Comparison of sequence alignment methods

Yu Gu

December 14, 2015
Department of Biostatistics and Computational Biology
School of Medicine and Dentist

University of Rochester

Outline

Outline

Why sequence alignment is needed?

Sequence alignment plays an important role in bioinformatics, which could be the fundamental tasks in bioinformatics. The purpose is to align sequences in such a way to reflect the biological relationship between the input sequences. Through arranging the sequences of DNA, RNA or protein, scientists expect to identify similar regions between these sequences, which might indicate functional, structual or evolutionary relationships.

Application of sequence alignment

Application of sequence alignment

- Detection of homologous regions

Application of sequence alignment

- Detection of homologous regions
- Detecting motifs and conserved regions within related families of proteins that then may be inferred to play a role in structure and function

Application of sequence alignment

- Detection of homologous regions
- Detecting motifs and conserved regions within related families of proteins that then may be inferred to play a role in structure and function
- Structure prediction, i.e., detecting structural building blocks, in which the role of a residue in secondary or tertiary structure is inferred.

Application of sequence alignment

- Detection of homologous regions
- Detecting motifs and conserved regions within related families of proteins that then may be inferred to play a role in structure and function
- Structure prediction, i.e., detecting structural building blocks, in which the role of a residue in secondary or tertiary structure is inferred.
- Constructing sequence profiles

Application of sequence alignment

- Detection of homologous regions
- Detecting motifs and conserved regions within related families of proteins that then may be inferred to play a role in structure and function
- Structure prediction, i.e., detecting structural building blocks, in which the role of a residue in secondary or tertiary structure is inferred.
- Constructing sequence profiles
- An important prerequisite for the construction of phylogenetic trees, which could be a predictor of evolutionary relationships

[Bodenhofer et al., 2015]

Basic Idea of Sequence Alignment

- Given N sequences and a scoring scheme for determining the best matches of the letters, find the optimal pairing of letters between the sequences.
- The scoring scheme used by pairwise sequence alignment (PSA) defines the best alignment of two sequences as the sum of score matrix for each pair of letters minus gap penalties.
- Multiple sequence alignment (MSA) is an extension of PSA. The most widely-used algorithms are progressive alignments, which aligning a set of N sequences by performing N-1 pairwise alignments to create a distance matrix.

Outline

Classification of PSA

Current computational alignment methods can be categorized into two groups: global alignment and local alignment.

Classification of PSA

Current computational alignment methods can be categorized into two groups: global alignment and local alignment.

- Global alignment methods fulfill global optimization, which spans the entire length of the input sequences, i.e., aligning the every residue of the sequences.

Classification of PSA

Current computational alignment methods can be categorized into two groups: global alignment and local alignment.

- Global alignment methods fulfill global optimization, which spans the entire length of the input sequences, i.e., aligning the every residue of the sequences.
- Local alignment identifies similar region within a long sequence, i.e., regions of aligned sub-sequences might be surrounded by sequences that are completely unrelated.

Dynamic Programming

Dynamic programming is widely applied to the sequence alignment problem. However, its high computational costs in time and memory might cause problem when aligning extremely long sequences or large number of sequences.

Dynamic Programming

Dynamic programming is widely applied to the sequence alignment problem. However, its high computational costs in time and memory might cause problem when aligning extremely long sequences or large number of sequences.

Main Algorithms

- the Needleman-Wunsch algorithm is applied to produce global alignment
- the Smith-Waterman algorithm is applied to produce local alignment

Dynamic Programming Cont.

Dynamic Programming Cont.

Scoring Scheme

- Protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other.
- DNA and RNA alignments use a scoring matrix to paired nucleic acids, such as assigning a positive match score, a negative mismatch score and a negative gap penalty.

Dynamic Programming Cont.

Scoring Scheme

- Protein alignments use a substitution matrix to assign scores to amino-acid matches or mismatches, and a gap penalty for matching an amino acid in one sequence to a gap in the other.
- DNA and RNA alignments use a scoring matrix to paired nucleic acids, such as assigning a positive match score, a negative mismatch score and a negative gap penalty.

Gap Penalty

The values assigned to gap openings and gap extensions, which could be either same or different. The values can also be either fixed or adjusted.

Needleman-Wunsch Algorithm

To fill the matrix F,

- Introduce gap penalty: the top row and left column are filled with $i \times d, i = 1, ...m$ and $j \times d, j = 1, ..., n$, where m and n are the length of two sequences respectively.
- $F(0, 0) = 0$ is the start position, and $F(n, m)$ is the start position of tracking back.
- For each cell, do the following calculation to get the score value, where $S(X_i, Y_i)$ is the pair-score obtained from score matrix,

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + S(X_i, Y_i) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

- Track back from the right-bottom position and end with $F(0, 0)$ to find the best path.

[Durbin et al., 1998]

Smith-Waterman Algorithm

To fill the score matrix F,

- Begin with filling zeros in the top row and left column.
- For each cell, do the following calculation to get the score value

$$F(i,j) = \max \begin{cases} 0 \\ F(i-1,j-1) + S(X_i, Y_i) \\ F(i-1,j) - d \\ F(i,j-1) - d \end{cases}$$

- Once $F(i,j) < 0$, assign zero to that cell, which allows us to restart alignment at this position
- Find the cell with highest score value and track back from this position until the first time touch the cell with score of zero to find the optimal path

[Durbin et al., 1998]

Implementation of Pairwise Algorithm

Implementation of Pairwise Algorithm

User input:

- Input data : FASTA file, two GeneBank identifiers to retrieve protein sequences, or two protein sequences
- Substitution matrix file name
- Gap-opening and gap-extension penalties

Implementation of Pairwise Algorithm

User input:

- Input data : FASTA file, two GeneBank identifiers to retrieve protein sequences, or two protein sequences
- Substitution matrix file name
- Gap-opening and gap-extension penalties

Output: a text file

- Alignment score
- Aligned sequences with gaps, could be the entire length of sequences for global alignment or sub-sequences for local alignment.

Comparison of NW and SW algorithm

Comparison of NW and SW algorithm

- The aligned sequence of NW global algorithm would start from at least one of the input sequences' first letter and end with at least one of the last letter, while the aligned sequence of SW local algorithm would be the sub-sequence of input sequences

Comparison of NW and SW algorithm

- The aligned sequence of NW global algorithm would start from at least one of the input sequences' first letter and end with at least one of the last letter, while the aligned sequence of SW local algorithm would be the sub-sequence of input sequences
- Both of the two algorithm might have more than one optimal path, however, NW global algorithm would always have the same boundary pairs, while SW local algorithm might start from and end with different positions, because the track back of local algorithm would start from the position with highest score, which might have multiple ones.

Comparison of NW and SW algorithm

- The aligned sequence of NW global algorithm would start from at least one of the input sequences' first letter and end with at least one of the last letter, while the aligned sequence of SW local algorithm would be the sub-sequence of input sequences
- Both of the two algorithm might have more than one optimal path, however, NW global algorithm would always have the same boundary pairs, while SW local algorithm might start from and end with different positions, because the track back of local algorithm would start from the position with highest score, which might have multiple ones.
- The locally aligned sequences will always start with a "match"

Example of NW and SW algorithm

Using BLOSUM50 and using equally gap penalty "8" for gap opening and gap extension
sequence 1:"HEAGAWGHEE" and sequence 2:"PAWHEAE"

```
> output
$path
        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
X.align "*"  "H"  "E"  "A"  "G"  "A"  "W"  "G"
Y.align "*"  "_"  "_"  "P"  "-"  "A"  "W"  "-"
        [,9] [,10] [,11] [,12] [,13]
X.align "H"  "E"   "_"   "E"   "*"
Y.align "H"  "E"   "A"   "E"   "*"

$fMatrix
        H   E   A   G   A   W   G   H   E   E
    0  -8 -16 -24 -32 -40 -48 -56 -64 -72 -80
P  -8  -2  -9 -17 -25 -33 -41 -49 -58 -65 -73
A -16 -10  -3  -4 -12 -20 -36 -41 -49 -57 -66
W -24 -18 -11  -6  -7 -15  -5 -13 -21 -37 -60
H -32 -14 -18 -13  -8  -9 -13  -7  -3 -11 -19
E -40 -22  -8 -16 -16  -9 -12 -15  -7   3  -5
A -48 -30 -16  -3 -11 -11 -12 -12 -15  -5   2
E -56 -46 -24 -11  -6 -12 -14 -15 -12  -9   1
```

```
$path.list
$path.list[[1]]
        [,1] [,2] [,3] [,4] [,5]
X.align "A"  "W"  "G"  "H"  "E"
Y.align "A"  "W"  "-"  "H"  "E"


$bestScore
[1] 28

$fMatrix
     H  E  A  G  A  W  G  H  E  E
  0  0  0  0  0  0  0  0  0  0  0
P 0  0  0  0  0  0  0  0  0  0  0
A 0  0  0  5  0  5  0  0  0  0  0
W 0  0  0  0  2  0 20 12  4  0  0
H 0 10  2  0  0  0 12 18 22 14  6
E 0  2 16  8  0  0  4 10 18 28 20
A 0  0  8 21 13  5  0  4 10 20 27
E 0  0  6 13 18 12  4  0  4 16 26
```

Outline

Clustal W

The "W" in the name stands for "weighting", which indicates assigning different weights to sequences and parameters at different positions in alignment.

Clustal W

The "W" in the name stands for "weighting", which indicates assigning different weights to sequences and parameters at different positions in alignment.

Basic multiple alignment consists of three main stages:

Clustal W

The "W" in the name stands for "weighting", which indicates assigning different weights to sequences and parameters at different positions in alignment.

Basic multiple alignment consists of three main stages:

- Calculate all pairwise sequence similarity to construct a distance matrix, giving the divergence of each of sequences via full dynamic programming
- Construct a guide tree from the distance matrix via neighbor-joining algorithm and derives sequence weights
- Do progressive alignment [Hogeweg and Hesper, 1984] according to the branching order in the guide tree

[Thompson et al., 1994]

Clustal W Highlights

Clustal W Highlights

- Apply neighbor-joining method for making initial guide tree, which provides more reliable tree topology and gives better estimates of tree branch length that used to weight sequences and adjust the alignment parameters dynamically

Clustal W Highlights

- Apply neighbor-joining method for making initial guide tree, which provides more reliable tree topology and gives better estimates of tree branch length that used to weight sequences and adjust the alignment parameters dynamically
- Assign individual weights to each sequence in a partial alignment to down-weight near-duplicate sequences and up-weight the most divergent ones

Clustal W Highlights

- Apply neighbor-joining method for making initial guide tree, which provides more reliable tree topology and gives better estimates of tree branch length that used to weight sequences and adjust the alignment parameters dynamically
- Assign individual weights to each sequence in a partial alignment to down-weight near-duplicate sequences and up-weight the most divergent ones
- Do dynamic calculation of sequence- and position-specific gap penalties as the alignment proceeds

[Thompson et al., 1994, Higgins et al., 1996]

MUSCLE

- MUSCLE (multiple sequence comparison by log-expectation) was introduced in 2004
- Matrix-based algorithm, start with a guide tree construction, the essential step is pairwise alignment
- Two distinguish features:
  - Using both the k-mer distance for an unaligned pair and the Kimura distance for an aligned pair to calculate all pairwise sequence distance
  - At the completion of any stage of the algorithm, a MSA is available and the algorithm can be terminated

MUSCLE Main Stages

MUSCLE Main Stages

Stage 1: draft progressive

calculate k-mer distance matrix through k-mer counting and then construct a rooted guide tree via UPGMA algorithm, and finally, do progressive alignment to obtain the first MSA

MUSCLE Main Stages

Stage 1: draft progressive

calculate k-mer distance matrix through k-mer counting and then construct a rooted guide tree via UPGMA algorithm, and finally, do progressive alignment to obtain the first MSA

Stage 2:improved progressive

compute Kimura distance matrix and re-estimating the guide tree via UPGMA allgorithm, and do progressive alignment to produce second MSA. Compare the guide tree from stage 1 and 2, identifying a set of nodes for which the branching order is different. Build new MSA if the order has changed, or keep the first MSA

MUSCLE Main Stages Cont.

Stage 3

refinement: delete an edge of the guide tree from stage 2, which divides the tree into two sub-trees and calculate the profile of multiple alignment for each sub-tree. Re-align the profiles from the two sub-tree to produce a new MSA. Keep the new MSA only if the new sum-of-pairs score is improved. Iterate this process until convergence or hitting the user-defined limit.

[Edgar, 2004]

Clustal Omega

- Completely rewritten and revised version of Clustal series of programs for MSA
- Retains the basic progressive alignment where the order of alignment is determined by a guide-tree
- Apply the mBed algorithm [Blackshields et al., 2010] for calculating guide trees, which allows it can deal with very large numbers of DNA/RNA or protein sequences
- Apply the HHalign method for aligning profile hidden Markov models, which considerably improves the accuracy

[Sievers and Higgins, 2014]

Clustal Omega Highlights

Clustal Omega Highlights

mBed algorithm:

It calculates the pairwise distance of all N sequences with respect to $logN$ randomly chosen seed sequences only, which reduces the time and memory complexity for guide tree calculation from $O(N^2)$ to $O(NlogN)$)

Clustal Omega Highlights

mBed algorithm:

It calculates the pairwise distance of all N sequences with respect to $logN$ randomly chosen seed sequences only, which reduces the time and memory complexity for guide tree calculation from $O(N^2)$ to $O(NlogN)$)

HHalign method:

HHalign is entirely based on Hidden-Markov Models(HMMs). Sequences and intermediary profiles are converted into HMMs, which are aligned in turn. There are two HMM alignment algorithm: Maximum Accuracy (MAC) algorithm and Viterbi algorithm. MAC is the default

[Sievers and Higgins, 2014, Sievers et al., 2011]

Outline

Simulation Data Generation

Simulation Data Generation

User Input

- lenth – lenth of sequences, can be scalar or vector
- sampleNum – number of sequences need to be generated
- type – specify the type of sequences
- dir – set up the working directory
- out.file – output file's name

Simulation Data Generation Cont.

Simulation Data Generation Cont.

Output

- generated sequences
- true alignment
- fasta file: if user input the file name, then it would be saved under the user-defined directory or current working directory

Compare Running Time

Data: sequence length is 500, number of sequences is 9

Clustal W

Parameters: gapOpening=10, gapExtension=0.2, substitution matrix=BLOSUM

$$t = 0.64, 0.66, 0.71, 0.66, 0.66, 0.64, 0.67, 0.67, 0.64, 0.72$$

Summary: min=0.64, mean=0.667, max=0.72

Compare Running Time Cont.

MUSCLE

Parameter: gapOpening=10, gapExtension=0.2, cluster=UPGMAMAX

$$t = 0.44, 0.46, 0.46, 0.46, 0.46, 0.44, 0.43, 0.46, 0.47, 0.46$$

Summary: min=0.43, mean=0.454, max=0.47

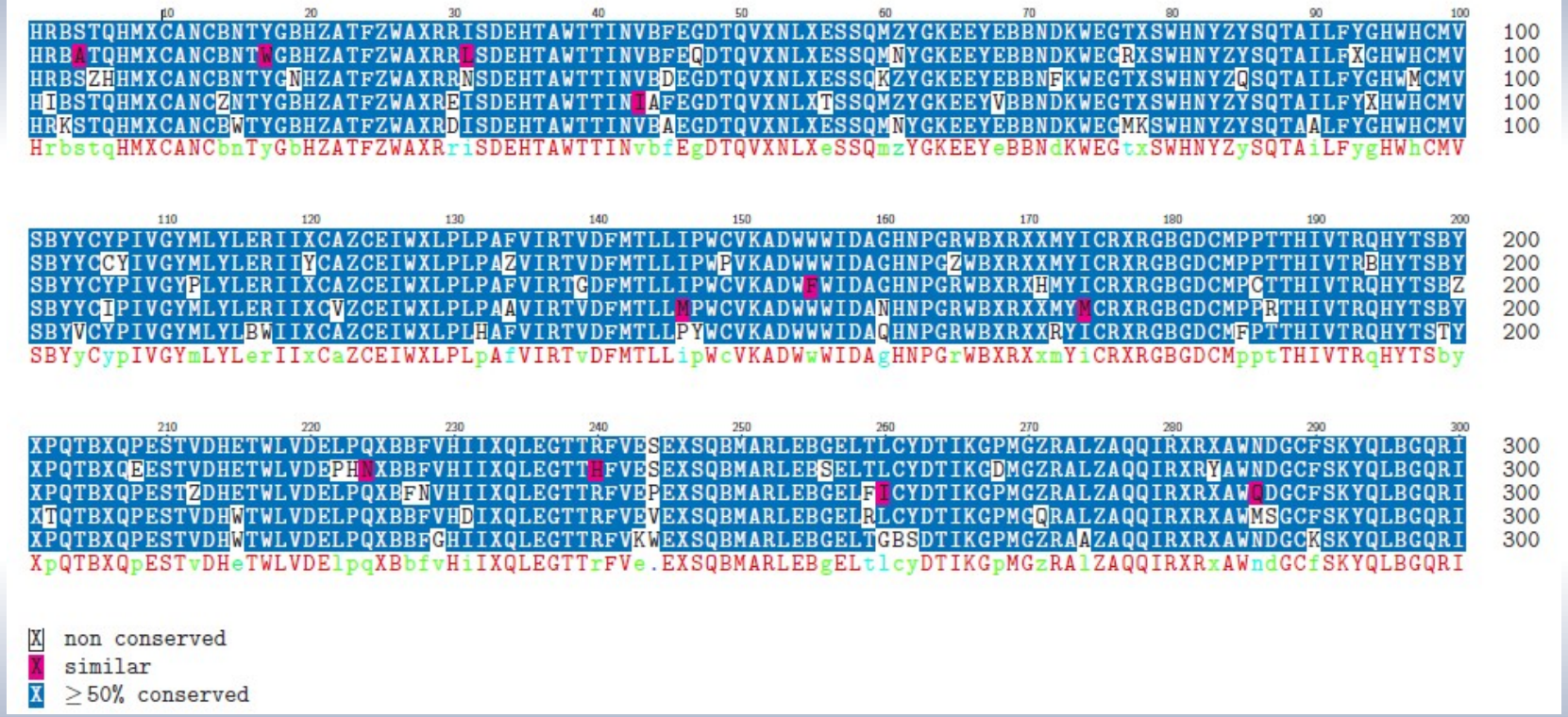
Clustal Omega

Parameter: cluster=mBed, cluster.size=100

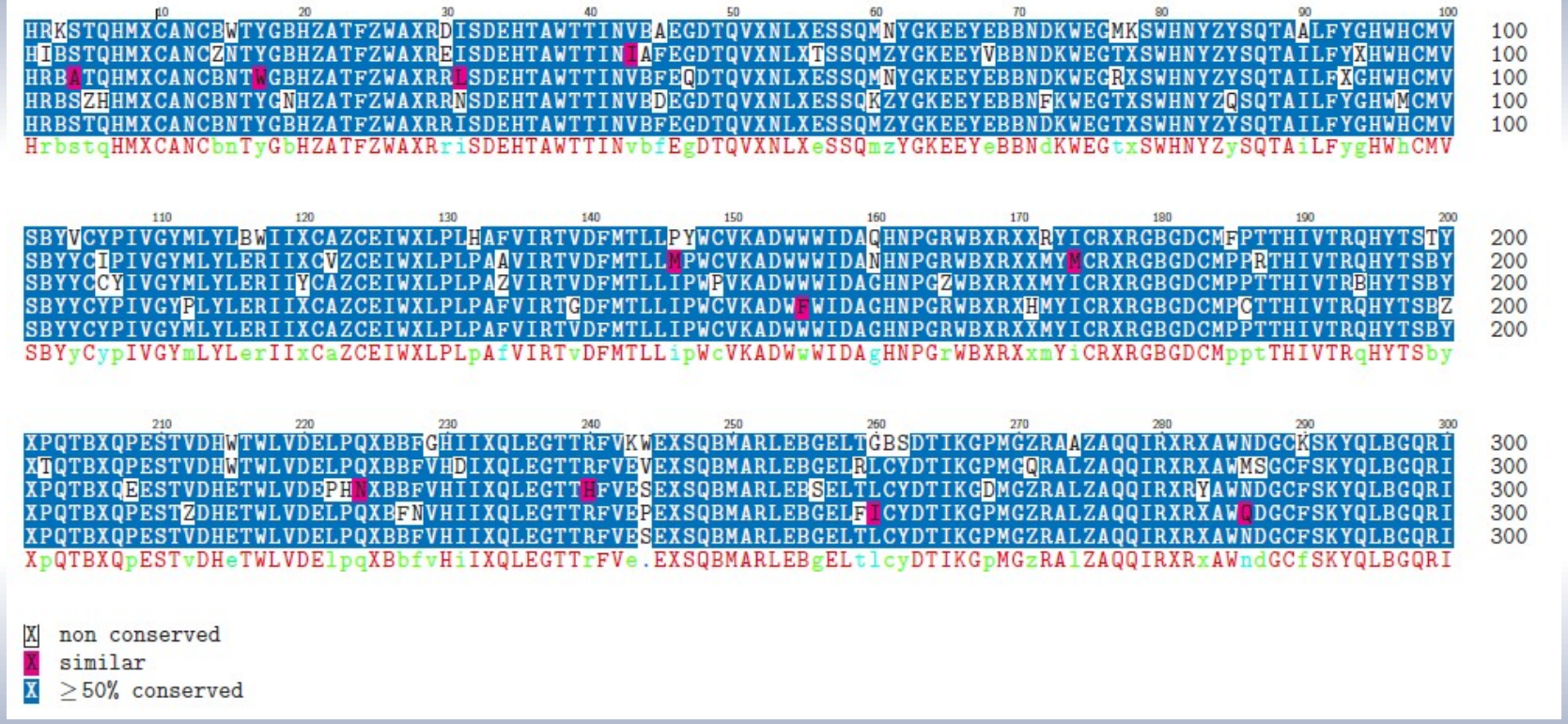
$$t = 0.56, 0.51, 0.5, 0.5, 0.51, 0.5, 0.57, 0.47, 0.5, 0.5$$

Summary: min=0.47, mean=0.512, max=0.57

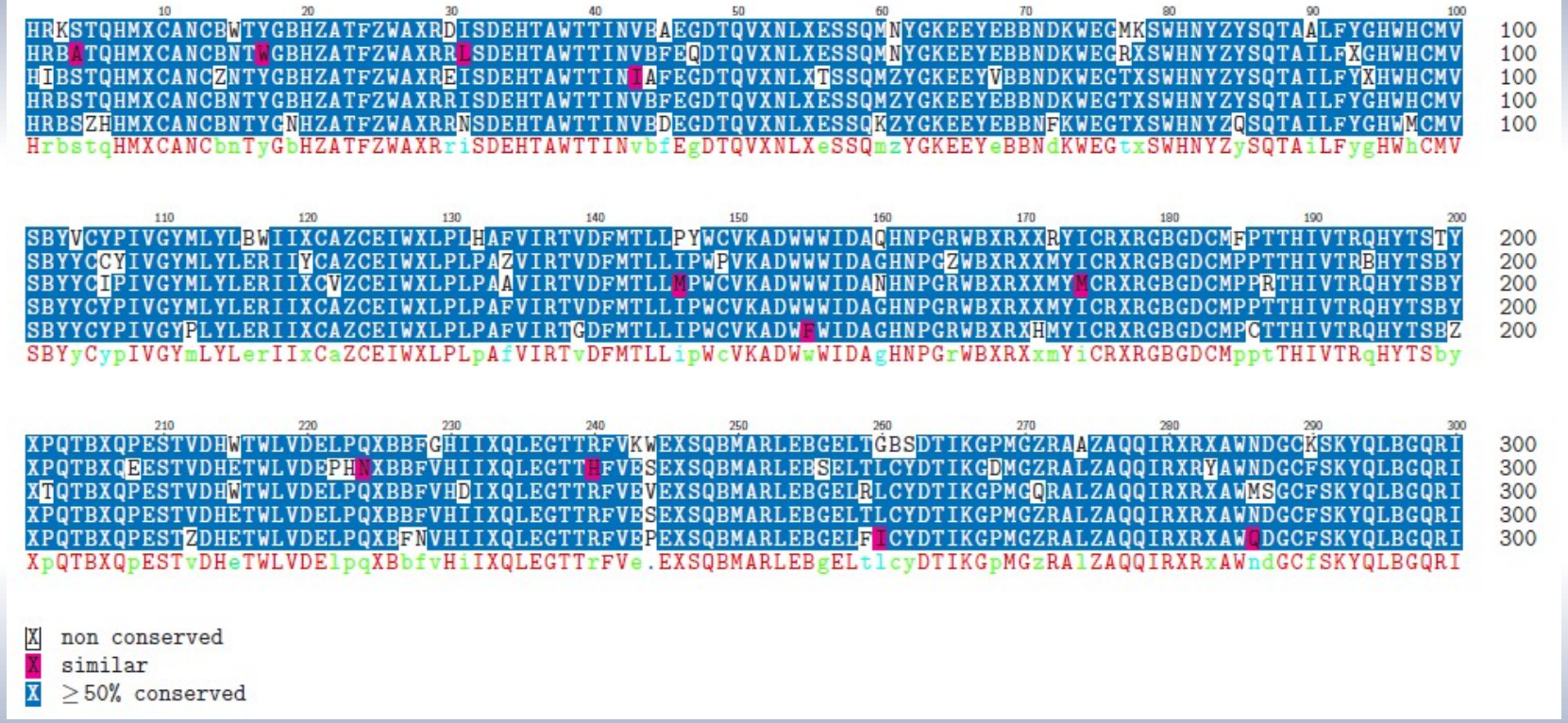
Compare Accuracy



Compare Accuracy Cont.



Compare Accuracy Cont.



Outline

References

Blackshields, G., Sievers, F., Shi, W., Wilm, A., and Higgins, D. G. (2010). Research sequence embedding for fast construction of guide trees for multiple sequence alignment. *Algorithm Mol Biol*, 5:21.

Bodenhofer, U., Bonatesta, E., Horejš-Kainrath, C., and Hochreiter, S. (2015). msa: an r package for multiple sequence alignment. *Bioinformatics*, page btv494.

Durbin, R., Eddy, S. R., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press.

Edgar, R. C. (2004). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792–1797.

Higgins, D. G., Thompson, J. D., and Gibson, T. J. (1996). [22] using clustal for multiple sequence alignments. *Methods in enzymology*, 266:383–402.

Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: an integrated method. *Journal of molecular evolution*, 20(2):175–186.

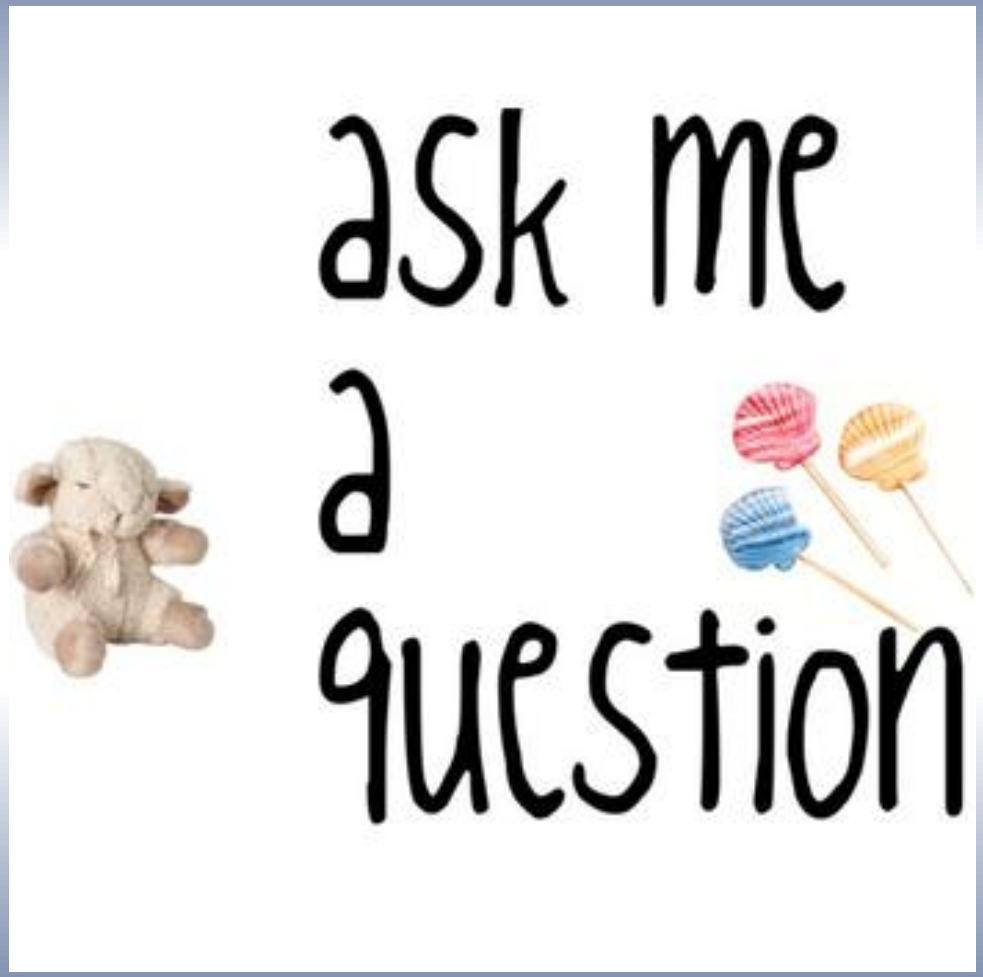
References (cont.)

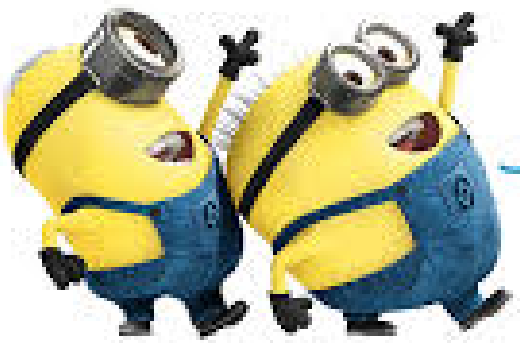
Sievers, F. and Higgins, D. G. (2014). Clustal omega, accurate alignment of very large numbers of sequences. pages 105–116.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1):539.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustal w: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, 22(22):4673–4680.

Questions





thank you!