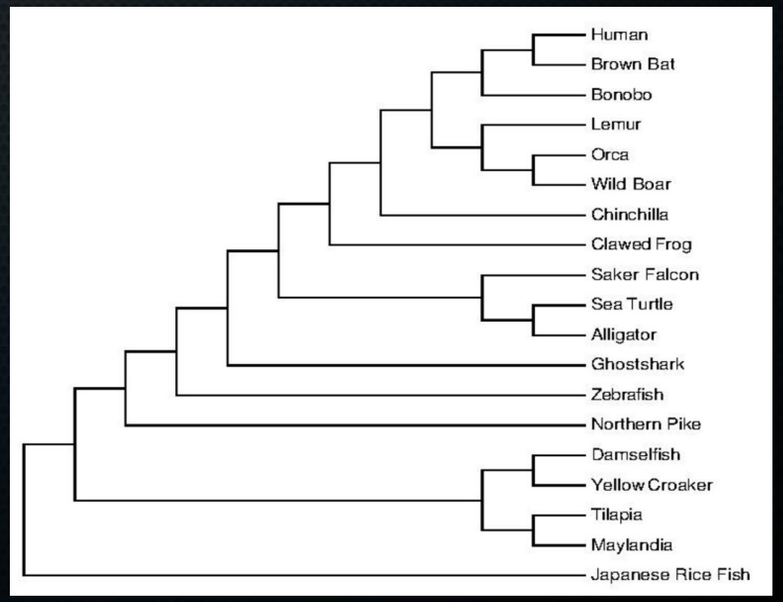
Clustering in practice: phylogeny

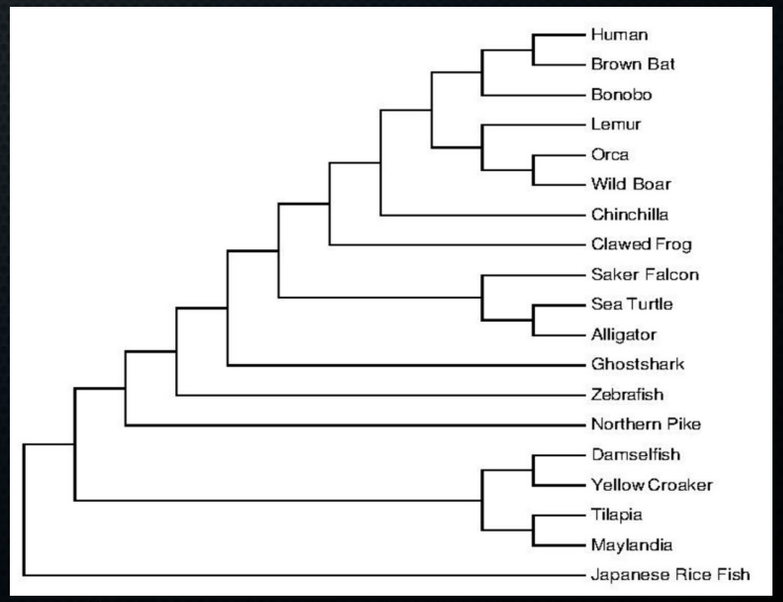
- Multiple-sequence alignment (MSA) is clustering of sequences according to similarity (and finding out which *parts* of the sequences are homologous/correspond to each other to assess this similarity), phylogenetic tree is made from differences.



Source: https://commons.wikimedia.org/wiki/File:Phylogenetic_Tree.pdf

Clustering in practice: phylogeny

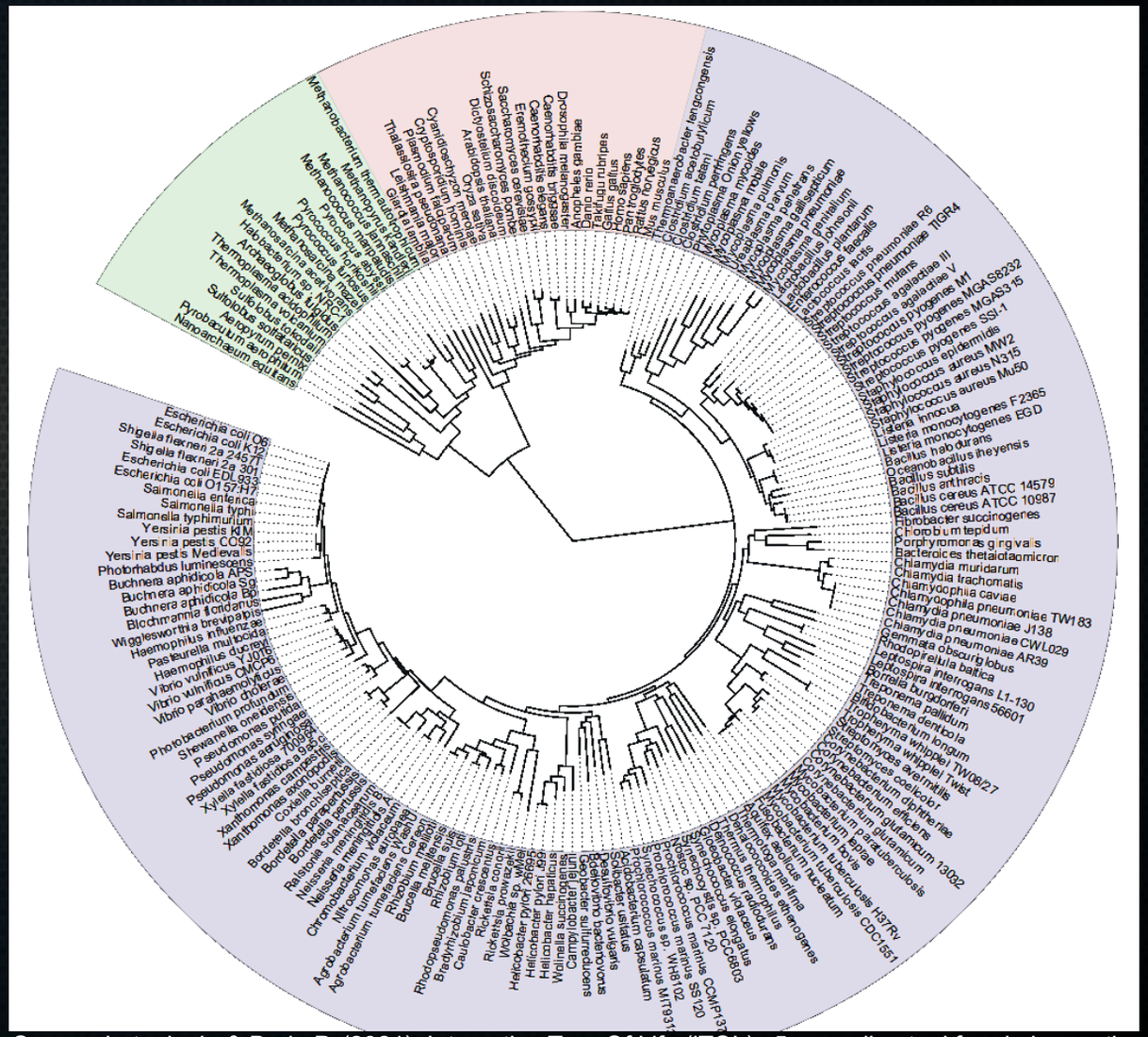
- Multiple-sequence alignment (MSA) is clustering of sequences according to similarity (and finding out which *parts* of the sequences are homologous/correspond to each other to assess this similarity), phylogenetic tree is made from differences.
- Look into how it works, and some problems or caveats due to clusters and MSAs not being *the correct* clusters/MSAs



Source: https://commons.wikimedia.org/wiki/File:Phylogenetic_Tree.pdf

Problems with phylogeny

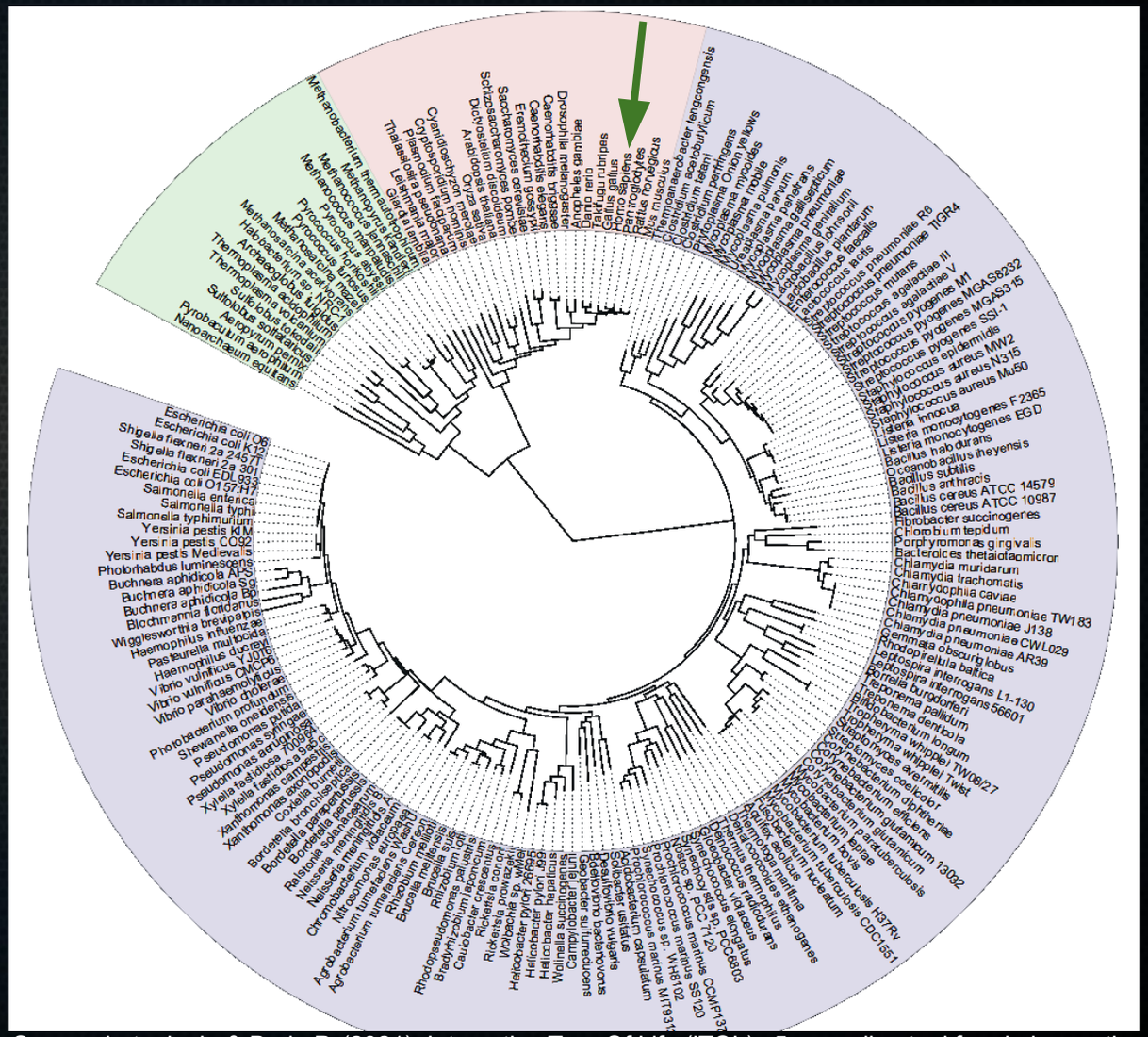
- What problems?



Source: Letunic, I., & Bork, P. (2021). Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic acids research, gkab301. Advance online publication. https://doi.org/10.1093/nar/gkab301

Problems with phylogeny

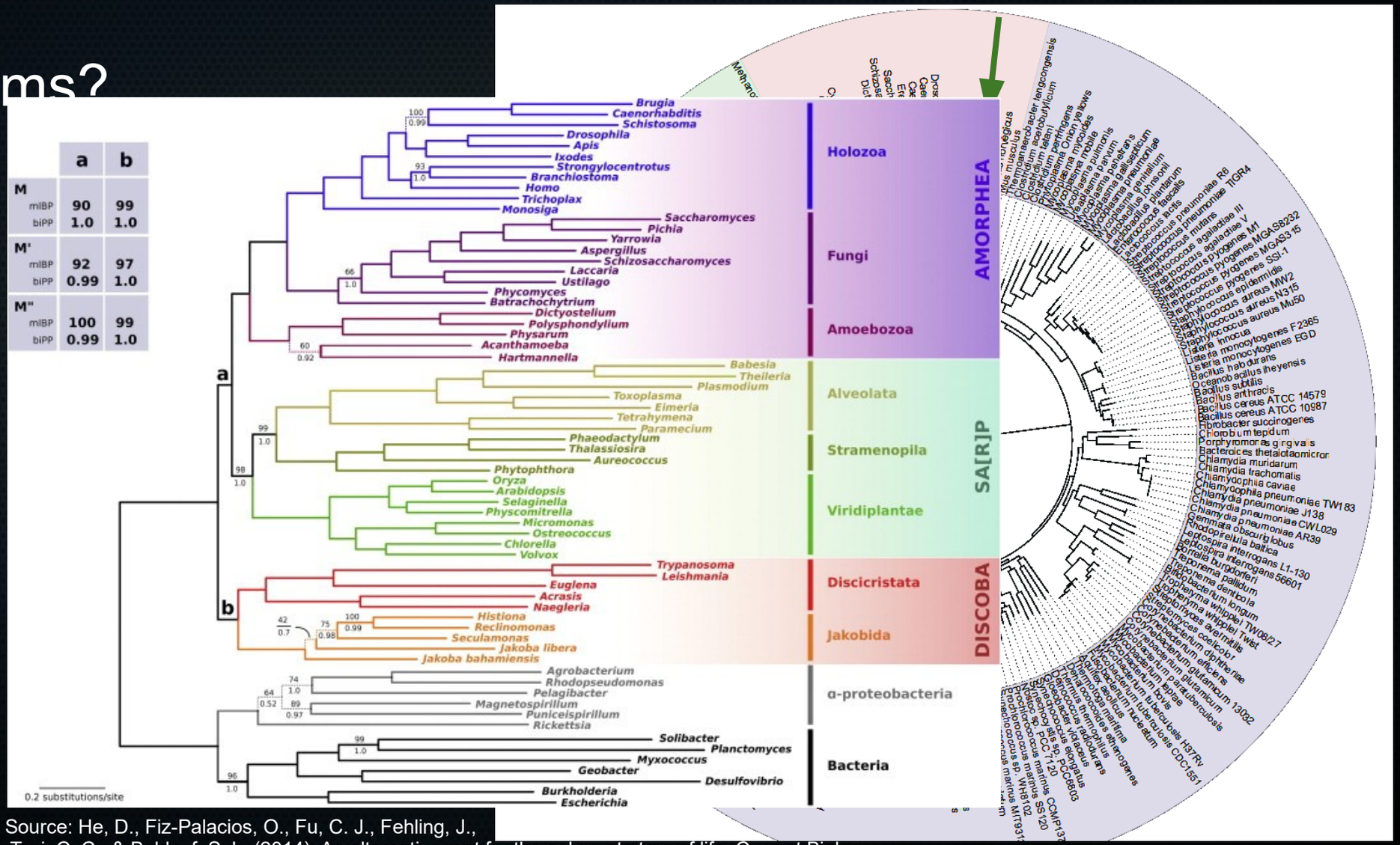
- What problems?



Source: Letunic, I., & Bork, P. (2021). Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. Nucleic acids research, gkab301. Advance online publication. https://doi.org/10.1093/nar/gkab301

Problems with phylogeny

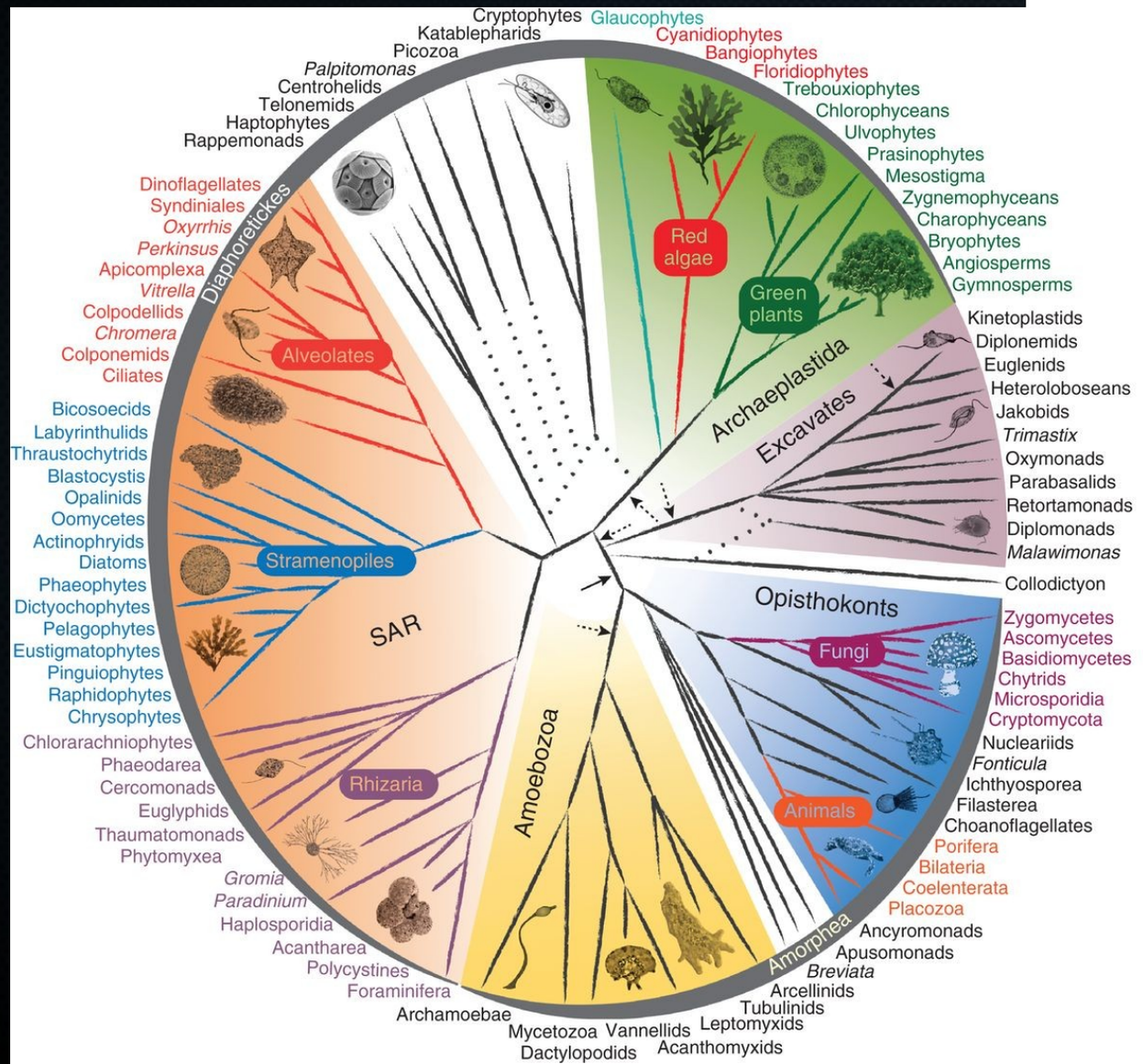
- What problems?

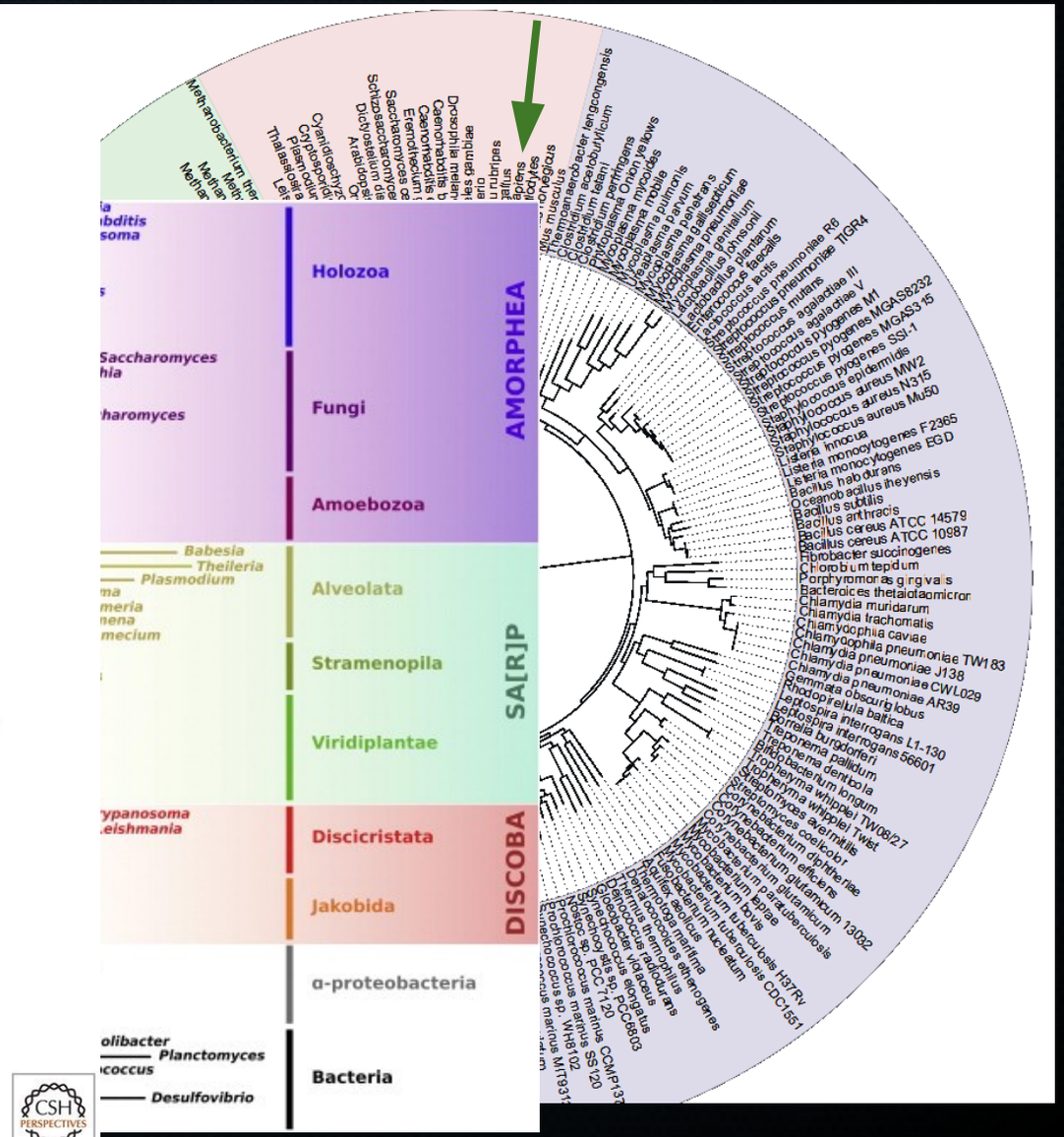


Source: He, D., Fiz-Palacios, O., Fu, C. J., Fehling, J., Tsai, C. C., & Baldauf, S. L. (2014). An alternative root for the eukaryote tree of life. Current Biology, 24(4), 465-470.

Problems with phylogeny

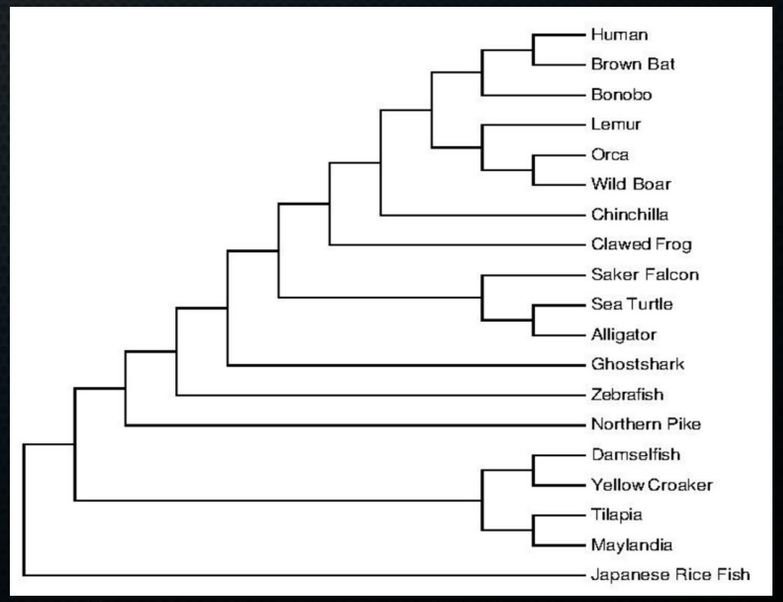
Source: Burki, F. (2014). The eukaryotic tree of life from a global phylogenomic perspective. Cold Spring Harbor perspectives in biology, 6(5), a016147.





Problems with phylogeny

- Pragmatically, these methods work extremely well and have yielded great insights.
- Under the hood, there are some caveats.



Source: https://commons.wikimedia.org/wiki/File:Phylogenetic_Tree.pdf

Phylogenetic shaky ground

- Problem 1: How do we get a tree?

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Phylogenetic shaky ground

- Problem 1: How do we get a tree?
  → Need to know correspondence between characters → which letter in which sequence corresponds to which letter in another sequence? → **alignment**

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Phylogenetic shaky ground

- Problem 1: correspondence between characters.

G	A	A	T	C	T
C	A	T	C		
T	A	T			

G	A	A	T	C	T
C	-	A	T	C	-
C	A	-	T	C	-

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

G	A	A	T	C	T
C	-	A	T	C	-
C	A	-	T	C	-

Every gap gives a gap penalty

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G A A T C T

-	-	A	T	C	G	A	C
-	0	-4	-8	-12	-16	-20	-24
C	-4	-3	-7	-3	-7	-11	-15
A	-8	1	-3	-7	-6	-2	-6
T	-12	-3	6	2	-2	-6	-5
A	-16	-7	2	3	-1	3	-1
C	-20	-11	-2	-1	0	-1	8

Every gap gives a gap penalty

Aligned via dynamic programming (Needleman Wunsch-like).

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
C	-	A	T	C	-
C	A	-	T	C	-

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
C	-	A	T	C	-
C	A	-	T	C	-

No way to know *which* A is gone.

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

If you‘d have a known ancestor of TAT, and it has ATAT →

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

If you‘d have a known ancestor of TAT, and it has ATAT →

G	A	A	T	C	T
-	-	A	T	A	T
A	T	A	T	-	-

Phylogenetic shaky ground

- Problem 1: correspondence between characters

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Would know this is best, but just pairwise alignment → can't know!

Two alternatives equally good!

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

If you'd have a known ancestor of TAT, and it has ATAT →

G	A	A	T	C	T
-	-	A	T	A	T
A	T	A	T	-	-

Phylogenetic shaky ground

- Problem 1: correspondence between characters
  So: can get best alignment between two sequences, but can already be non-unique. → in practice one is picked at random.

G	A	A	T	C	T
C	A	T	C		
T	A	T			

Two alternatives equally good!

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

Phylogenetic shaky ground

- What about three sequences?

G	A	A	T	C	T
C	A	T	C		
T	A	T			

For all three (or more!), have many different possibilities.
Many of those are as good. What do we pick?

Phylogenetic shaky ground

- What about *many more* sequences?

Source: https://2017-lapaz-assembly.readthedocs.io/en/latest/qiime1.html

```
>C12.06102014.R2.D01.TCACGGGAGTTG_1 HWI-M03127:41:ACE13:1:2107:7445:12380
GACGTAGGGAGCGAGCGTTCTCCGGATTTACTGGGCGTAAAGGGTCCGCAGGCGGGTGTTTAAGTTTCACCTGACAGCCTCCGGCTTAACCGGGGGAGGTGGTGGAAAACTGGAG
ACCTTGAGACATCGAGAGGCAGAGGGAATTCCGTGTGGAGCAGTGAAATGCGTAGAGATGCGGAAGAACACCAGAGGCGAAGGCGCTCTGCTGGCGATGCTCTGACGCTCAGGGA
CGAAAGCCAGGGGAGCAAACAGG
>C12.06102014.R2.D01.TCACGGGAGTTG_2 HWI-M03127:41:ACE13:1:1111:28215:12905
TACCAGCTCCCCGAGTGGTCGGGACGATTATTGGGCCTAAAGCATCCGTAGCCGGCTTCACAGGTCTCTTGTTAAATCCAACGGCTCAACCGTTGGACTGCAGGGGATACCATGG
AGCTAGGGGGCGGGAGAGGCGGACGGTACTCCATGGGTAGGGGTAAAATCCGTTGATCCATGGAAGACCACCAGTGGCGAAGGCGGTCCGCTAGAACGCGCCCGACGGTGAGGGA
TGAAAGCTGGGGGAGCGAACCGG
>C12.06102014.R2.D01.TCACGGGAGTTG_3 HWI-M03127:41:ACE13:1:2109:17440:18773
TACCAGCTCCCCGAGTGGTCGGGACGATTATTGGGCCTAAAGCATCCGTAGCCGGCTTCACAGGTCTCTTGTTAAATCCAACGGCTCAACCGTTGGACTGCAGGGGATACCATGG
AGCTAGGGGGCGGGAGAGGCGGACGGTACTCCATGGGTAGGGGTAAAAACCTTTGATCCATGGAAGAGCACCAGTGGCGAAGGCGGTCCGCTAGAACGCGCCCGACGGTGAGGGA
TGAAAGCTGGGGGAGCGGACCGG
>C12.06102014.R2.D01.TCACGGGAGTTG_4 HWI-M03127:41:ACE13:1:1106:5583:21341
TACCAGCTCCCCGAGTGGTCGGGACGATTATTGGGCCTAAAGCATCCGTAGCCGGCTTCACAGGTCTCTTGTTAAATCCAACGGCTCAACCGTTGGACTGCAGGGGATACCATGG
AGCTAGGGGGCGGGAGAGGCGGACGGTACTCCATGGGTAGGGGTAAAATCCTTTGATCCATGGAAGACCACCAGTGGCGAAGGCGGTCCGCTAGAACGCGCCCGACGGTGAGGGA
TGAGAGCTGGGGGAGCGAACCGG
>C12.06102014.R2.D01.TCACGGGAGTTG_5 HWI-M03127:41:ACE13:1:2106:15029:18423
TACCAGCTCCCCGAGTGGTCGGGACGATTATTGGGCCTAAAGCATCCGTAGCCGGCTTCACAGGTCTCTTGTTAAATCCAACGGCTCAACCGTTGGACTGCAGGGGATACCATGG
```

Phylogenetic shaky ground

- What about *many more* sequences?
- Becomes impossible to try all possibilities, cannot use Dynamic Programming (Needleman-Wunsch-like) approach for more than 3 or 4 sequences. So we use heuristics to get something.

Progressive tree alignment

- Idea: we can do pairwise alignment well

Progressive tree alignment

- Idea: we can do pairwise alignment well
- So: do pairwise alignment on all sequences, get their pairwise distances (non-agreeing bases and gaps), cluster them hierarchically.

Progressive tree alignment

- Idea: we can do pairwise alignment well
- So: do pairwise alignment on all sequences, get their pairwise distances (non-agreeing bases and gaps), cluster them hierarchically.
- Then, to get a multiple sequence alignment: align the two closest sequences, add the third closest to that, fourth closest, etc.
  Note: this means that you continously make a pairwise alignment, where the already aligned sequences are taken as one.

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA…
3 ...CCCCCCCCCC…
4 ...GGGGGCCCAC…

Pairwise alignment
(Needleman-Wunsch)

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

Example
(used EMBOSS Needle)

```
#=======================================
#
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 13
# Identity:       7/13 (53.8%)
# Similarity:     7/13 (53.8%)
# Gaps:           6/13 (46.2%)
# Score: 13.5
#
#
#=======================================

EMBOSS_001         1 -ATG--CTGTTTA     10
                     |||  |   |||
EMBOSS_001         1 AATGGCC---TTA     10
```

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

Example
(used EMBOSS Needle)

```
#=======================================
#
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 13
# Identity:       7/13 (53.8%)
# Similarity:     7/13 (53.8%)
# Gaps:           6/13 (46.2%)
# Score: 13.5
#
#
#=======================================

EMBOSS_001         1 -ATG--CTGTTTA     10
                     |||  |   |||
EMBOSS_001         1 AATGGCC---TTA     10
```

	1	2	3	4
1	50	13.5		
2	-	50		
3	-	-	50	
4	-	-	-	50

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

Example
(used EMBOSS Needle)

```
#=======================================
#
# Aligned_sequences: 2
# 1: EMBOSS_001
# 2: EMBOSS_001
# Matrix: EDNAFULL
# Gap_penalty: 10.0
# Extend_penalty: 0.5
#
# Length: 13
# Identity:       7/13 (53.8%)
# Similarity:     7/13 (53.8%)
# Gaps:           6/13 (46.2%)
# Score: 13.5
#
#
#=======================================

EMBOSS_001         1 -ATG--CTGTTTA     10
                     |||  |   |||
EMBOSS_001         1 AATGGCC---TTA     10
```

Note that real similarity is not measured by what letters match exactly, but by substitution matrices (based on many aligned sequences and substitution rates in those sequences), think BLOSUM62 and PAM for proteins.

	1	2	3	4
1	50	13.5		
2	-	50		
3	-	-	50	
4	-	-	-	50

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

Example
(used EMBOSS Needle)

	A	T	G	C	S	W	R	Y	K	M	B	V	H	D	N
A	5	-4	-4	-4	-4	1	1	-4	-4	1	-4	-1	-1	-1	-2
T	-4	5	-4	-4	-4	1	-4	1	1	-4	-1	-4	-1	-1	-2
G	-4	-4	5	-4	1	-4	1	-4	1	-4	-1	-1	-4	-1	-2
C	-4	-4	-4	5	1	-4	-4	1	-4	1	-1	-1	-1	-4	-2
S	-4	-4	1	1	-1	-4	-2	-2	-2	-2	-1	-1	-3	-3	-1
W	1	1	-4	-4	-4	-1	-2	-2	-2	-2	-3	-3	-1	-1	-1
R	1	-4	1	-4	-2	-2	-1	-4	-2	-2	-3	-1	-3	-1	-1
Y	-4	1	-4	1	-2	-2	-4	-1	-2	-2	-1	-3	-1	-3	-1
K	-4	1	1	-4	-2	-2	-2	-2	-1	-4	-1	-3	-3	-1	-1
M	1	-4	-4	1	-2	-2	-2	-2	-4	-1	-3	-1	-1	-3	-1
B	-4	-1	-1	-1	-1	-3	-3	-1	-1	-3	-1	-2	-2	-2	-1
V	-1	-4	-1	-1	-1	-3	-1	-3	-3	-1	-2	-1	-2	-2	-1
H	-1	-1	-4	-1	-3	-1	-3	-1	-3	-1	-2	-2	-1	-2	-1
D	-1	-1	-1	-4	-3	-1	-1	-3	-1	-3	-2	-2	-2	-1	-1
N	-2	-2	-2	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Source: http://rosalind.info/glossary/dnafull/

Table 1. The IUPAC nucleic acid notation

	Symbol	Meaning	Mnemonic
DNA Bases	G	Guanine	Guanine
	T	Thymine	Thymine
	A	Adenine	Adenine
	C	Cytosine	Cytosine
Ambiguity Characters	R	G + A	puRine
	Y	T + C	pYrimidine
	S	G + C	Strong interactions (3 H bonds)
	W	T + A	Weak interactions (2 H bonds)
	K	G + T	Keto
	M	A + C	aMino
	D	G + T + A	Not-C (D follows C in alphabet)
	H	T + A + C	Not-G (H follows G)
	B	G + T +C	Not-A (B follows A)
	V	G + A + C	Not-T or U (V follows U)
	N	G + A + T + C	aNy

Source: http://rosalind.info/glossary/iupac-notation/

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

	1	2	3	4
1	50	13.5	0	1
2	-	50	0	3
3	-	-	50	16
4	-	-	-	50

Similarity

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

	1	2	3	4
1	50	13.5	0	1
2	-	50	0	3
3	-	-	50	16
4	-	-	-	50

Similarity

Make a guide tree

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

	1	2	3	4
1	50	13.5	0	1
2	-	50	0	3
3	-	-	50	16
4	-	-	-	50

Similarity

Make a guide tree

Guide tree does not correspond with pairwise distance matrix, what gives!?

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment (Needleman-Wunsch)

	1	2	3	4
1	50	13.5	0	1
2	-	50	0	3
3	-	-	50	16
4	-	-	-	50

Similarity

Make a guide tree

Used Clustal Omega, which calculates distances slightly differently (optimised for large alignment + with different IUB DNA matrix where every match scores 1.9 and every mismatch is 0*)

*with X and N matching anything.

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Needleman-Wunsch)

	1	2	3	4
1	50	13.5	0	1
2	-	50	0	3
3	-	-	50	16
4	-	-	-	50

Similarity

Make a guide tree

Our guide tree
would look like this.

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment
(Clustal Omega)

	1	2	3	4
1	50	?	?	?
2	-	50	?	?
3	-	-	50	?
4	-	-	-	50

Similarity

Make a guide tree
(with Clustal Omega
distances)

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment (Clustal Omega)

	1	2	3	4
1	50	?	?	?
2	-	50	?	?
3	-	-	50	?
4	-	-	-	50

Similarity

Make a guide tree (with Clustal Omega distances)

Make a MSA by aligning along this guide tree

```
1      -ATGCTGTTTA      10
2      AATGGCCTTA-      10
        ***   **
```

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment (Clustal Omega)

	1	2	3	4
1	50	?	?	?
2	-	50	?	?
3	-	-	50	?
4	-	-	-	50

Similarity

Make a guide tree (with Clustal Omega distances)

Make a MSA by aligning along this guide tree

```
4        --GGGGGCCCAC     10
1        -ATGCTGTTTA-     10
2        AATGGCCTTA--     10
             *
```

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment (Clustal Omega)

	1	2	3	4
1	50	?	?	?
2	-	50	?	?
3	-	-	50	?
4	-	-	-	50

Similarity

Make a guide tree (with Clustal Omega distances)

Make a MSA by aligning along this guide tree

```
3      --CCCCCCCCCC     10
4      --GGGGGCCCAC     10
1      -ATGCTGTTTA-     10
2      AATGGCCTTA--     10
```

Progressive tree alignment

1 ...ATGCTGTTTA...
2 ...AATGGCCTTA...
3 ...CCCCCCCCCC...
4 ...GGGGGCCCAC...

Pairwise alignment (Clustal Omega)

	1	2	3	4
1	50	?	?	?
2	-	50	?	?
3	-	-	50	?
4	-	-	-	50

Similarity

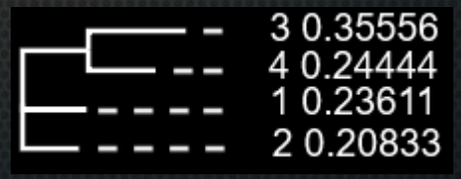
Make a guide tree (with Clustal Omega distances)

Make a MSA by aligning along this guide tree

```
3      --CCCCCCCCCC     10
4      --GGGGGCCCAC     10
1      -ATGCTGTTTA-     10
2      AATGGCCTTA--     10
```

Can now:

- calculate distances between aligned sequences
- choose a joining criterion (here: neighbour-linking)
- make an (unrooted) phylogenetic tree!



Progressive tree alignment

- Caveat: this introduces a feedback loop and doesn‘t ensure that we get the best tree!

Progressive tree alignment

- Caveat: this introduces a feedback loop and doesn‘t ensure that we get the best tree!
- We say: we want to align these sequences, and then cluster them to make a tree.

Progressive tree alignment

- Caveat: this introduces a feedback loop and doesn‘t ensure that we get the best tree!
- We say: we want to align these sequences, and then cluster them to make a tree.
- We cannot align all the sequences together: incalculable.

Progressive tree alignment

- Caveat: this introduces a feedback loop and doesn‘t ensure that we get the best tree!
- We say: we want to align these sequences, and then cluster them to make a tree.
- We cannot align all the sequences together: incalculable.
- So what we do: pairwise alignments, score how well the sequence pairs align (distance/similarity), make a guide tree based on that, use it to align all together, then make a tree. This does *not* guarantee the *best* alignment of all sequences (global alignment).

Progressive tree alignment

- Caveat: this introduces a feedback loop and doesn‘t ensure that we get the best tree!
- We say: we want to align these sequences, and then cluster them to make a tree.
- We cannot align all the sequences together: incalculable.
- So what we do: pairwise alignments, score how well the sequence pairs align (distance/similarity), make a guide tree based on that, use it to align all together, then make a tree. This does *not* guarantee the *best* alignment of all sequences (global alignment).
- So we need a non-optimal/non-true tree to align the sequences, to make our tree. Do you see the strange recursion there?

Summary so far:

- Want to cluster nucleotide sequences based on evolutionary relatedness (phylogeny, phylogenetic tree)
- Need to align them all to calculate distances
- Can‘t do that directly

Summary so far:

- Want to cluster nucleotide sequences based on evolutionary relatedness (phylogeny, phylogenetic tree)
- Need to align them all to calculate distances
- Can‘t do that directly
- Instead align each sequence with each other sequence, and calculate how good they align (similarity, i.e. inverse of distance) → cluster this hierarchically → guide tree
- Now we can make our total alignment by adding sequences along the guide tree → not guaranteed to be the best alignment!

Summary so far:

- Want to cluster nucleotide sequences based on evolutionary relatedness (phylogeny, phylogenetic tree)
- Need to align them all to calculate distances
- Can‘t do that directly
- Instead align each sequence with each other sequence, and calculate how good they align (similarity, i.e. inverse of distance) → cluster this hierarchically → guide tree
- Now we can make our total alignment by adding sequences along the guide tree → not guaranteed to be the best alignment!
- With the MSA in hand, we can calculate all distances, cluster hierarchically, and make a phylogenetic tree

So are we done?

So are we done?

- Well, no:

So are we done?

- Well, no:
  - When aligning, we use gap penalties. How do you pick them and how to think about them?

So are we done?

- Well, no:
  - When aligning, we use gap penalties. How do you pick them and how to think about them?
  - What about aligning thousands or hundreds of thousands of sequences? Does progressive alignment work fine?

So are we done?

- Well, no:
  - When aligning, we use gap penalties. How do you pick them and how to think about them?
  - What about aligning thousands or hundreds of thousands of sequences? Does progressive alignment work fine?
  - What about current best practice?

Gaps and evolution

- Actually have separate scores for opening a gap (introducing a mismatch/SNP) and extending a gap.

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

G	A	A	T	C	T
C	A	T	C		
T	A	T			

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

Gaps and evolution

- Actually have separate scores for opening a gap (introducing a mismatch/SNP) and extending a gap.
- Shouldn‘t hypothesise mutations all over the place, but a mutation can delete more than one base: cost opening >> extending

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

G	A	A	T	C	T
C	A	T	C		
T	A	T			

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

Gaps and evolution

- Actually have separate scores for opening a gap (introducing a mismatch/SNP) and extending a gap.
- Shouldn‘t hypothesise mutations all over the place, but a mutation can delete more than one base: cost opening >> extending

```
3     --CCCCCCCCCC     10
4     --GGGGGCCCAC     10
1     -ATGCTGTTTA-     10
2     AATGGCCTTA--     10
```

G	A	A	T	C	T
C	A	T	C		
T	A	T			

G	A	A	T	C	T
-	-	-	T	A	T
-	T	A	T	-	-

Gaps and evolution

- For phylogenetic tree: good to have this balance in principle.

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

Gaps and evolution

- For phylogenetic tree: good to have this balance in principle.

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

- But: SNPs and small indels are *far* from the sole mediators of evolution!
  We also have huge deletions, whole genome duplications, or large chromosomal rearrangements. That is just *one* evolutionary event, however our gap extension penalty will not favour an alignment that shows these!

Gaps and evolution

- For phylogenetic tree: good to have this balance in principle.

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

- But: SNPs and small indels are *far* from the sole mediators of evolution!
  We also have huge deletions, whole genome duplications, or large chromosomal rearrangements. That is just *one* evolutionary event, however our gap extension penalty will not favour an alignment that shows these!
  → Our idea is that the tree shows evolutionary relatedness, but given how it works, it does that best for small changes only.

Gaps and evolution

- What about clustering nucleotide sequences for a different reason than phylogeny?

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

Gaps and evolution

- What about clustering nucleotide sequences for a different reason than phylogeny?

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

- For example: align homologs of a protein that catalyses lipid flipping (flippase) → want to know if they are functional in all ancestors and in which organisms the gene lost function.

Gaps and evolution

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

- What about clustering nucleotide sequences for a different reason than phylogeny?
- For example: align homologs of a protein that catalyses lipid flipping (flippase) → want to know if they are functional in all ancestors and in which organisms the gene lost function.
- Then: gap penalty and extension penalty could perhaps better be the same → every deletion of a residue in the protein means that the protein is less likely to belong with its functional brethren.

Gaps and evolution

```
3     --CCCCCCCCCC    10
4     --GGGGGCCCAC    10
1     -ATGCTGTTTA-    10
2     AATGGCCTTA--    10
```

- What about clustering nucleotide sequences for a different reason than phylogeny?
- For example: align homologs of a protein that catalyses lipid flipping (flippase) → want to know if they are functional in all ancestors and in which organisms the gene lost function.
- Then: gap penalty and extension penalty could perhaps better be the same → every deletion of a residue in the protein means that the protein is less likely to belong with its functional brethren.
- Note also that for proteins, the functional unit of evolution is the amino acid. Hence might be better to align proteins first, then codon triplets coding for them in DNA (Kapli, P., Yang, Z., & Telford, M. J. (2020). Phylogenetic tree building in the genomic age. Nature Reviews Genetics, 21(7), 428-444.).

So are we done?

- Well, no:
  - ~~When aligning, we use gap penalties. How do you pick them and how to think about them?~~
    →separate opening and extending costs.
    →adding per-base extension penalties makes large evolutionary events less likely to show up as one event in your evolutionary tree.
  - What about aligning thousands or hundreds of thousands of sequences? Does progressive alignment work fine?
  - What about current best practice?

Progressive alignment of many sequences

- Still a problem:
  Calculating all pairwise distances between 100.000 or more sequences requires *N*^2 calculations. Infeasible.

Progressive alignment of many sequences

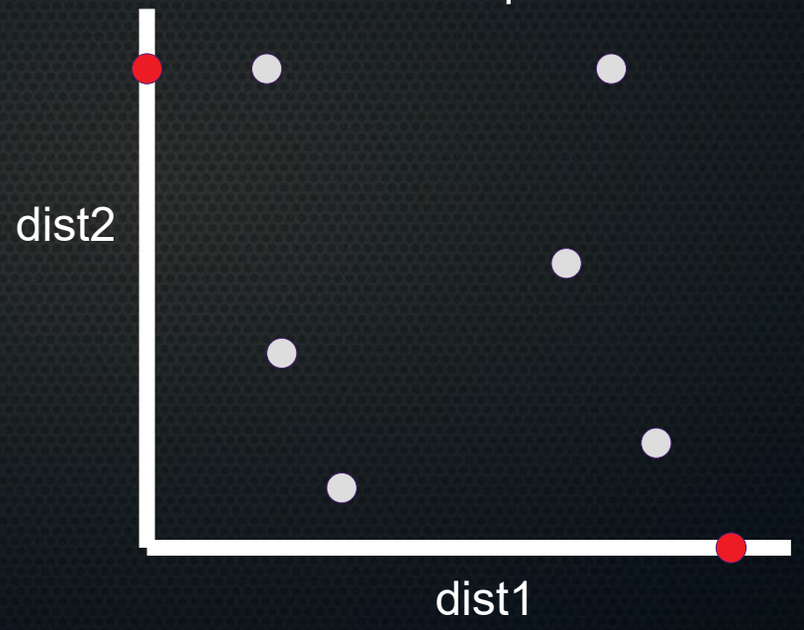
- Still a problem:
  Calculating all pairwise distances between 100.000 or more sequences requires *N*^2 calculations. Infeasible.
- Solution: take log2(100.000) *seed sequences*. Calculate distance of each sequence to each seed sequence (by pairwise alignment).

Progressive alignment of many sequences

- Now you can treat each sequence as a vector of these distances:

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$

Illustration of 8 sequences, 2 of which are seed sequences



Progressive alignment of many sequences

- Now you can treat each sequence as a vector of these distances:

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$

Illustration of 8 sequences, 2 of which are seed sequences

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \end{bmatrix} = \begin{bmatrix} dist_1 \\ dist_2 \end{bmatrix}$$

Progressive alignment of many sequences

- Now you can treat each sequence as a vector of these distances:

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$

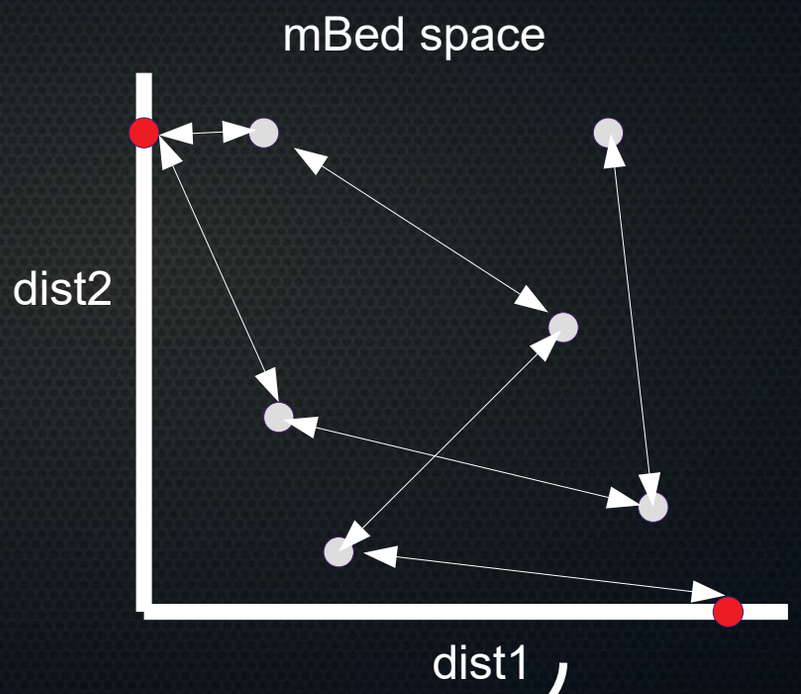
Illustration of 8 sequences, 2 of which are seed sequences

Distance of a seed sequence to itself is 0, to the other seed sequence quite large!

Progressive alignment of many sequences

- Now you can treat each sequence as a vector of these distances:

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$



Can easily calculate distances between vectors (linear algebra is fast!)

Progressive alignment of many sequences

- In essence we thus assume that the seed sequences are sufficiently dissimilar that calculating distances to them is informative enough to approximate a full distance matrix.

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$

Progressive alignment of many sequences

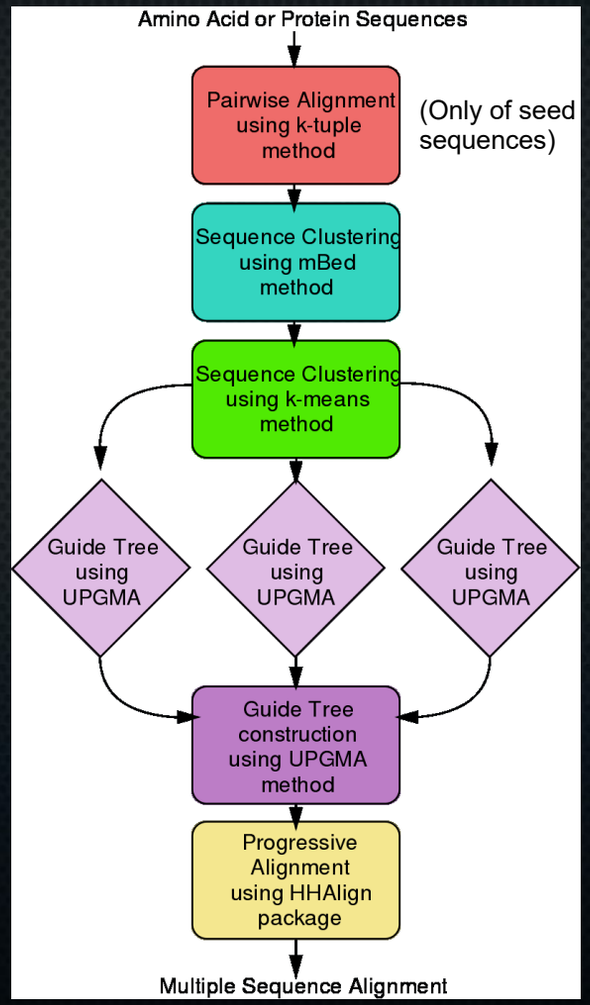
- In essence we thus assume that the seed sequences are sufficiently dissimilar that calculating distances to them is informative enough to approximate a full distance matrix.
- Checks are in place to make sure of this
  (no duplicates, sample stratified by sequence length)

(Blackshields, G., Sievers, F., Shi, W., Wilm, A., & Higgins, D. G. (2010). Sequence embedding for fast construction of guide trees for multiple sequence alignment. Algorithms for Molecular Biology, 5(1), 1-11.).

$$\begin{bmatrix} dist(thisSeq, seedSeq_1) \\ dist(thisSeq, seedSeq_2) \\ dist(thisSeq, seedSeq_3) \\ \dots \\ dist(thisSeq, seedSeq_n) \end{bmatrix}$$

Progressive alignment of many sequences

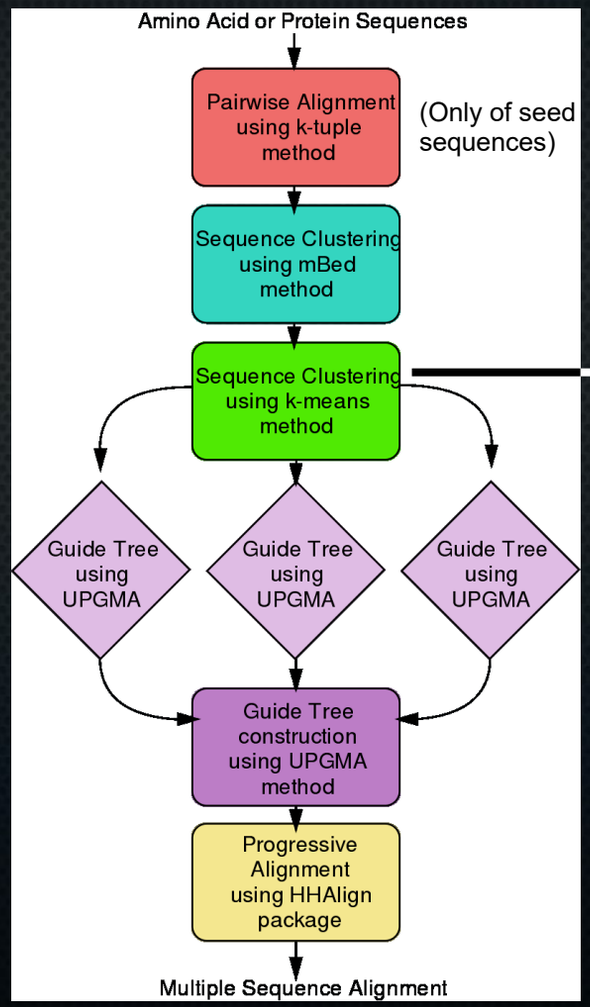
- Actually aligners use more short-cuts and tricks for large alignments.

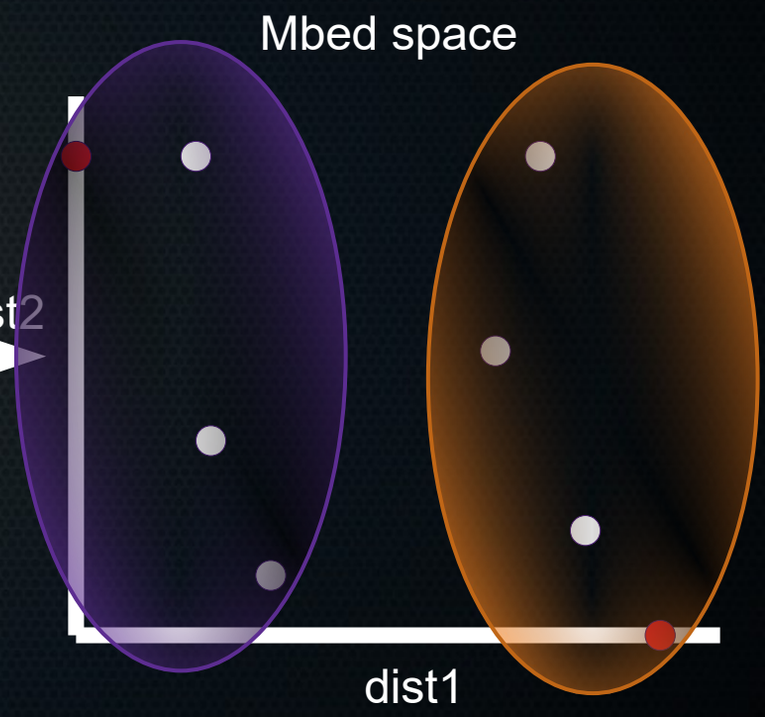


Source: https://en.wikipedia.org/wiki/Clustal#/media/File:Clustal_Omega_Algorithm_Flowchart.svg

Progressive alignment of many sequences

- Actually aligners use more short-cuts and tricks for large alignments.

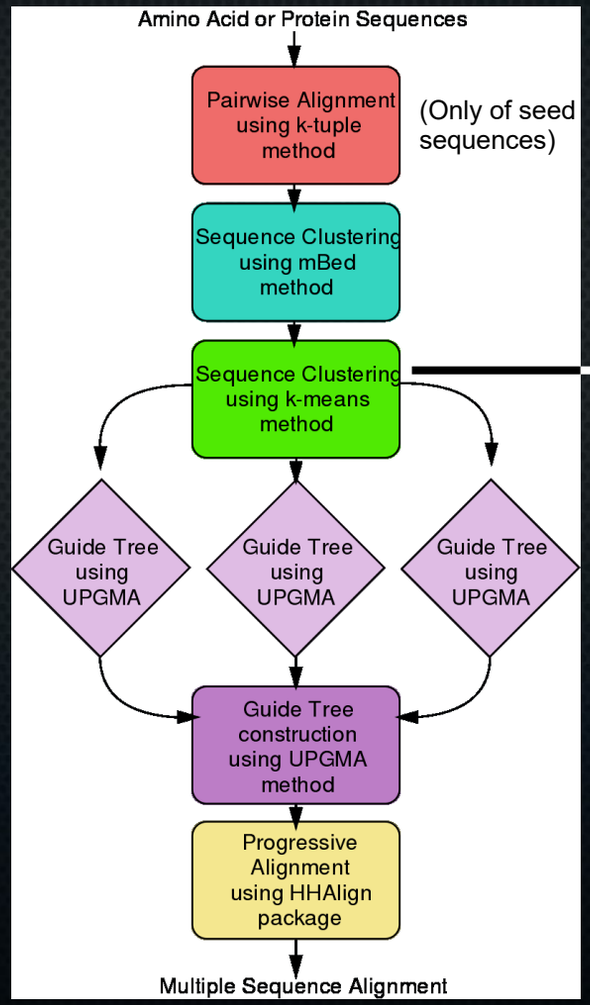


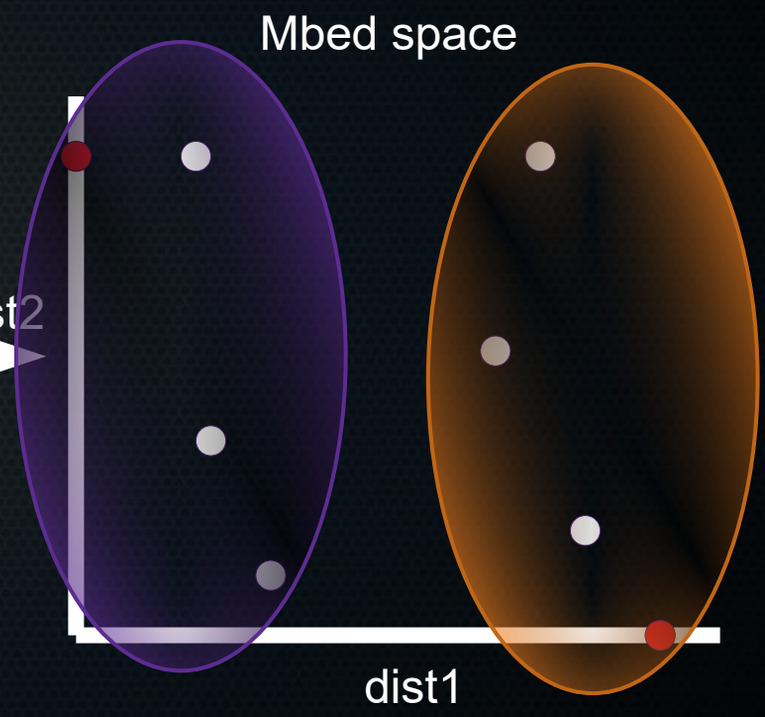


Source: https://en.wikipedia.org/wiki/Clustal#/media/File:Clustal_Omega_Algorithm_Flowchart.svg

Progressive alignment of many sequences

- Actually aligners use more short-cuts and tricks for large alignments.





Make separate guide trees per cluster for large datasets, connect them later

Source: https://en.wikipedia.org/wiki/Clustal#/media/File:Clustal_Omega_Algorithm_Flowchart.svg

So are we done?

- Well, no:
  - ~~When aligning, we use gap penalties. How do you pick them and how to think about them?~~
    →separate opening and extending costs.
    →adding per-base extension penalties makes large evolutionary events less likely to show up as one event in your evolutionary tree.
  - ~~What about aligning thousands or hundreds of thousands of sequences? Does progressive alignment work fine?~~
    →No, need to use embedding to reduce distance calculations
  - What about current best practice?

Current best practice

- Two parts:
  - How do we get the MSA?
  - How do we make the tree once we have the MSA?

Current best practice: making the MSA

- Progressive alignment used often (Clustal, Muscle)
- Alternative: consistency-based methods (T-Coffee, ProbCons)
  →Do pairwise alignment, but also keep track of not-quite optimal pairwise alignment
  → When making the total MSA, look back at these not-quite optimal pairwise alignments, and try to maximise total score of all sequences.

Current best practice: making the MSA

- Progressive alignment used often (Clustal, Muscle)
- Alternative: consistency-based methods (T-Coffee, ProbCons)
- Alternative 2: statistical/Bayesian approaches (Bali-Phy, StatAlign)
  → assume complete evolutionary models (i.e. how often deletions occur, base changes, etc.), and fit tree and MSA simultaneously → most sound, but computationally very expensive.

Current best practice

- Two parts:
  - ~~How do we get the MSA?~~
  - How do we make the tree once we have the MSA?

Current best practice: making the tree

- Most-used: Bayesian (MrBayes, RevBayes) and Maximum Likelihood (IQ-TREE, RAxML) approaches.
  → won’t go into this here, but safe to say that there’s a lot more sophistication when going from MSA to phylogenetic tree!

Current best practice

- Two parts:
  - ~~How do we get the MSA?~~
  - ~~How do we make the tree once we have the MSA?~~

So are we done?

- Well, yes!