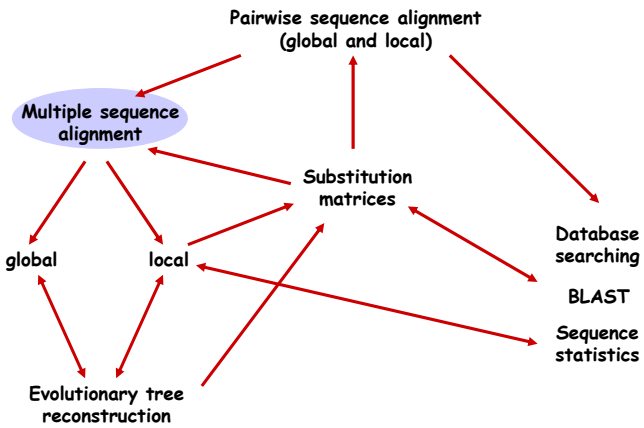


Global Multiple Sequence Alignment

```
HUMAN  MKWVTFISLL FLFSSAYSRG V..FRRDA.H KSEVAHRFKD LGEENFKALV
RABIT  MKWVTFISLL FLFSSAYSRG V..FRREA.H KSEIAHRFND VGEEHFIGLV
PIG    ~~WVTFISLL FLFSSAYSRG V..FRRDT.Y KSEIAHRFKD LGEQYFKGLV
CHICK  MKWVTLISFI FLFSSATSRN LQRFARDAEH KSEIAHRYND LKEETFKAVA
```

Align *k* sequences, so that residues in each column share a property of interest:

- a common ancestor
- a structural or functional role

Global Multiple Sequence Alignment

Given sequences $s_1 \ldots s_k$ of lengths $n_1 \ldots n_k$

seek $s'_1 \ldots s'_k$ of length $l \geq max\{n_i\}$ such that

- Obtain s_i from s'_i by removing gaps
- No column contains all gaps
- The score of the alignment is optimal

Scoring function: Sum-of-Pairs

$$Score = \Sigma^k_{a=1} \Sigma^k_{a=1} \Sigma_{b > a}\, p(s'_a[i], s'_b[i])$$

(1) **A_TT**
(2) **A_T_**
(3) **ACAT**

$p[\_,\_] = 0$

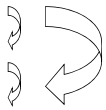
$$Score = p[s_1,s_2] + p[s_1,s_3] + p[s_2,s_3]$$
$$= 0 + g + g = 2g$$

Note: this example uses a similarity function. We can also use Sum-of-Pairs with distance scoring.

Scoring function: Sum-of-Pairs

$$Score = \sum^{k}_{a=1} \sum^{k}_{a=1} \sum_{b > a} p(s'_a[i], s'_b[i])$$

(1) **A_TT**
(2) **A_T_**
(3) **ACAT**



$p[\_,\_] = 0$

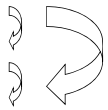
$$Score = p[s_1,s_2] + p[s_1,s_3] + p[s_2,s_3]$$
$$= M + m + m = 2m + M$$

Note: this example uses a similarity function. We can also use Sum-of-Pairs with distance scoring.

Scoring function: Sum-of-Pairs

$$Score = \sum^{k}_{a=1} \sum^{k}_{a=1} \sum_{b > a} p(s'_a[i], s'_b[i])$$

(1) **A_TT**
(2) **A_T_**
(3) **ACAT**



$p[\_,\_] = 0$

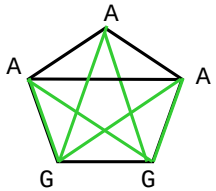
$$Score = p[s_1,s_2] + p[s_1,s_3] + p[s_2,s_3]$$
$$= g + M + g = 2g + M$$

Note: this example uses a similarity function. We can also use Sum-of-Pairs with distance scoring.

Scoring Multiple Alignments

Sum of Pairs

SP = 6

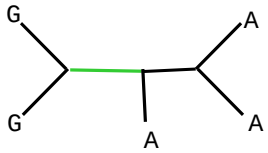


$$Score = \sum^{k}_{x=1} \sum_{y > x} d(s'_x[j], s'_y[j])$$

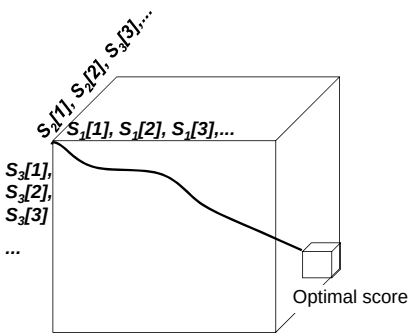
A
A
A
G
G

Tree alignment

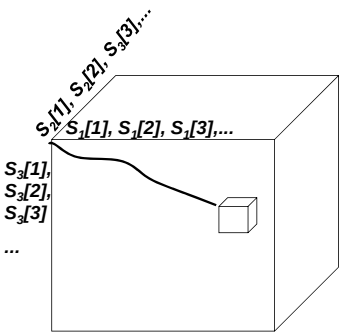
TA = 1



Dynamic Programming for Multiple Alignment



Dynamic Programming for Multiple Alignment



Each cell has $O(2^k)$ neighboring cells

Calculating the sum-of-pairs score for each neighbor is $O(k^2)$

Number of cells in matrix: $O(n^k)$

Total computational complexity:

$O(n^k 2^k k^2)$

Limits:
- ~ k = 8 - 10 sequences
- ~ n = 500 residues

MSA is NP-complete for Sum-of-Pairs scoring

Observations

1. A multiple alignment induces pairwise alignments
2. A column in the induced pairwise alignment may contain all gaps, even though no column in the MSA contains all gaps.

```
(1) AG_CT
(2) AG_CT
(3) ACT_T
```

3. The pairwise alignments induced by the *optimal multiple alignment* are *not* the same as the *optimal pairwise alignments.*

Optimal Pairwise Alignments

```
(1) ACT
(2) AGT
```

1 substitution

Optimal Multiple Alignment

```
(1) AC_T
(2) A_GT
(3) ACGT
```

2 indels

Although this costs more, it may be a biologically more realistic alignment

Since exact methods for MSA have exponential time complexity, heuristic approaches are used. Progressive alignment is the most commonly used.

Basic progressive alignment strategy:

- Compute *D,* a matrix of distances between all pairs of sequences
- From *D,* construct a "guide tree" *T*
- Construct MSA by pairwise alignment of partial alignments ("profiles") guided by T
- Improve alignment by postprocessing steps.

Optimal Pairwise Alignments

(1) ACTCAT
(2) AGTCAT
(3) ACGTCCT

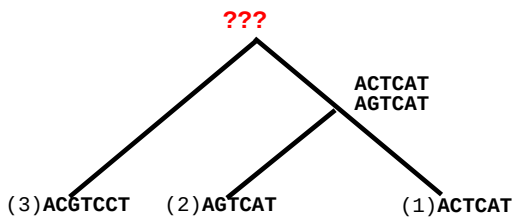
(1) ACTCAT
(2) AGTCAT — 3

(2) A_GTCAT
(3) ACGTCCT — 5

(1) AC_TCAT
(3) ACGTCCT — 5

d(x,y) = 3
d(x,"_") = 2

Progressive Alignment



- Use *profile alignment* to merge sequences according to a guide tree.
- Typically, most closely related sequences are merged first.

Merging strategy:
Align the profile (1,2) with sequence (3)

(1) ACTCAT
(2) AGTCAT
(3) ACGTCCT

(1) ACTCAT
(2) AGTCAT — 3

(2) A_GTCAT
(3) ACGTCCT — 5

(1) AC_TCAT
(3) ACGTCCT — 5

d(x,y) = 3
d(x,"_") = 2

```
  _ A C T C A T
  _ A G T C A T
    4
_
A
C
G
T
C
C
T
```

d(x,y) = 3
d(x,"_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8				
A							
C							
G							
T							
C							
C							
T							

A C
A G
- -

d(x,y) = 3

d(x, "_") = 2

Note: no penalty for mutations in the profile.
We paid for those in a previous step

A C
A G
_ _

4+4

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4						
C	8						
G	12						
T	16						
C	...						
C							
T							

d(x,y) = 3

d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0					
C	8						
G	12						
T	16						
C	...						
C							
T							

A
A
A

d(x,y) = 3

d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4				
C	8						
G	12						
T	16						
C	...						
C							
T							

A C
A G
A -

d(x,y) = 3
d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4	8			
C	8						
G	12						
T	16						
C	...						
C							
T							

d(x,y) = 3
d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4	8	12		
C	8						
G	12						
T	16						
C	...						
C							
T							

d(x,y) = 3
d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4	8	12	16	
C	8						
G	12						
T	16						
C	...						
C							
T							

d(x,y) = 3
d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4	8	12	16	20
C	8	4					
G	12						
T	16						
C	...						
C							
T							

A _
A _
A C

d(x,y) = 3

d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	0	4	8	12	16	20
C	8	4	3				
G	12						
T	16						
C	...						
C							
T							

A C
A G
A C

d(x,y) = 3

d(x, "_") = 2

	_	A	C	T	C	A	T
	_	A	G	T	C	A	T
_		4	8	12	16	20	24
A	4	**0**	4	8	12	16	20
C	8	**4**	**3**	7	8	12	16
G	12	8	**7**	9	12	14	18
T	16	12	11	**7**	11	15	14
C	20	16	15	11	**7**	11	15
C	24	20	19	15	11	**13**	17
T	28	24	23	19	15	17	**13**

d(x,y) = 3

d(x, "_") = 2

Optimal Pairwise Alignments

(1) ACTCAT
(2) AGTCAT

(2) A_GTCAT
(3) ACGTCCT

(1) AC_TCAT
(3) ACGTCCT

Progressive alignment

(1,2) + (3)

(3) ACGTCCT
(1) AC_TCAT
(2) AG_TCAT

4m+2g

An alternate alignment

(1) AC_TCAT
(2) A_GTCAT
(3) ACGTCCT

2m+4g

Optimal Pairwise Alignments

(1) ACTCAT
(2) AGTCAT

(2) A_GTCAT
(3) ACGTCCT

(1) AC_TCAT
(3) ACGTCCT

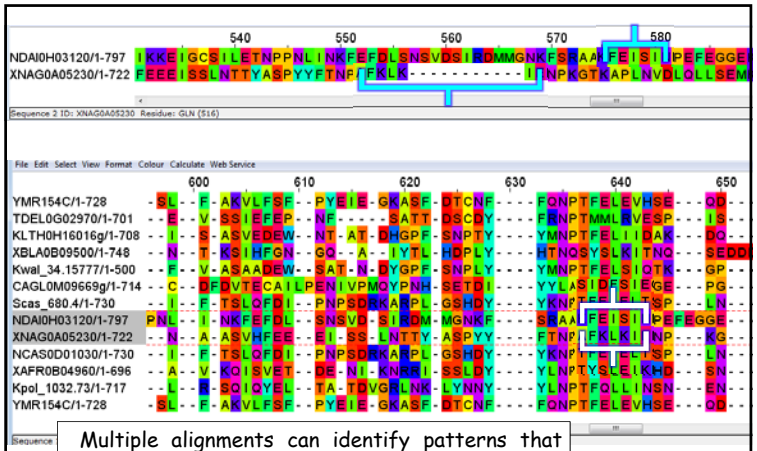
Progressive alignment

(1,2) + (3)

(3) ACGTCCT
(1) AC_TCAT 16
(2) AG_TCAT

An alternate alignment

(1) AC_TCAT
(2) A_GTCAT 14
(3) ACGTCCT



Multiple alignments can identify patterns that are conserved in a family but not apparent in a pairwise alignment of two family members.

Progressive alignment

- "Once a gap, always a gap"
  - You can't go back and correct a bad decision at an earlier step.
- Progressive alignment is not guaranteed to give the optimal alignment.
- But it does have better complexity…

Complexity of progressive alignment

- Distance matrix
  - Each pairwise alignment $O(n^2)$
  - Number of pairwise alignments $O(k^2)$
- Iterative construction of MSA
  - Number of merge steps $O(k)$
  - Each pairwise alignment $O(k^2n^2)$

Entire method $O(k^2n^2)$

Summary: Progressive alignment heuristics

- Not guaranteed to give the optimal MSA
- Bad choice of gaps propagates
- Complexity
  - Progressive: $O(k^2n^2)$
  - versus DP: $O(n^k\, 2^k\, k^2)$
- Typically, merge the most closely related sequences first.

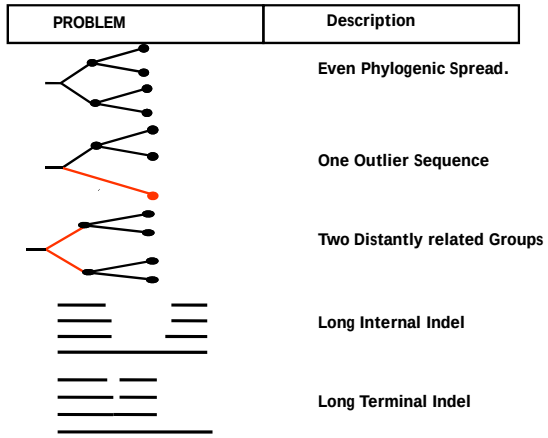
Mathematical correctness is not a guarantee of biological accuracy. The performance of MSA programs is typically evaluated using benchmarks based on biological data:

- Curated structural alignment
- Automated structural alignment
- Real or simulated sequence

Various benchmarks are designed to mimic properties of different types of data sets encountered in practice, especially those that are challenging to align:

- Highly divergent sequences, e.g., <50% or <30% identity
- A family of related sequences plus several outliers, or "orphan" sequences
- Related sequences that differ due to large N or C terminal extensions or large internal insertions or deletions

Benchmark challenges



Source: BaliBase, Thompson et al, NAR, 1999,

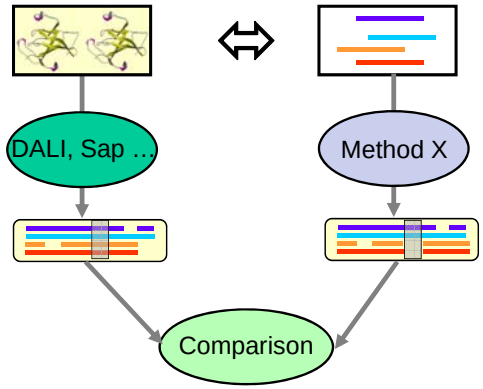
Mathematical correctness is not a guarantee of biological accuracy. The performance of MSA programs is typically evaluated using benchmarks based on biological data:

- Curated structural alignment
- Automated structural alignment
- Real or simulated sequence

Various benchmarks are designed to mimic properties of different types of data sets encountered in practice, especially those that are challenging to align:

- Highly divergent sequences, e.g., <50% or <30% identity
- A family of related sequences plus several outliers, or "orphan" sequences
- Related sequences that differ due to large N or C terminal extensions or large internal insertions or deletions

BaliBase: Reference MSAs based on structural alignment.



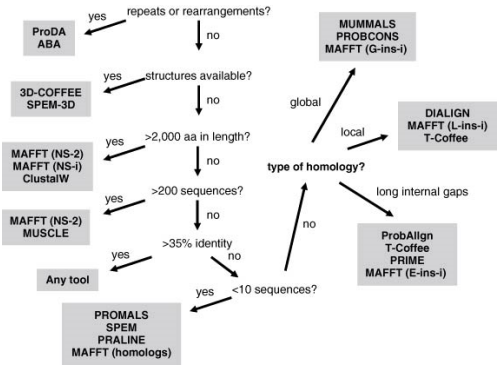
Note that implementation choices result in substantial differences in running time:

Aligner	Performance*	Time
DIALIGN	57.2	12 h, 25 min
CLUSTALW	58.9	2 h, 57 min
T-Coffee	63.6	***144 h, 51 min***
MUSCLE	64.8	3 h, 11 min
MAFFT	64.8	***2h,36min***
ProbCons	66.9	19 h, 41 min
ProbCons-ext	***68.0***	37 h, 46 min

* Fraction of correctly aligned residue pairs

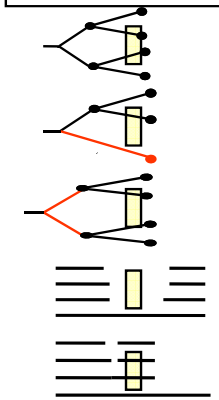
Do et al, Genome Research, 2005

Which program to choose?



Do and Katoh, 2008

PROBLEM	Description



Even Phylogenic Spread.

One Outlayer Sequence

Two Distantly related Groups

Long Internal Indel

Long Terminal Indel

Source: BaliBase, Thompson et al, NAR, 1999,

Approaches for improving MSA (Speed or accuracy)

- Iterative refinement of the MSA
- Faster estimation of the guide tree
- Better scoring
  - Combining information from various sources
  - Consistency in alignments of 3 sequences
  - Weighting sequences pairs
- Position specific gap penalties

Iterative refinement

Progressive "alignment suffers from its greediness"
Notredame et al, JMB 2000

1. Randomly select one sequence, remove it and realign it with the rest of the alignment
2. Remove each sequence in turn and realign with the remaining alignment. Select the best of these as the new alignment.
3. Randomly split into two sub alignments and realign them.

Apply strategy repeatedly until convergence or out of computer time

Approaches for improving MSA (Speed or accuracy)

- Iterative refinement of the MSA
- Faster estimation of the guide tree
- Better scoring
  - Combining information from various sources
  - Consistency in alignments of 3 sequences
  - Weighting sequences pairs
- Position specific gap penalties

Combining information from multiple sources

T. Coffee, Notredame, Higgins, Heringa, JMB 2000

```
DRHNSNIKV
DLKPENLLI
```

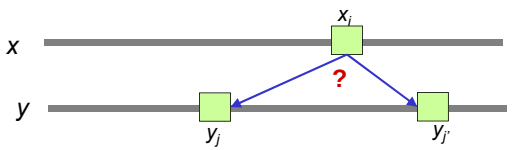
22% identity

```
DRHNSNIKVDDG_QLFHIDFGHFLD
YLHSLDIYRDLKPENLIDQQGYIQV
```

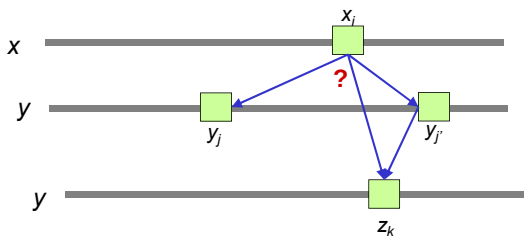
12% identity

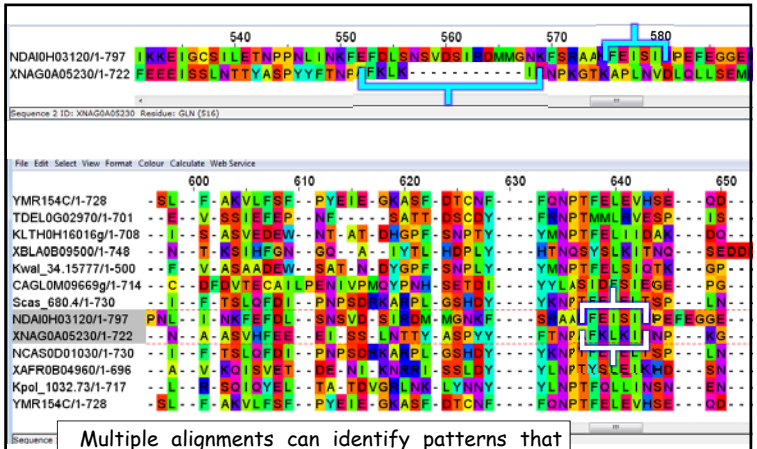
Construct a *library* of pairwise alignments

Consistency



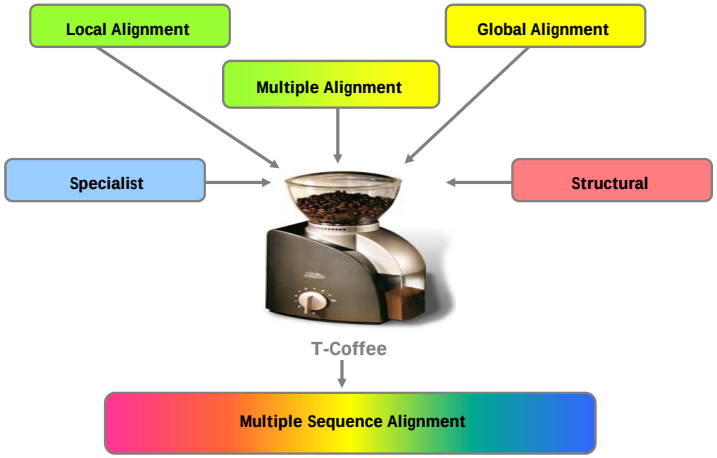
Consistency





Multiple alignments can identify patterns that are conserved in a family but not apparent in a pairwise alignment of two family members.

Combining information from multiple sources:

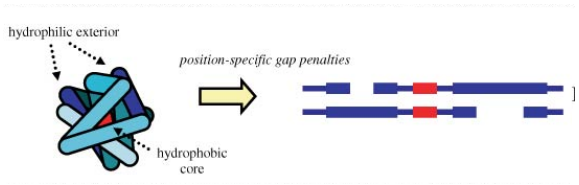


Approaches for improving MSA (Speed or accuracy)

- Iterative refinement of the MSA
- Faster estimation of the guide tree
- Better scoring
  - Combining information from various sources
  - Consistency in alignments of 3 sequences
  - Weighting sequences pairs
- Position specific gap penalties

Position specific gap penalties

Penalize gaps in hydrophobic and hydrophillic regions differently



Do and Katoh, 2008

Other improvements

- Sequence weighting

Assign weights so that **these sequences** do not dominate.

Do and Katoh, 2008