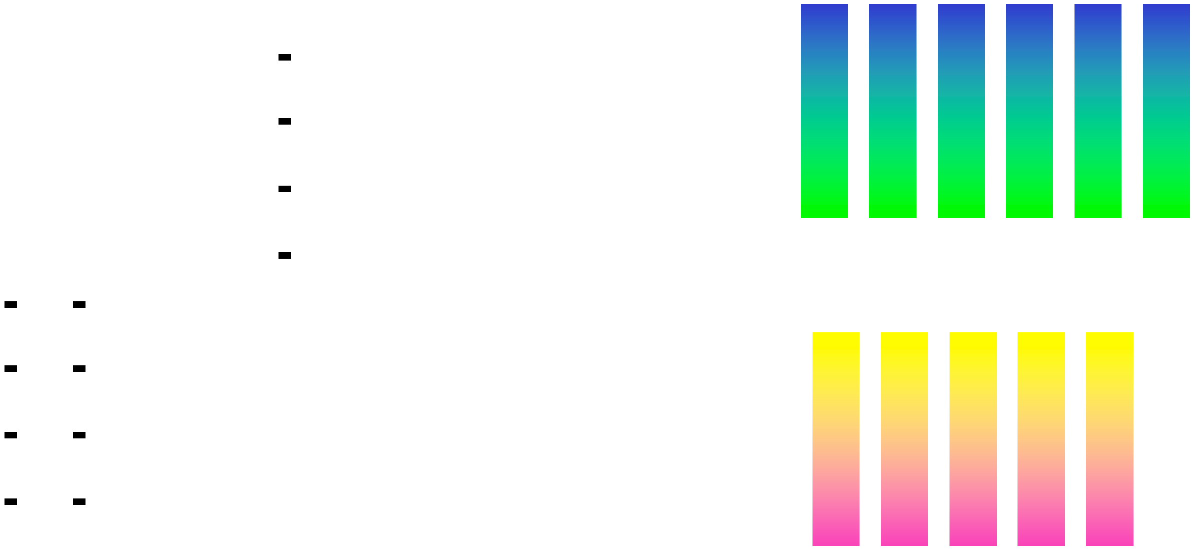
Feng, Doolittle 1987

Merging alignments

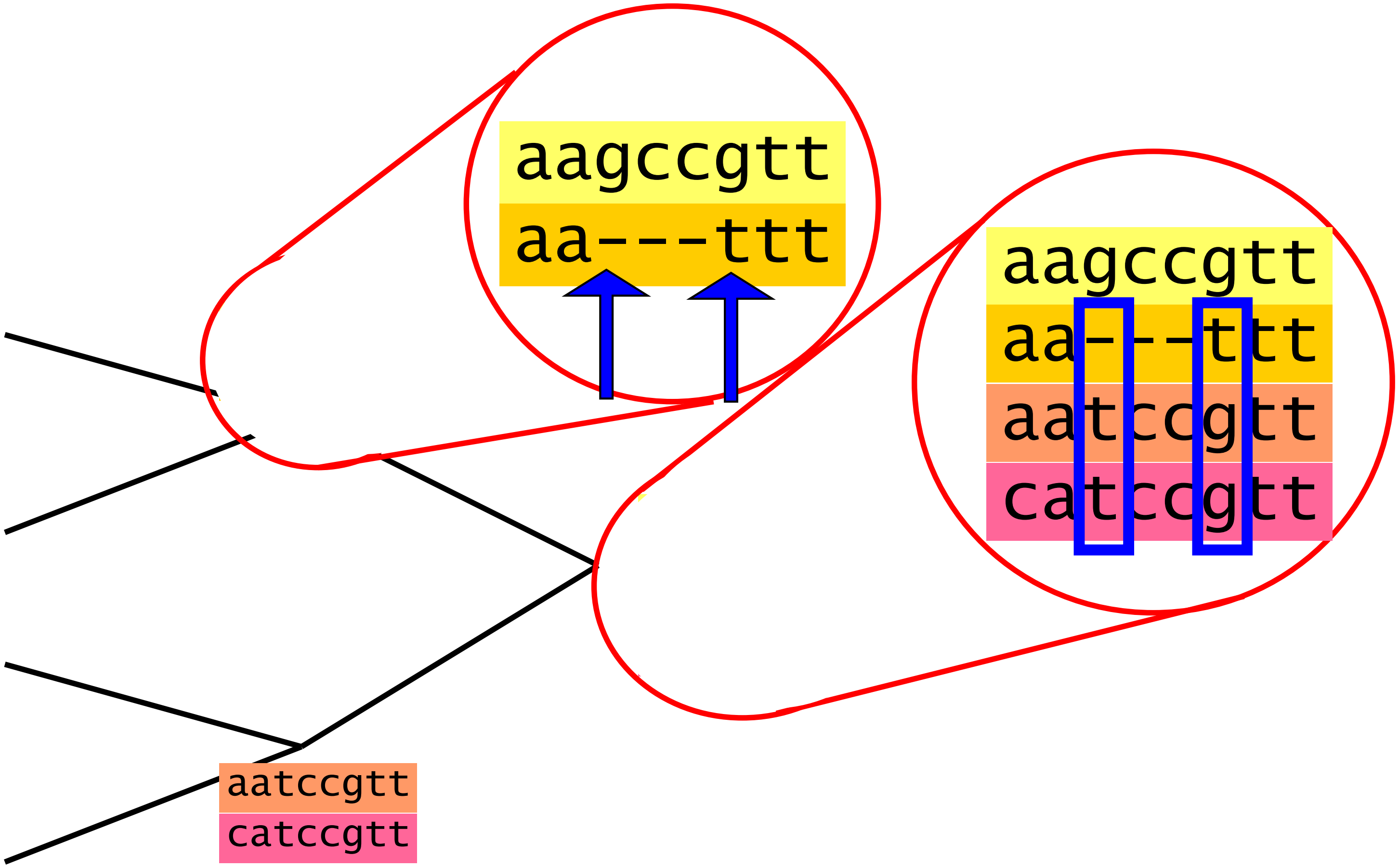


Feng, Doolittle 1987

Merging alignments

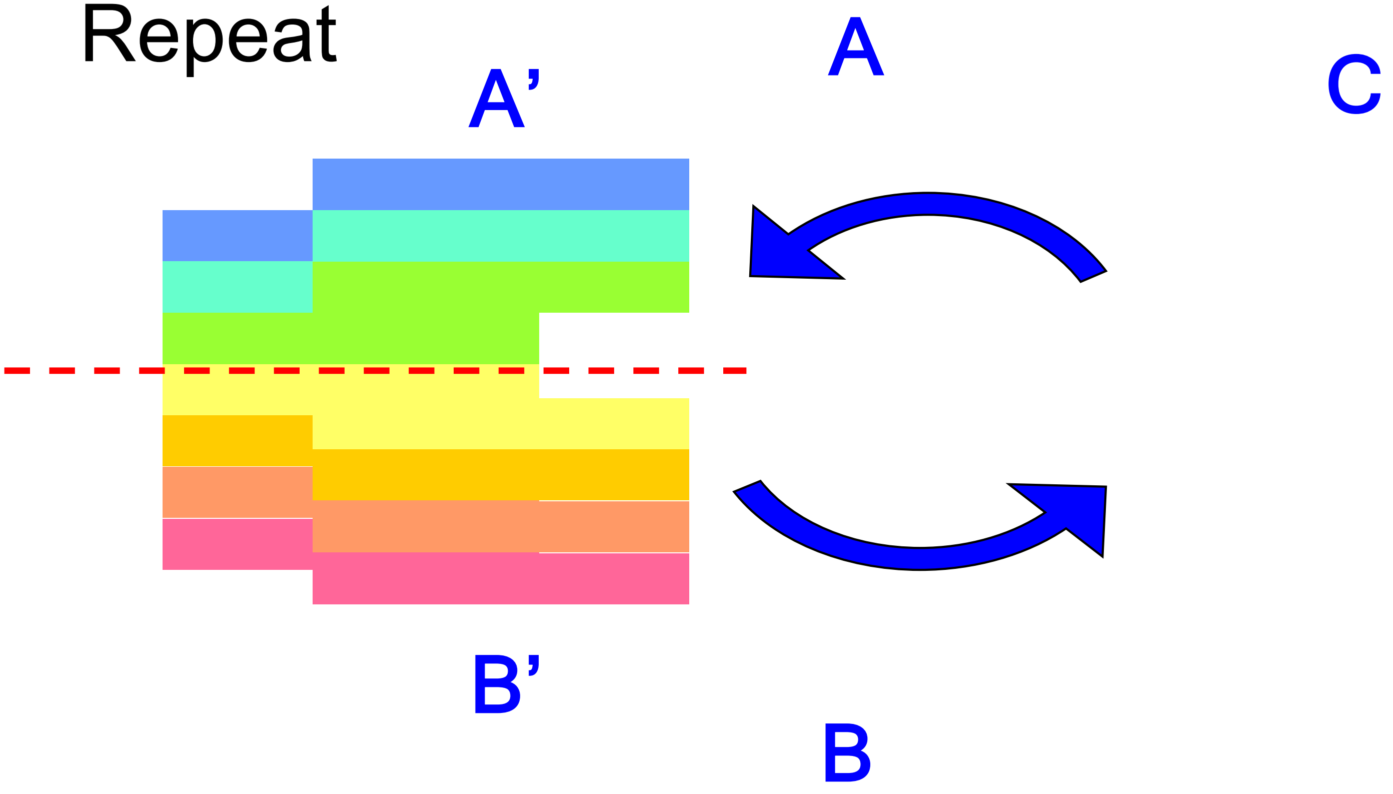


Merging alignments



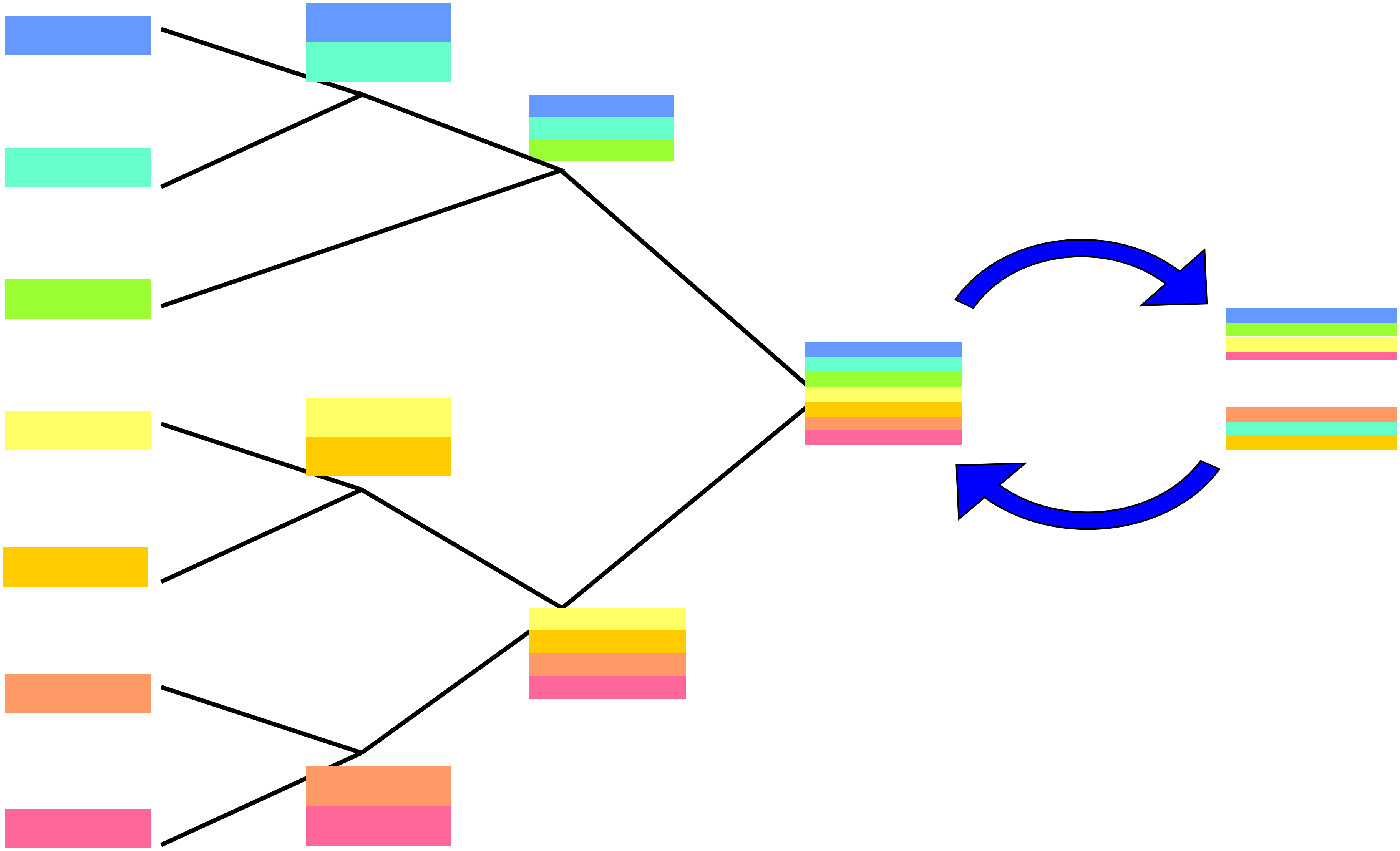
Polishing the alignment

- Split alignment into two groups
- Realign groups
- Repeat



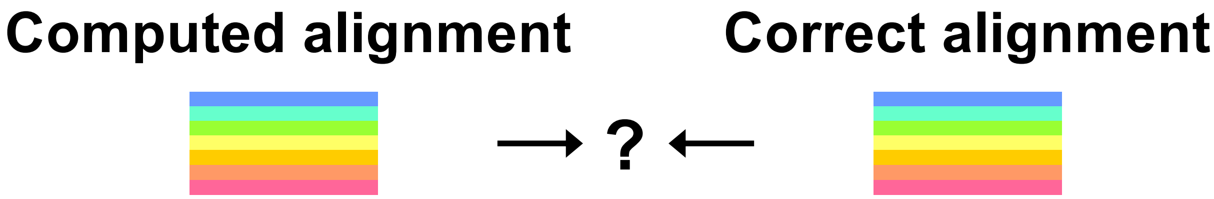
Berger, Munson 1991

Summary of main stages



- Construct tree
- Merge alignments
- Polish

Alignment quality



Computed alignment **Correct alignment**

$$\textbf{SPS score} = \frac{\#\text{ substitutions recovered}}{\#\text{ substitutions in correct alignment}}$$

$$\textbf{TC score} = \frac{\#\text{ columns correctly recovered}}{\#\text{ columns in correct alignment}}$$

Bahr et al. 2001

Benchmark datasets

- Benchmark suites
  - BAliBase [Thompson et al. 1999; Bahr et al. 2001]
  - PALI [Balaji et al 2001]
  - SABmark [Van Walle et al 2004]
  - All based on structural alignment
- Characteristics
  - 899 alignments
  - 10 sequences per alignment, on average
  - 400 columns per alignment, on average
- Core columns

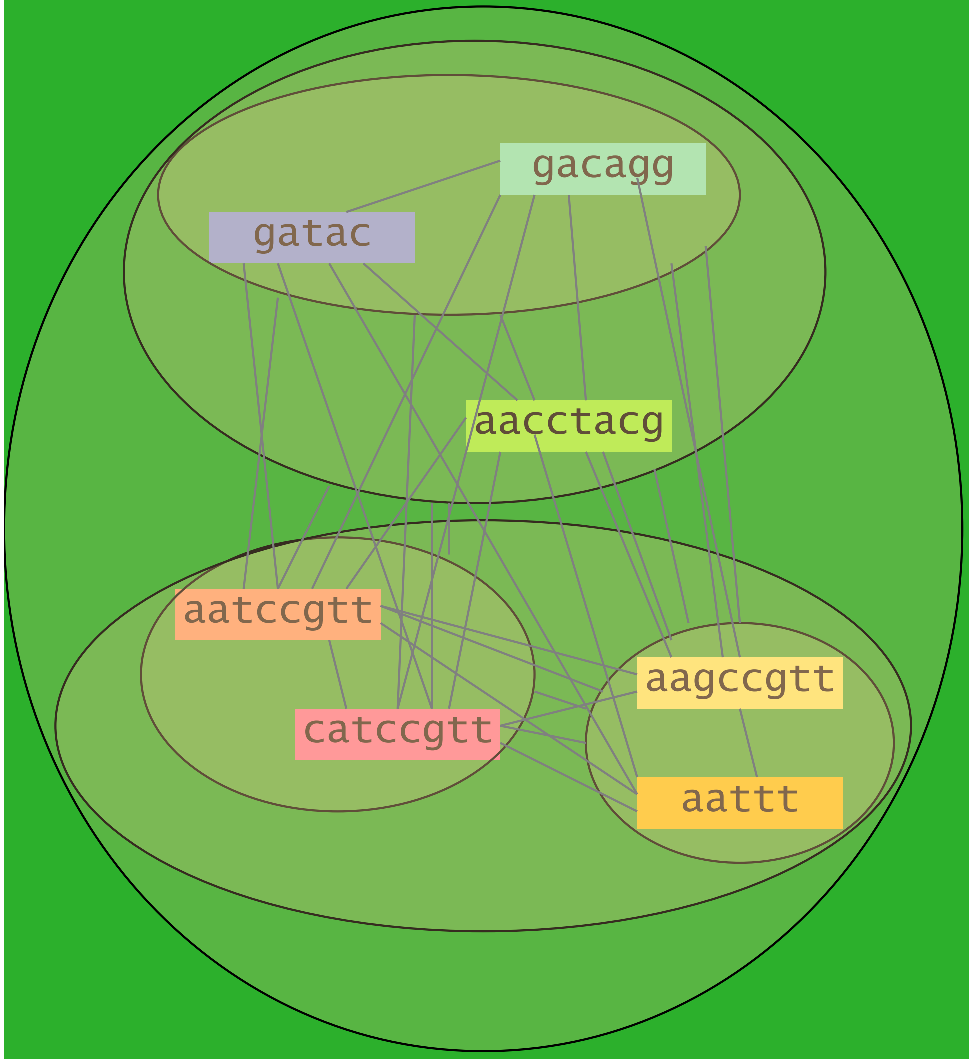
Form-and-polish review

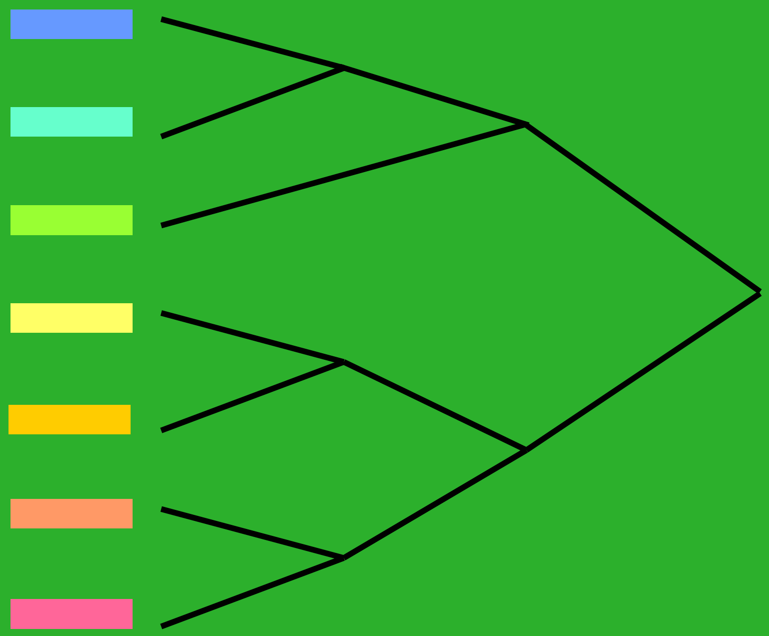
1. Choosing parameters
2. Constructing the merge tree
   a. Grouping sequences
   b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Grouping methods

- Neighbor joining (NJ) [Saitou, Nei 1987]
- Unweighted-pair group method with arithmetic mean (UPGMA) [Sneath, Sokal 1973]
- Minimum spanning tree (MST)
- Dynamic alignment distance (DAD)

Grouping sequences





- Methods differ in measuring distances for new groups

Comparing grouping methods

Grouping method	BAliBase	SABmark	PALI	Average
MST	**79.4**	**44.1**	-0.7	**67.8**
UPGMA	-1.4	-1.4	**80.5**	-0.7
NJ	-2.0	-2.0	-3.3	-2.2
DAD	-1.2	-0.6	-7.5	-2.9

- Best grouping method ≠ best phylogeny method

Form-and-polish review

1. Choosing parameters
2. Constructing the merge tree
   a. Grouping sequences
   b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Measuring distances

Percent identity

```
AHDHHSSQ
ANEH--TR
```

Compressed identity

```
AHDHHSSQ
ANEH--TR
```

Normalized alignment cost

```
AHDHHSSQ
ANEH--TR
```

Comparing distance methods

Tree method	BAliBase	SABmark	PALI	Average
Normalized cost	**81.6**	**48.2**	**83.0**	**70.9**
Compressed identity	-2.2	-4.1	-3.2	-3.1
Percent identity	-3.1	-4.7	-3.1	-3.6

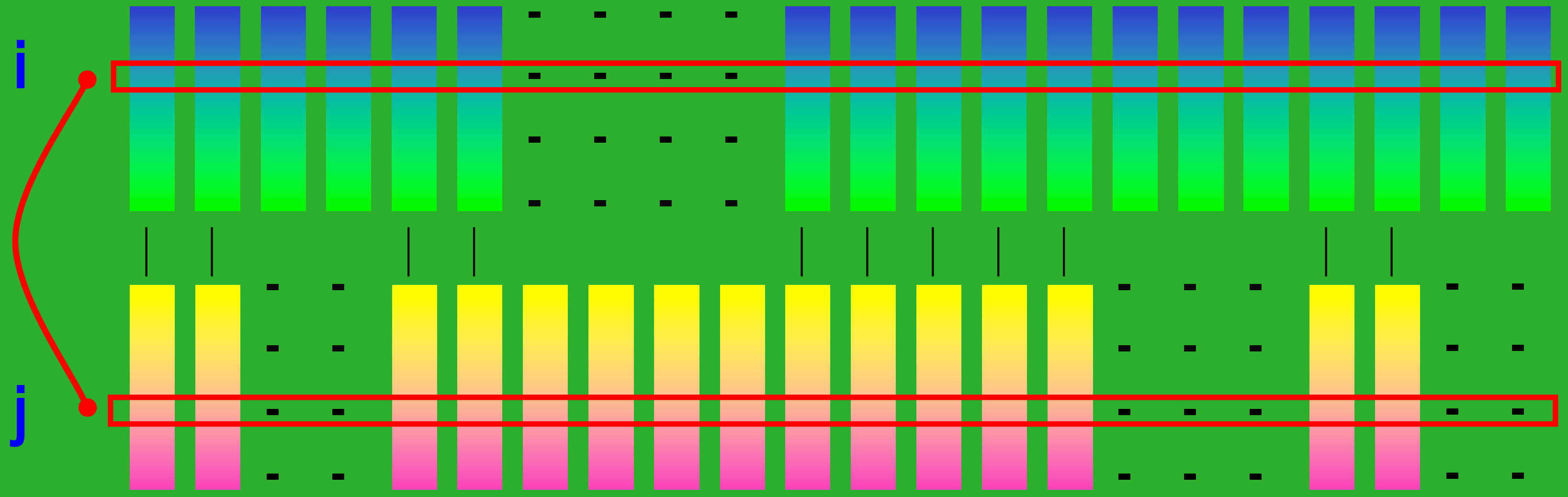
- Normalized cost is very simple, and gives greatest gains

Form-and-polish review

1. Choosing parameters
2. Constructing the merge tree
   a. Grouping sequences
   b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Aligning alignments

$$\sum_{i,j} w_{i,j} \left(\sum_{\text{columns}} \left(\begin{matrix}\text{substitution}\\ \text{score}\end{matrix} \right) + \lambda \left(\begin{matrix}\text{gap}\\ \text{length}\end{matrix} \right) + \gamma \left(\begin{matrix}\text{gap}\\ \text{count}\end{matrix} \right) \right)$$



Merging methods

Exact gap counts

[Gotoh 1993 ; Kececioglu, Starret 2004]

- k sequences, n columns
- $O(5^k n^2)$ worst case
- $O(k^2 n^2)$ time in practice

Pessimistic gap counts

[Altschul 1989; Kececioglu, Zhang 1998]

- Overestimates gap startups
- $O(kn + n^2)$ worst case
- 100-fold speedup for 20 sequences

Comparing merging methods

Merging method	BAliBase	SABmark	PALI	Average
Exact	**82.4**	**48.4**	**84.0**	**71.6**
Pessimistic	-0.8	-0.2	-1.0	-0.7

- Pessimistic heuristic may be sufficient for large inputs

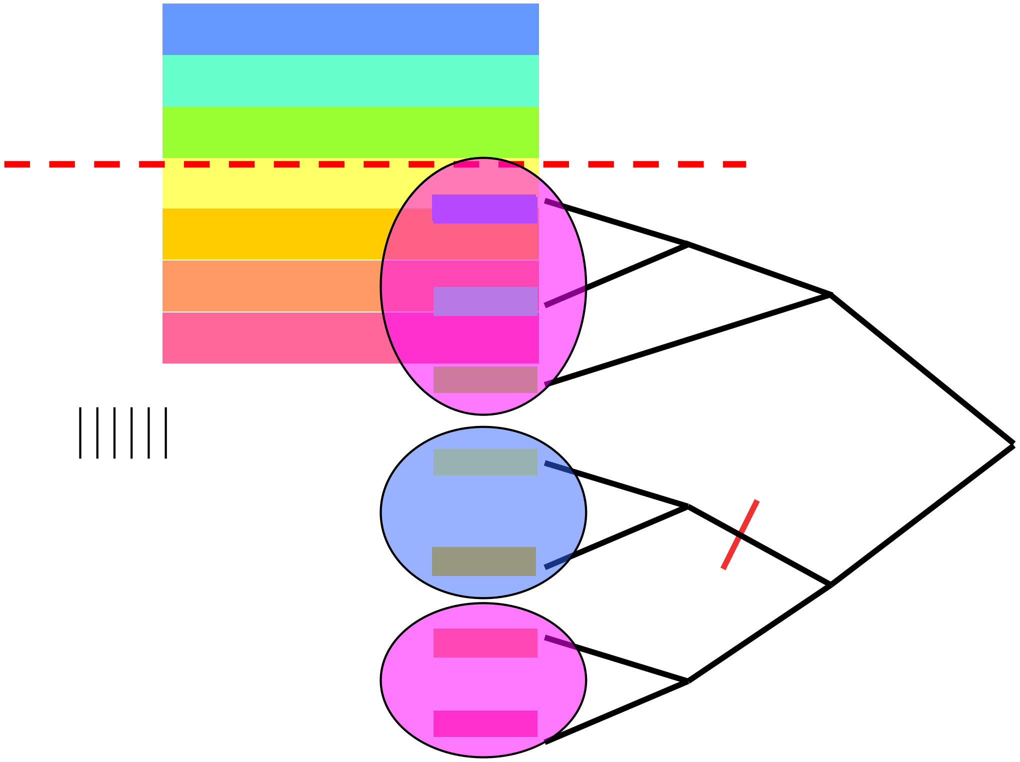
Form-and-polish review

1. Choosing parameters
2. Constructing the merge tree
    a. Grouping sequences
    b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Polishing methods

Two-cut method

- Random partition [Probcons Do et al. 2005]
- Tree-based partition



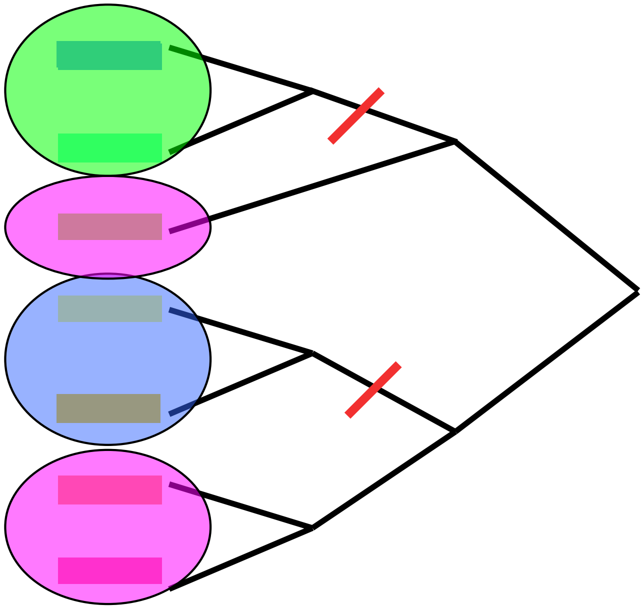
Polishing methods

Two-cut method

- Random partition [Probcons Do et al. 2005]
- Tree-based partition
  - Randomly cut edges [MAFFT Katoh et al. 2005]
  - Exhaustively cut edges [Muscle Edgar 2004]

Three-cut method

- Tree-based, random



Polishing methods

Two-cut method

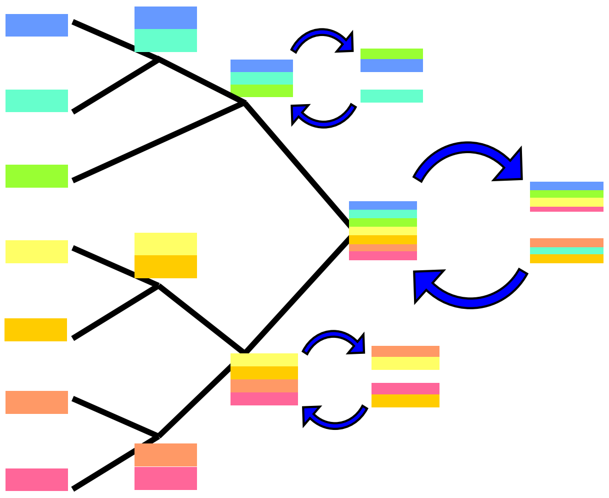
- Random partition [Probcons Do et al. 2005]
- Tree-based partition
  - Randomly cut edges [MAFFT Katoh et al. 2005]
  - Exhaustively cut edges [Muscle Edgar 2004]

Three-cut method

- Tree-based, random

On-the-fly method

[Subbiah, Harrison 1989]



Comparing polishing methods

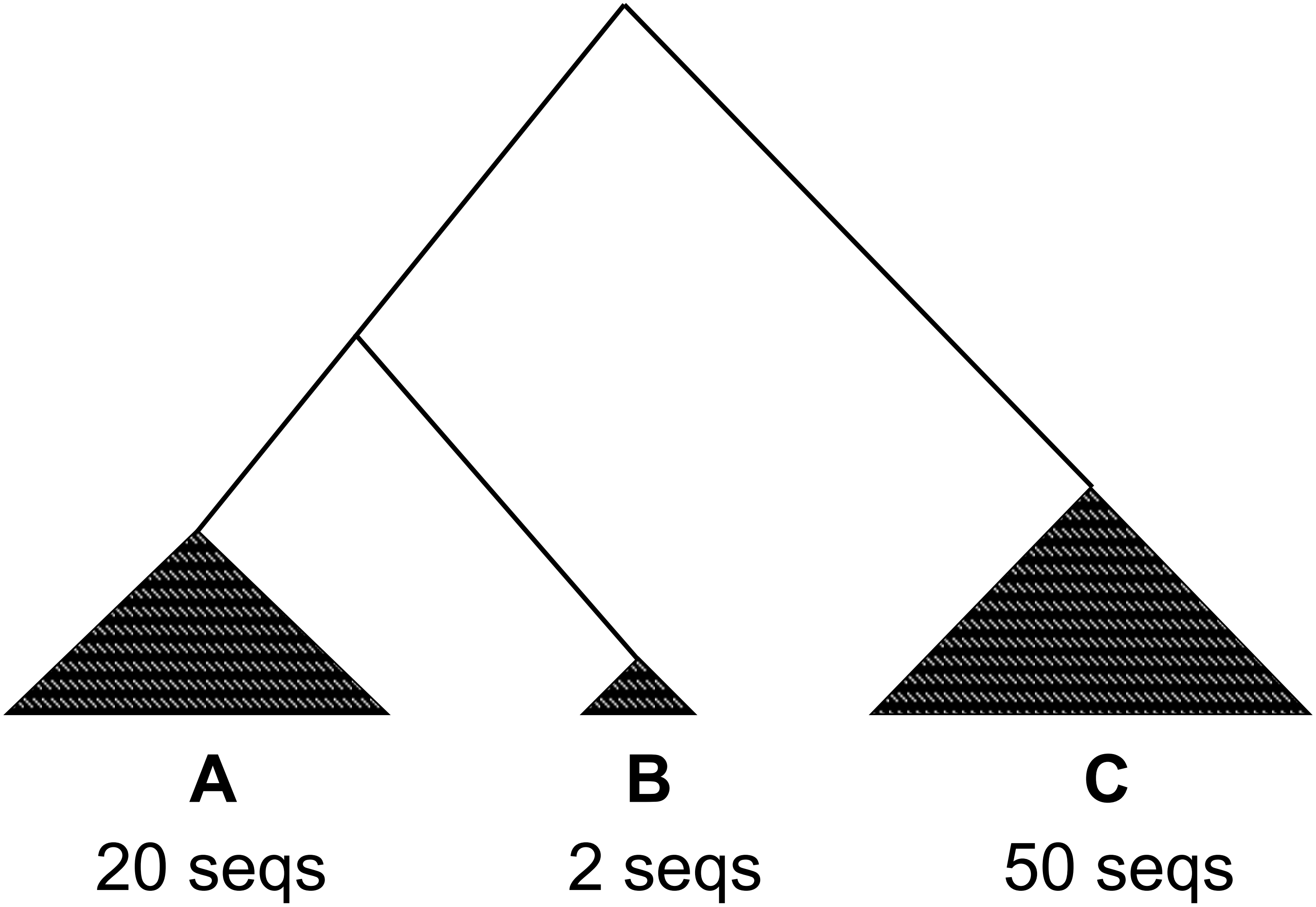
Polishing method	BAliBase	SABmark	PALI	Average
3-cut + on-the-fly	-0.1	**50.2**	-0.2	**73.1**
3-cut	-0.2	-0.5	**84.8**	-0.2
2-cut	**84.4**	-0.4	-0.1	-0.2
2-cut + on-the-fly	-0.8	-0.2	-0.3	-0.4
On-the-fly	-1.1	-0.6	-0.4	-0.7
none	-2.0	-1.8	-0.8	-1.5

- 3-cut achieves 2-cut quality in less time
- On-the-fly speeds up 2-cut convergence

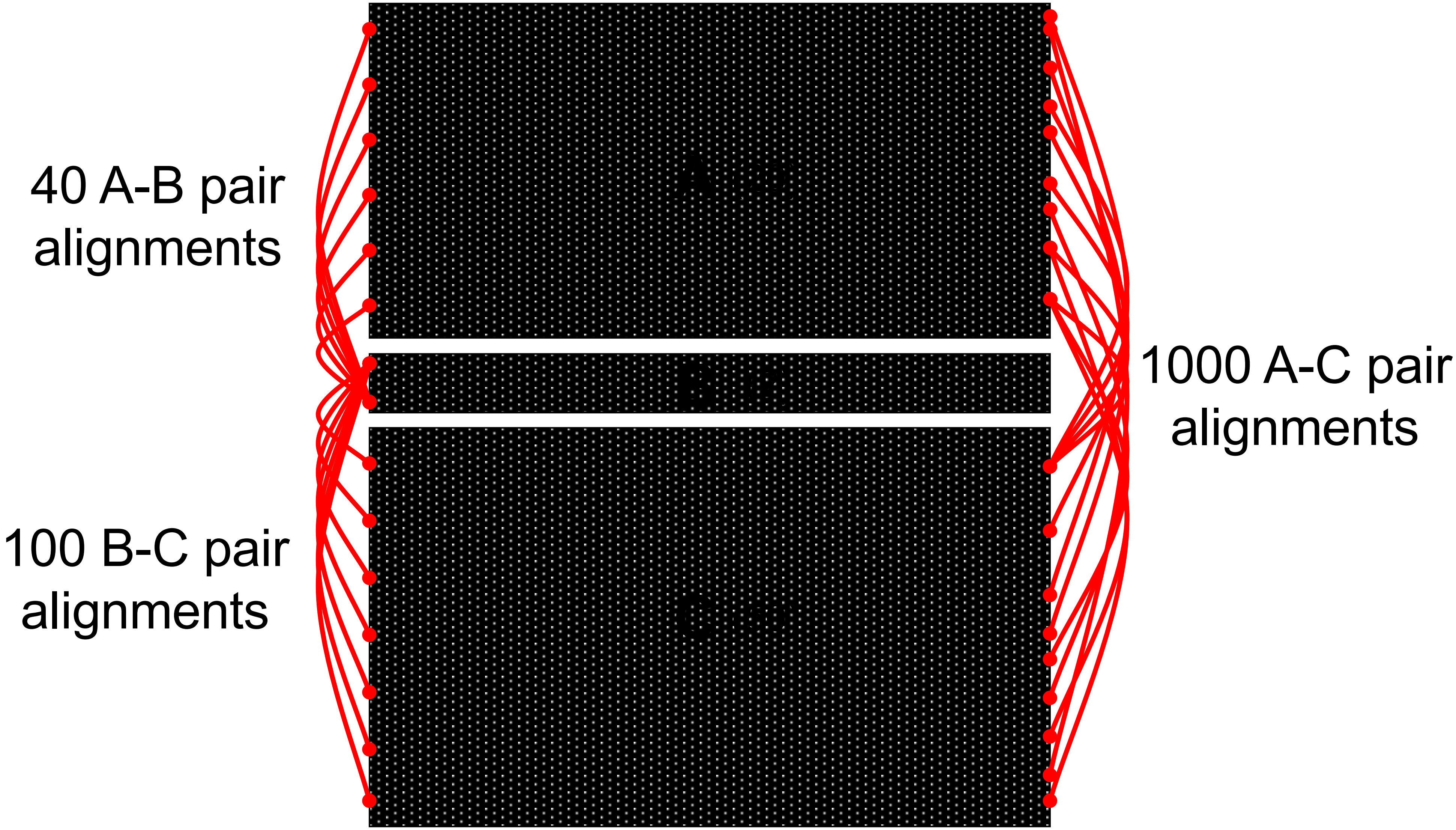
Form-and-polish review

1. Choosing parameters
2. Constructing the merge tree
    a. Grouping sequences
    b. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Weighting sequence pairs



Weighting sequence pairs



Weighting methods

Covariance weights [Altschul, et al. 1989]

- Based on correlation between paths
- Approximated in practice [Gotoh 1995]
- Used in MAFFT

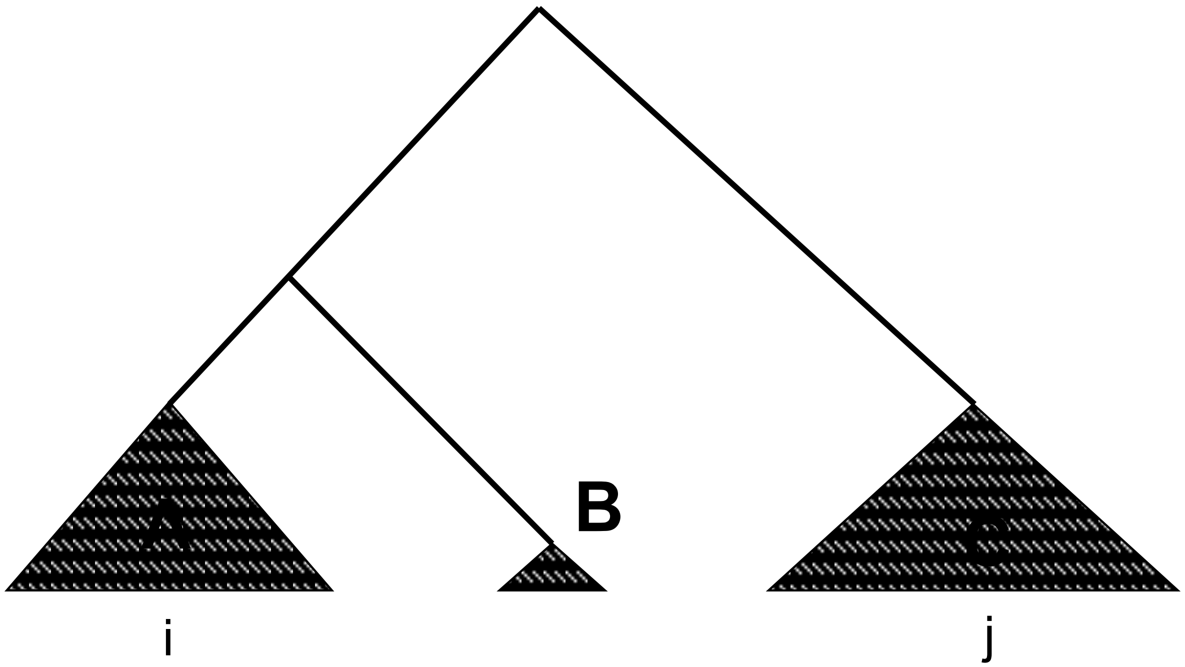
Division weights [Thompson, et al. 1994]

- Edge lengths divided among leaves
- Used in ClustalW, Muscle

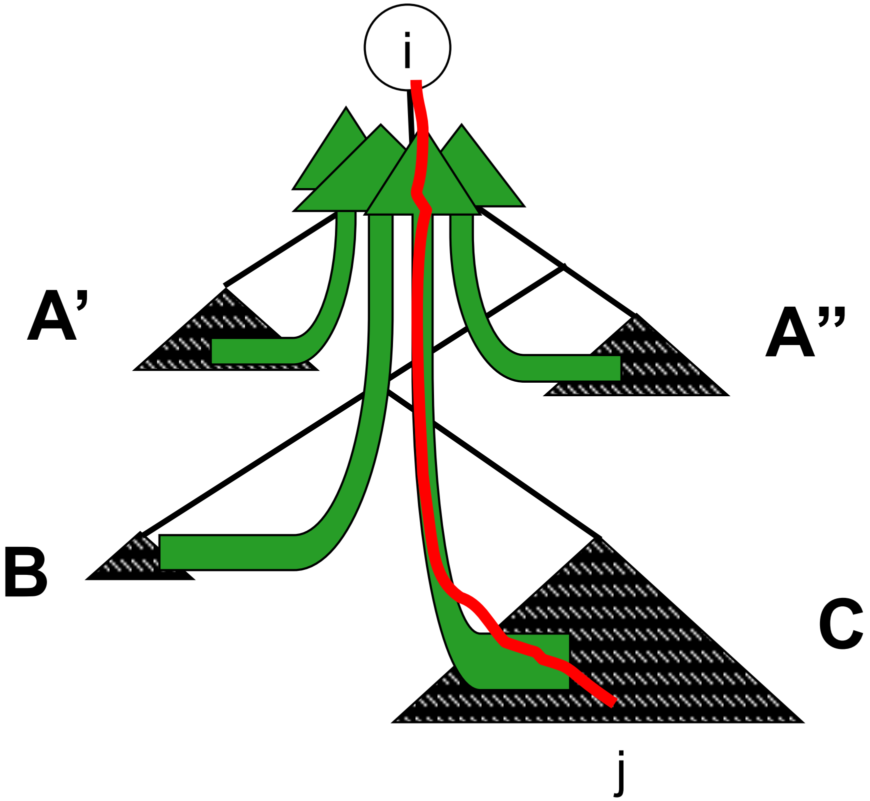
Influence weights

- Based on the influence of leaf j on i, $\omega_{i,j}$

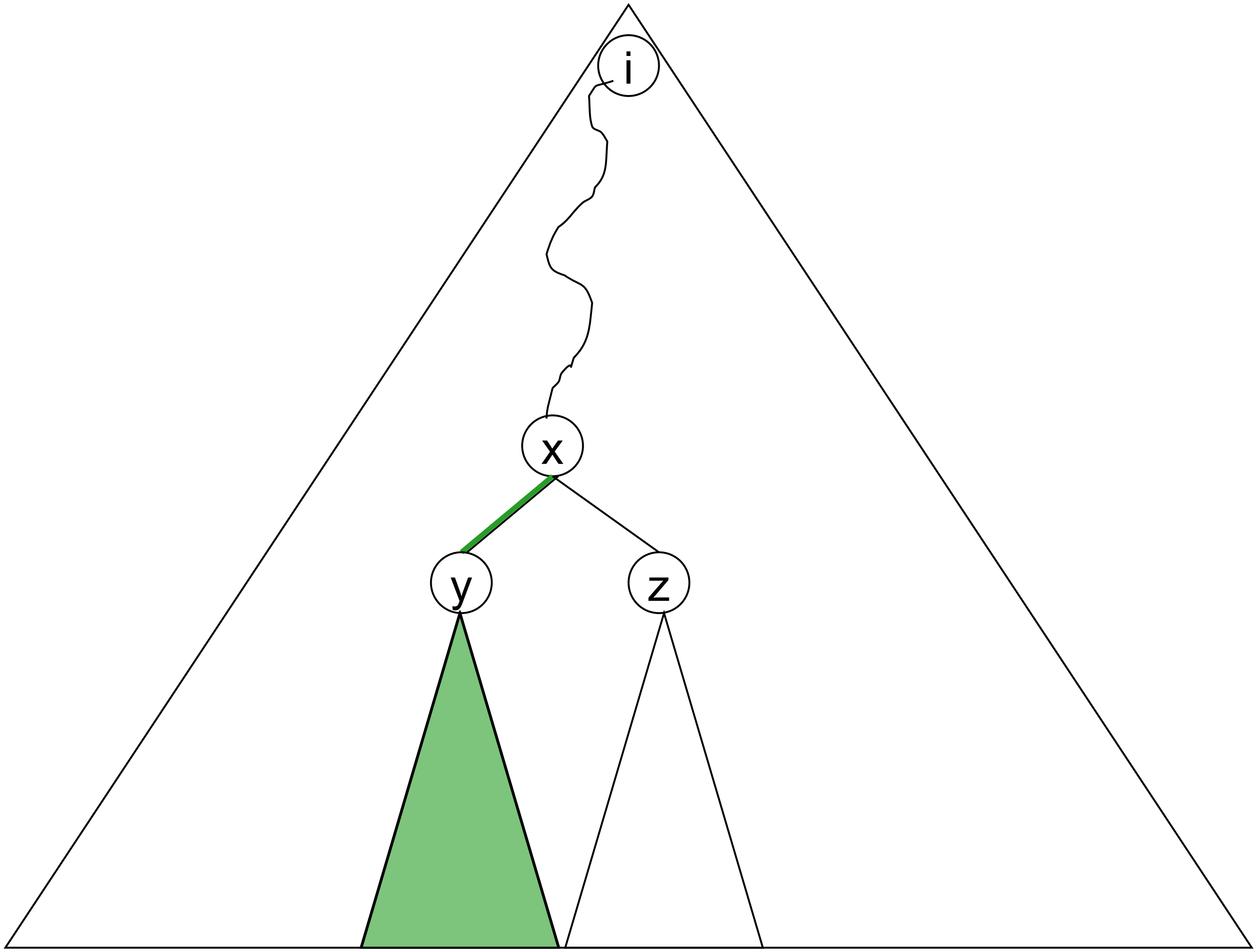
Influence weights



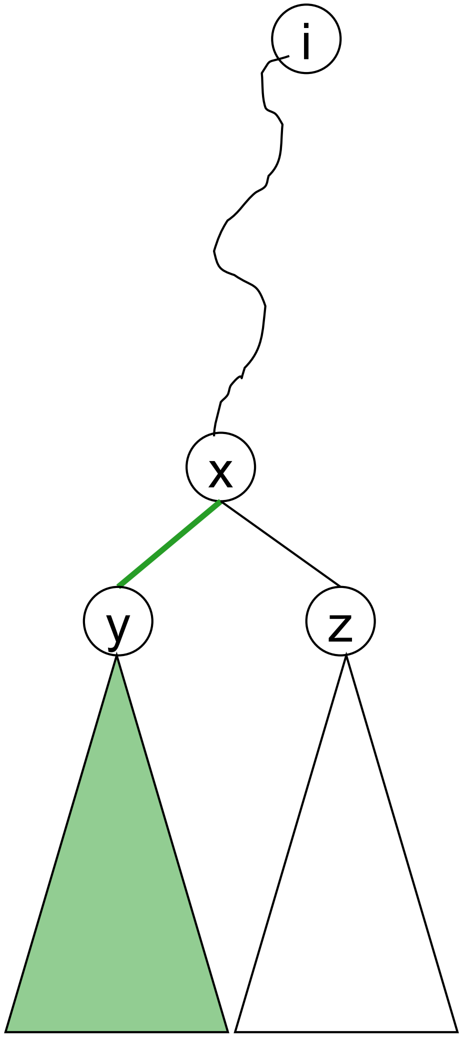
Influence weights



Influence weights



Influence weights



$T(y)$: tree under y (subtree)

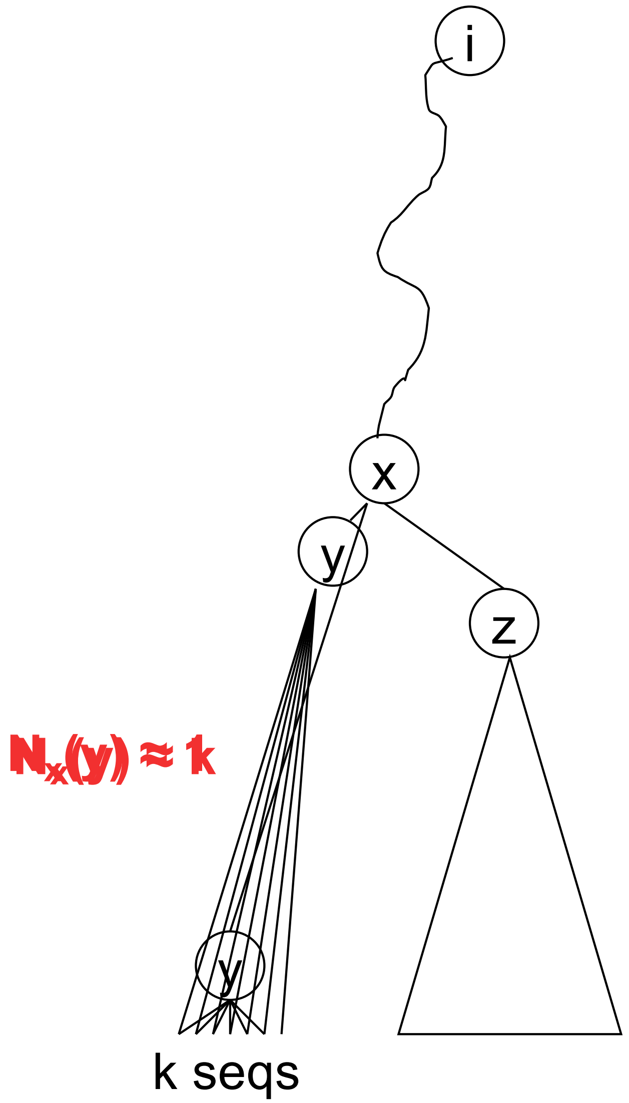
$S_x(y) : \ell(x,y) + \sum_{e \in T(y)} (\ell(e))$ (size)

$L(y)$: set of leaves under y (leaf set)

$H_x(y)$: avg path length from x to $L(y)$ (height)

$$N_x(y) = \frac{S_x(y)}{H_x(y)}$$ (effective # sequences)

Influence weights



T(y) : tree under y (subtree)

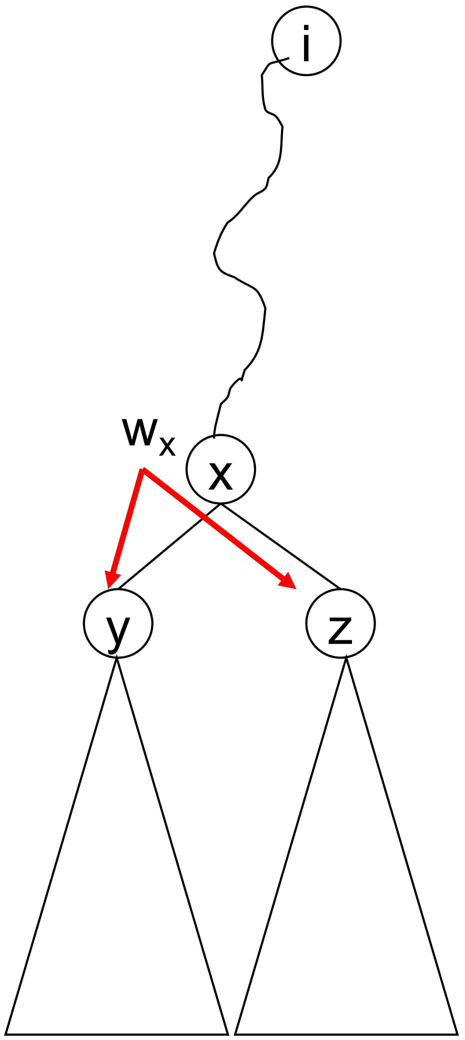
$S_x(y) : \ell(x,y) + \sum_{e \in T(y)} (\ell(e))$ (size)

L(y) : set of leaves under y (leaf set)

$H_x(y)$: avg path length from x to L(y) (height)

$$N_x(y) = \frac{S_x(y)}{H_x(y)}$$ (effective # sequences)

Influence weights



T(y) : tree under y (subtree)

$S_x(y) : \ell(x,y) + \sum_{e \in T(y)} (\ell(e))$ (size)

L(y) : set of leaves under y (leaf set)

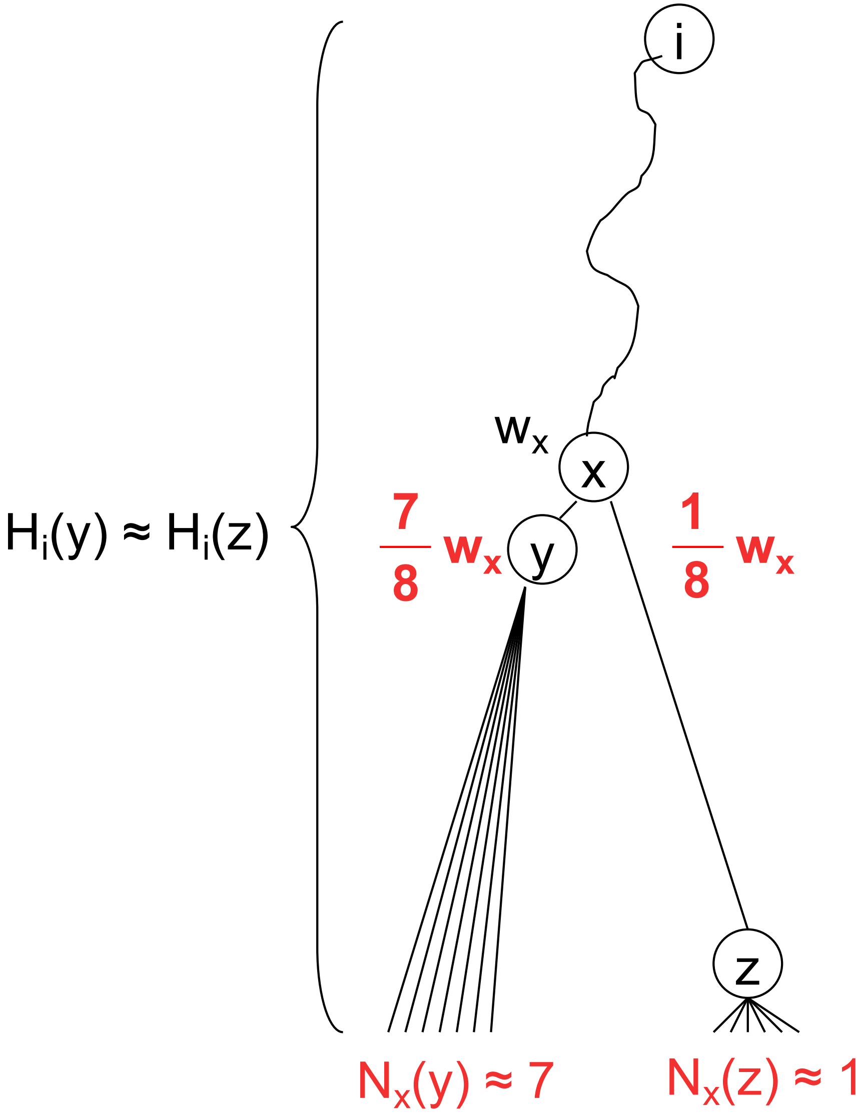
$H_x(y)$: avg path length from x to L(y) (height)

$$N_x(y) = \frac{S_x(y)}{H_x(y)}$$ (effective # sequences)

Split $w_x = w_y + w_z$ according to the ratio:

$$\frac{w_y}{w_z} = \frac{N_x(y)}{N_x(z)} \frac{H_i(z)}{H_i(y)}$$

Influence weights



$T(y)$: tree under y (subtree)

$S_x(y) : \ell(x,y) + \sum_{e \in T(y)} (\ell(e))$ (size)

$L(y)$: set of leaves under y (leaf set)

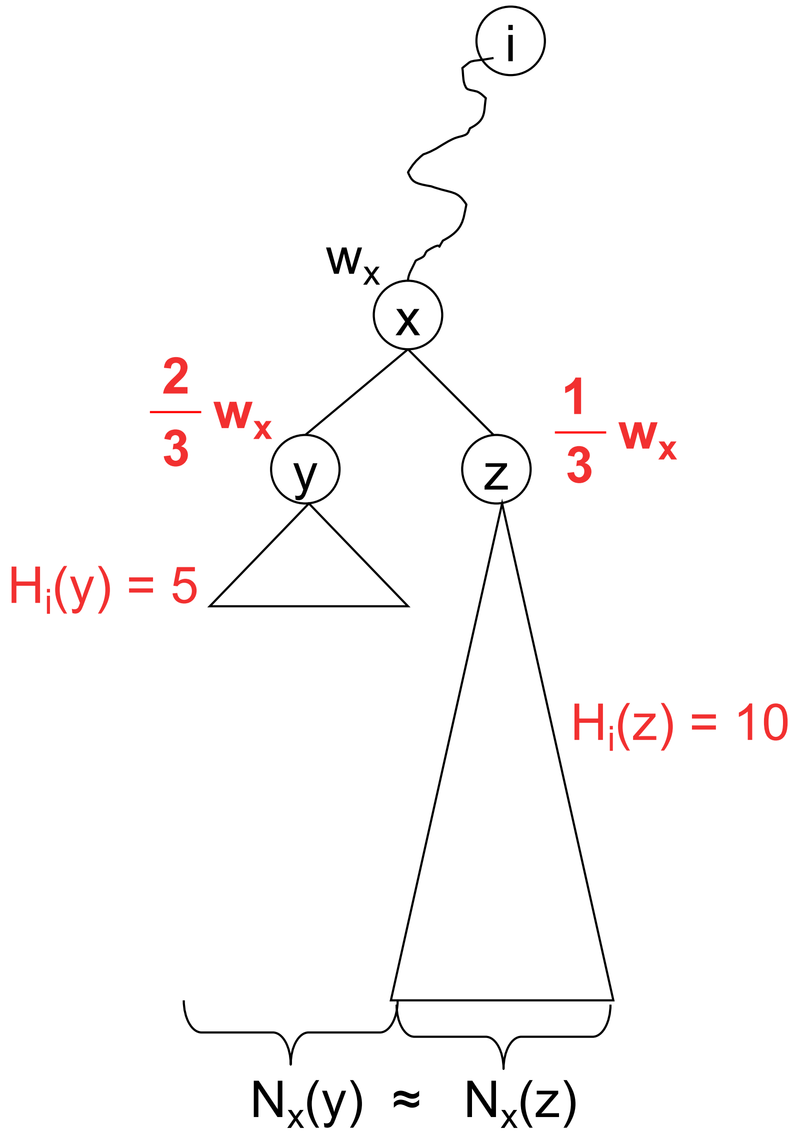
$H_x(y)$: avg path length from x to $L(y)$ (height)

$$N_x(y) = \frac{S_x(y)}{H_x(y)}$$ (effective # sequences)

Split $w_x = w_y + w_z$ according to the ratio:

$$\frac{w_y}{w_z} = \frac{N_x(y)}{N_x(z)} \quad \frac{H_i(z)}{H_i(y)}$$

Influence weights



$T(y)$: tree under y (subtree)

$S_x(y) : \ell(x,y) + \sum_{e \in T(y)} (\ell(e))$ (size)

$L(y)$: set of leaves under y (leaf set)

$H_x(y)$: avg path length from x to $L(y)$ (height)

$$N_x(y) = \frac{S_x(y)}{H_x(y)}$$ (effective # sequences)

Split $w_x = w_y + w_z$ according to the ratio:

$$\frac{w_y}{w_z} = \frac{N_x(y)}{N_x(z)} \frac{H_i(z)}{H_i(y)}$$

Influence weights

- Influence $\omega(i,j)$ is the weight w_j
- Not symmetric: $\omega(i,j) \neq \omega(j,i)$
- Define $w_{ij} = \sqrt{\omega(i,j)\, \omega(j,i)}$

$$\text{SP score} = \sum_{i,j} w_{i,j} \; \text{score}(\, i, j \,)$$

Comparing weighting methods

Weighting method	Average		BAliBase references 2 & 3
	SPS	TC	SPS
Influence	**71.6**	**55.5**	**83.3**
Uniform	**71.6**	**55.5**	-0.5
Division	**71.6**	-0.1	-0.8
Covariance	-0.3	-0.1	-1.8

- Weights have little impact for these suites

Form-and-polish review

1. Choosing parameters
2. Constructing the merge tree
    1. Grouping sequences
    2. Measuring distances
3. Weighting sequence pairs
4. Merging alignments
5. Polishing the alignment

Choosing parameters

Default parameter selection:

- Seed value by inverse alignment
  - InverseAlign [Kececioglu, Kim 2006] on BAliBase
  - Substitution matrix fixed at BLOSUM62
- Evaluated 800 parameter choices near seed

Default can be poor on some sequences:

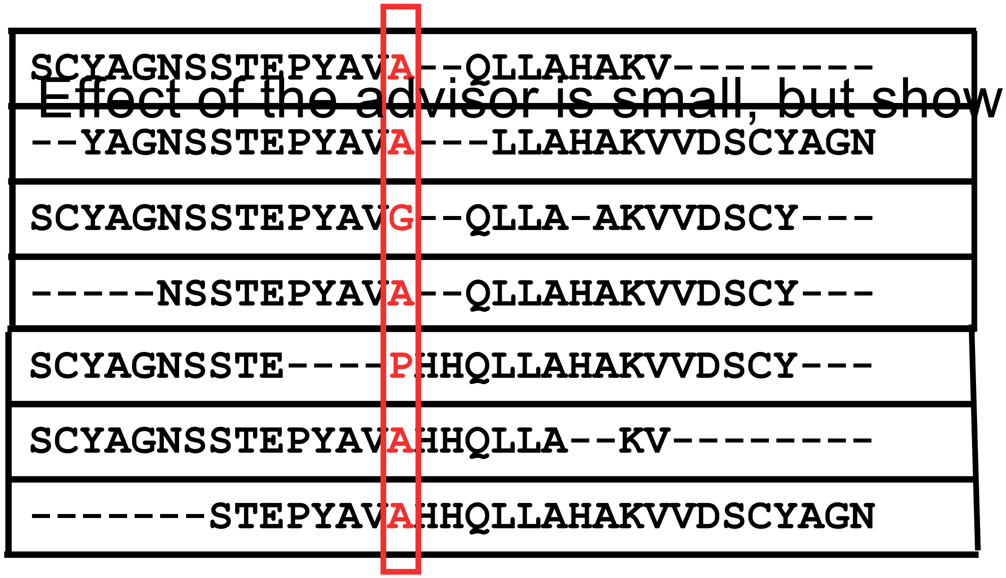
- SABmark superfamily group 287:

Default parameters:	20%
Best parameters:	75%

Choosing parameters

Parameter choice	BAliBase	SABmark	PALI	Average
Default	84.3	50.2	84.6	73.1
Oracle (12 options)	+2.7	+4.2	+2.5	+3.0
Oracle (4 options)	+1.9	+2.7	+1.6	+2.0
Advisor (4 options)	+0.4	+0.3	+0.3	**+0.3**

- Effect of the advisor is small, but shows significant potential



```
SCYAGNSSTEPYAVA--QLLAHAKV--------
--YAGNSSTEPYAVA---LLAHAKVVDSCYAGN
SCYAGNSSTEPYAVG--QLLA-AKVVDSCY---
-----NSSTEPYAVA--QLLAHAKVVDSCY---
SCYAGNSSTE----PHHQLLAHAKVVDSCY---
SCYAGNSSTEPYAVAHHQLLA--KV--------
-------STEPYAVAHHQLLAHAKVVDSCYAGN
```

Core column: >90% identity
(compressed alphabet)

Impact of methods

Stage	Average	Best Method
(Baseline)	**67.1**	
Tree	+0.7	MST
Distance	**+3.1**	Normalized cost
Merge	+0.7	Exact counts
Polish	+1.5	3-cut
Parameters	+0.3	Advisor
(Combined)	**73.4**	

Opal!

Comparing to other tools

Tool	Average			
	SPS	TC		**Consistency**
MAFFT	72.9	**60.4**	←	5% gain
Probcons	73.1	59.0	←	
Opal with advisor	**73.4**	58.7		
Opal, default parameters	73.1	58.4		
T-Coffee	69.4	54.7	←	**Hydrophobicity**
Muscle	69.0	53.8	←	4% gain
Opal baseline	67.1	49.1		
ClustalW	63.9	43.0	←	

Conclusion

- Best-of-breed methods identified
- Opal achieves state-of-the-art accuracy
  - Does not use consistency or hydrophobicity
- Greatest gains from:
  - normalized alignment cost for distances
  - 3-cut for polishing

Future work

- Incorporate hydrophobicity in aligning alignments

- Design unbiased recovery measures for alignments with overrepresented groups

- Investigate parameter advisor methods

Acknowledgements

- Eagu Kim
- David Maddison
- Marcy McClure
- Dean Starrett

- Research supported in part by
  - NSF IGERT in Genomics Fellowship
  - NSF Grant DBI-0317498

- Travel fellowship from
  - US National Science Foundation