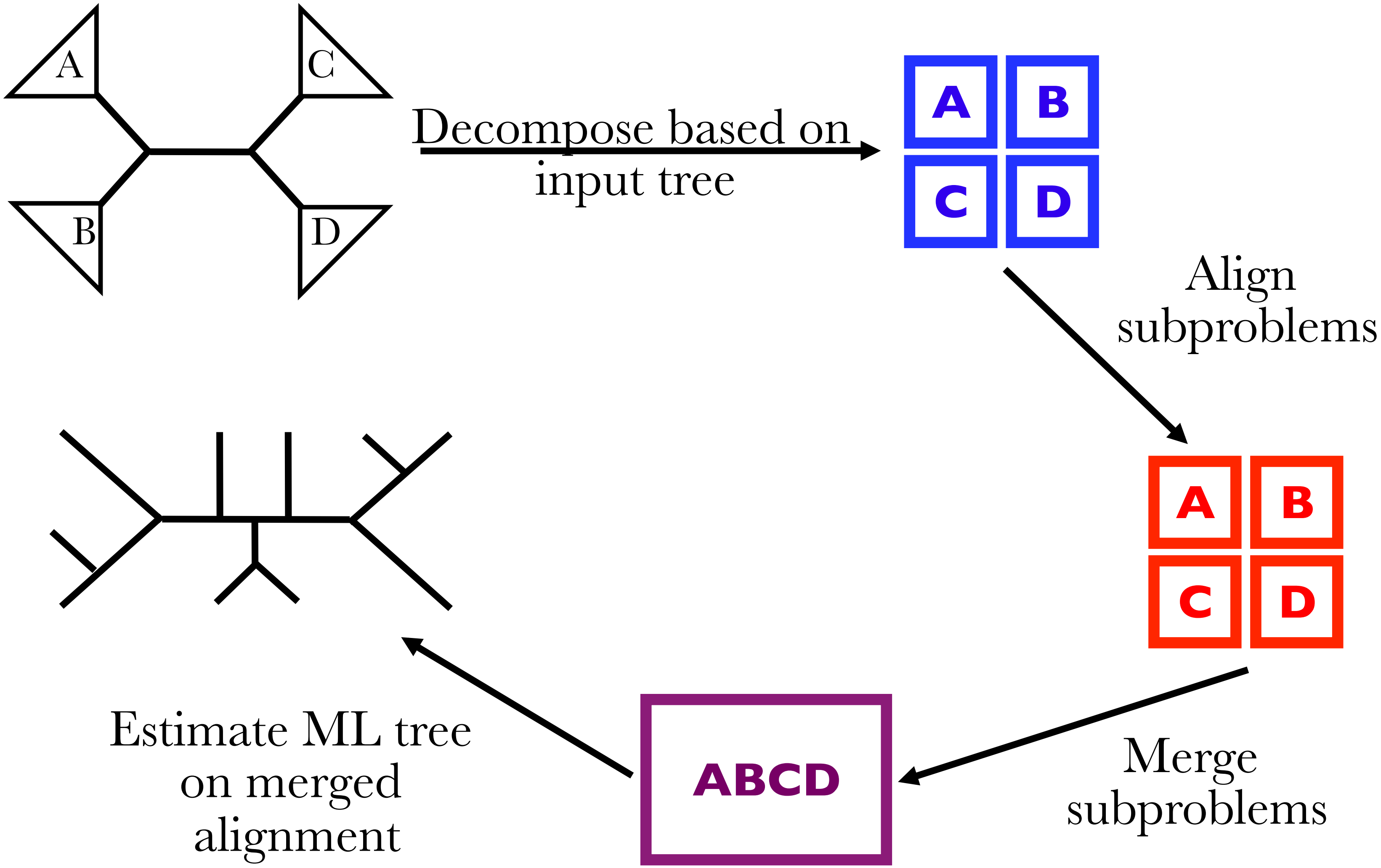
Multiple Alignment

a talk given by Mark Holder at the
Molecular evolution workshop 2015 at the MBL
Woods Hole MA

butchered by Peter Beerli
(removed some slides to gain time)

SATé repeats the following steps until termination



Decompose based on input tree

Align subproblems

Merge subproblems

Estimate ML tree on merged alignment

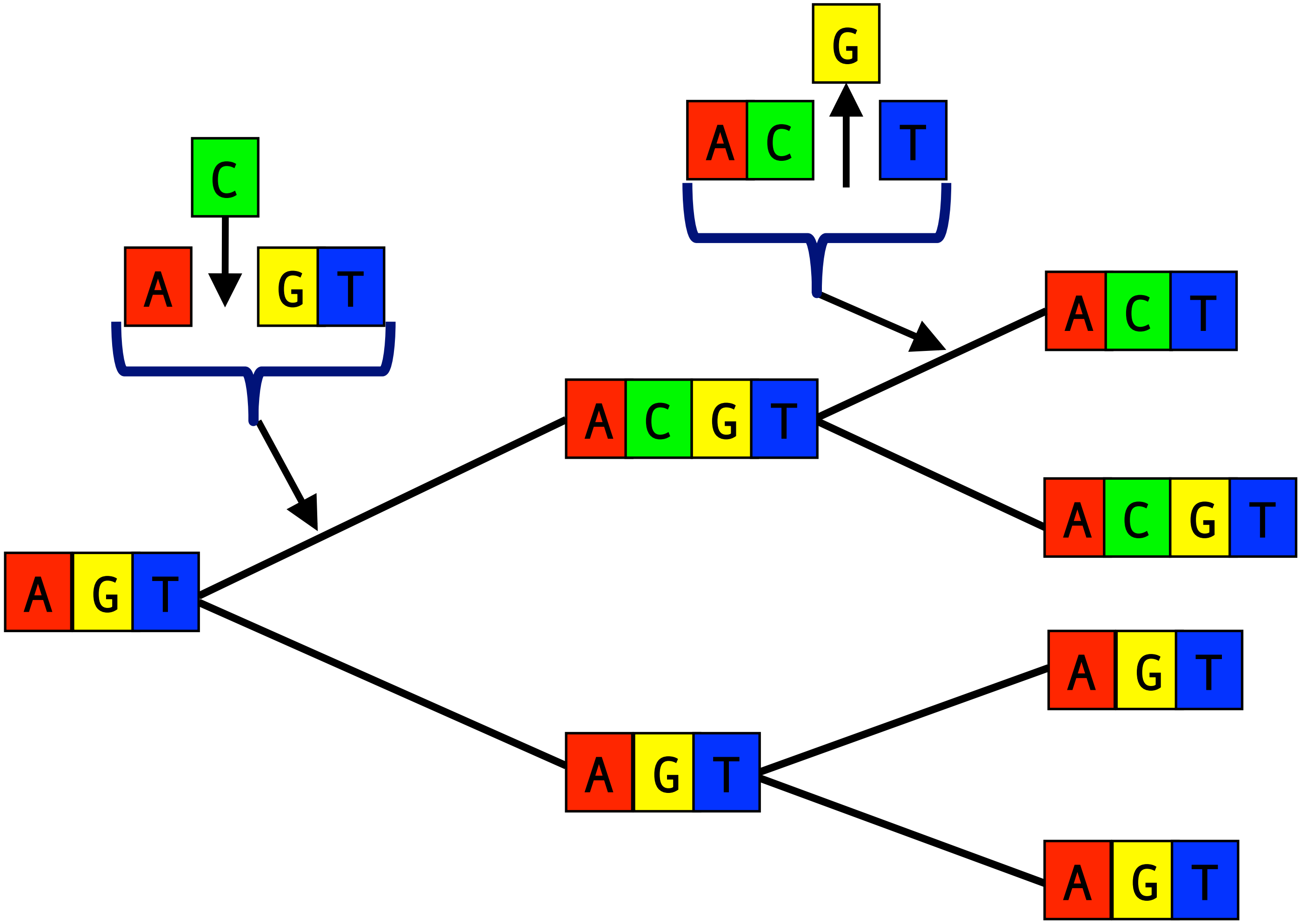
Woods Hole 2015 - brief primer on Multiple Sequence Alignment – Mark Holder

Many forms of sequence alignment are used in bioinformatics:

- Structural Alignment
- Local alignments
- **Global, evolutionary alignment**
  - Inputs: unaligned sequences thought to be homologous over their full length
  - we often ignore events like transpositions or inversions

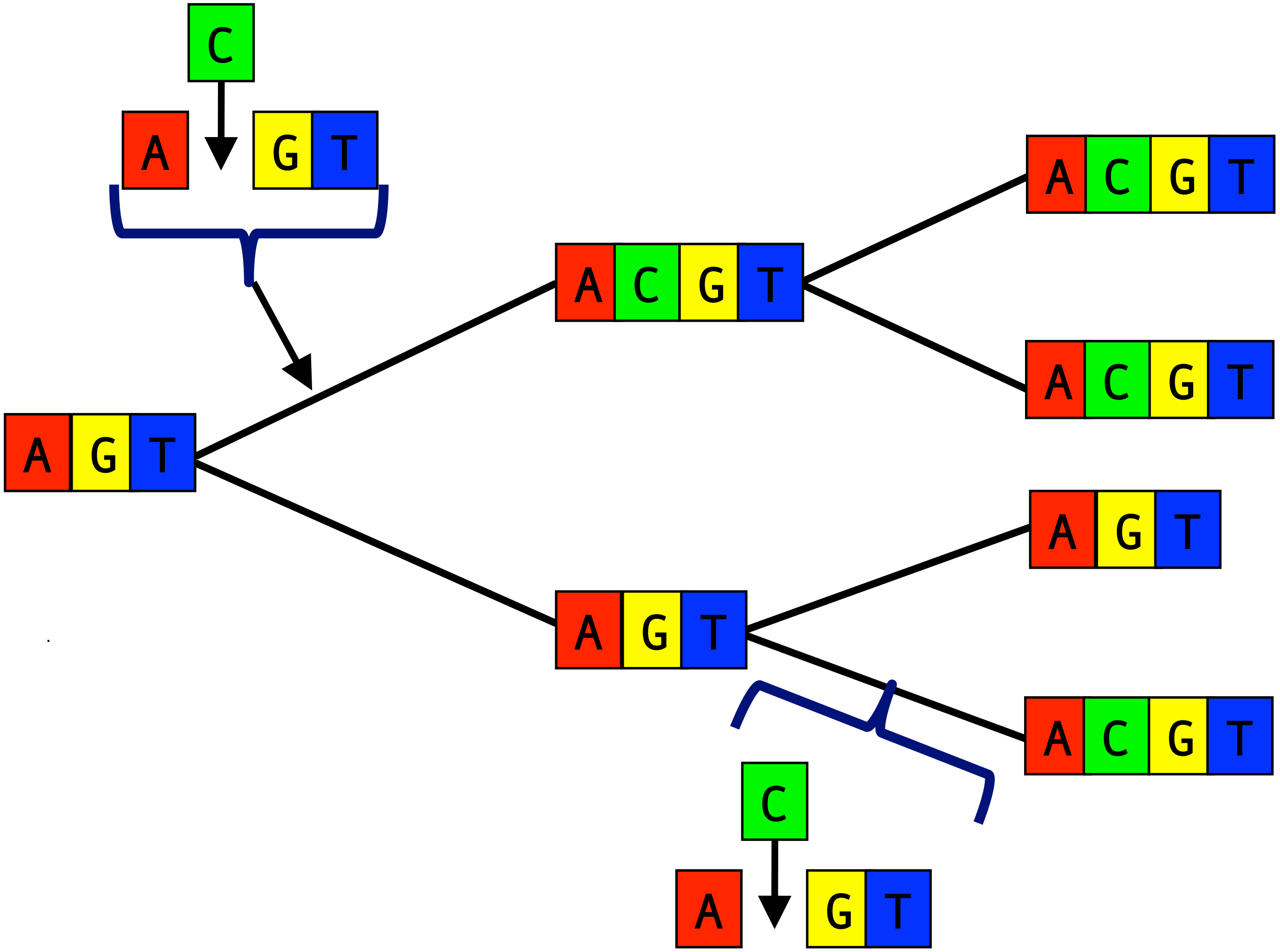
The goal of MSA is to introduce gaps such that:

- residues in the same column are homologous – all descended from a residue in the common ancestor, and
- all descendants of a residue are put in the same column.



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

slide by Derrick Zwickl



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

slide by Derrick Zwickl

Main points

- Accuracy of alignment is important.
- Pairwise alignment is tractable.
- Most MSA programs use progressive alignment:
  - a series of pairwise operations.
  - these algorithms are **not** guaranteed to return the optimal solution.
  - the criteria used are not ideal from an evolutionary standpoint.

Main points (continued)

- Simultaneous inference of MSA and tree is the most appropriate choice, but is computationally demanding. See: Poisson Indel Process (Bouchard-Côté and Jordan, 2013), Bali-Phy, Handel, AliFritz, and POY software
- Many people cull ambiguously aligned regions.

human	KRSV
chimp	KRV
gorilla	KSV
orang	KPRV

How we align these sequences affects tree estimation.

human	KRSV		human	KRSV		human	KRSV
chimp	KR-V		chimp	K-RV		chimp	KR-V
gorilla	KS-V		gorilla	K-SV		gorilla	K-SV
orang	KPRV		orang	KPRV		orang	KPRV

Pairwise alignment

Gap penalties and a substitution matrix imply a score for any alignment. Pairwise alignment involves finding the alignment that maximizes this score.

- substitution matrices assign positive values to matches or substitutions between similar residues (for example Leucine→Isoleucine).
- infrequent types of substitutions receive negative scores
- indels are rare, so gaps are heavily penalized (negative scores).

second base in codon

first base in codon	T	C	A	G	third base in codon
T	TTT Phe	TCT Ser	TAT Tyr	TGT Cys	T
	TTC Phe	TCC Ser	TAC Tyr	TGC Cys	C
	TTA Leu	TCA Ser	TAA stop	TGA stop	A
	TTG Leu	TCG Ser	TAG stop	TGG Trp	G
C	CTT Leu	CCT Pro	CAT His	CGT Arg	T
	CTC Leu	CCC Pro	CAC His	CGC Arg	C
	CTA Leu	CCA Pro	CAA Gln	CGA Arg	A
	CTG Leu	CCG Pro	CAG Gln	CGG Arg	G
A	ATT Ile	ACT Thr	AAT Asn	AGT Ser	T
	ATC Ile	ACC Thr	AAC Asn	AGC Ser	C
	ATA Ile	ACA Thr	AAA Lys	AGA Arg	A
	ATG Met	ACG Thr	AAG Lys	AGG Arg	G
G	GTT Val	GCT Ala	GAT Asp	GGT Gly	T
	GTC Val	GCC Ala	GAC Asp	GGC Gly	C
	GTA Val	GCA Ala	GAA Glu	GGA Gly	A
	GTG Val	GCG Ala	GAG Glu	GGG Gly	G

BLOSUM 62 Substitution matrix

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	−1	5																		
N	−2	0	6																	
D	−2	−2	1	6																
C	0	−3	−3	−3	9															
Q	−1	1	0	0	−3	5														
E	−1	0	0	2	−4	2	5													
G	0	−2	0	−1	−3	−2	−2	6												
H	−2	0	1	−1	−3	0	0	−2	8											
I	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
L	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
K	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
M	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
F	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
P	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
S	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
T	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
W	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Y	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
V	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4
	A	**R**	**N**	**D**	**C**	**Q**	**E**	**G**	**H**	**I**	**L**	**K**	**M**	**F**	**P**	**S**	**T**	**W**	**Y**	**V**

Scoring an alignment with the BLOSUM 62 matrix

Pongo	V	D	E	V	G	G	E	L	G	R	L	F	V	V	P	T	Q
Gorilla	V	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	2	-2	0	6	-6	-3	-4	-2	-2	4	0	4	-1	7	4	1

The score for the alignment is

$$D_{ij} = \sum_k d_{ij}^{(k)}$$

If i indicates *Pongo* and j indicates *Gorilla*. (k) is just an index for the column.

$$D_{ij} = 12$$

Scoring an alignment with gaps

If we were to use a gap penalty of -8:

Pongo	V	D	E	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	V	-	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1

By introducing gaps we have improved the score:

$$D_{ij} = 40$$

Gap Penalties

Penalizing gaps more heavily than substitutions avoids alignments like this:

```
Pongo    VDEVGGE-LGRLFVVPTQ
Gorilla  VDEVGG-DLGRLFVVPTQ
```

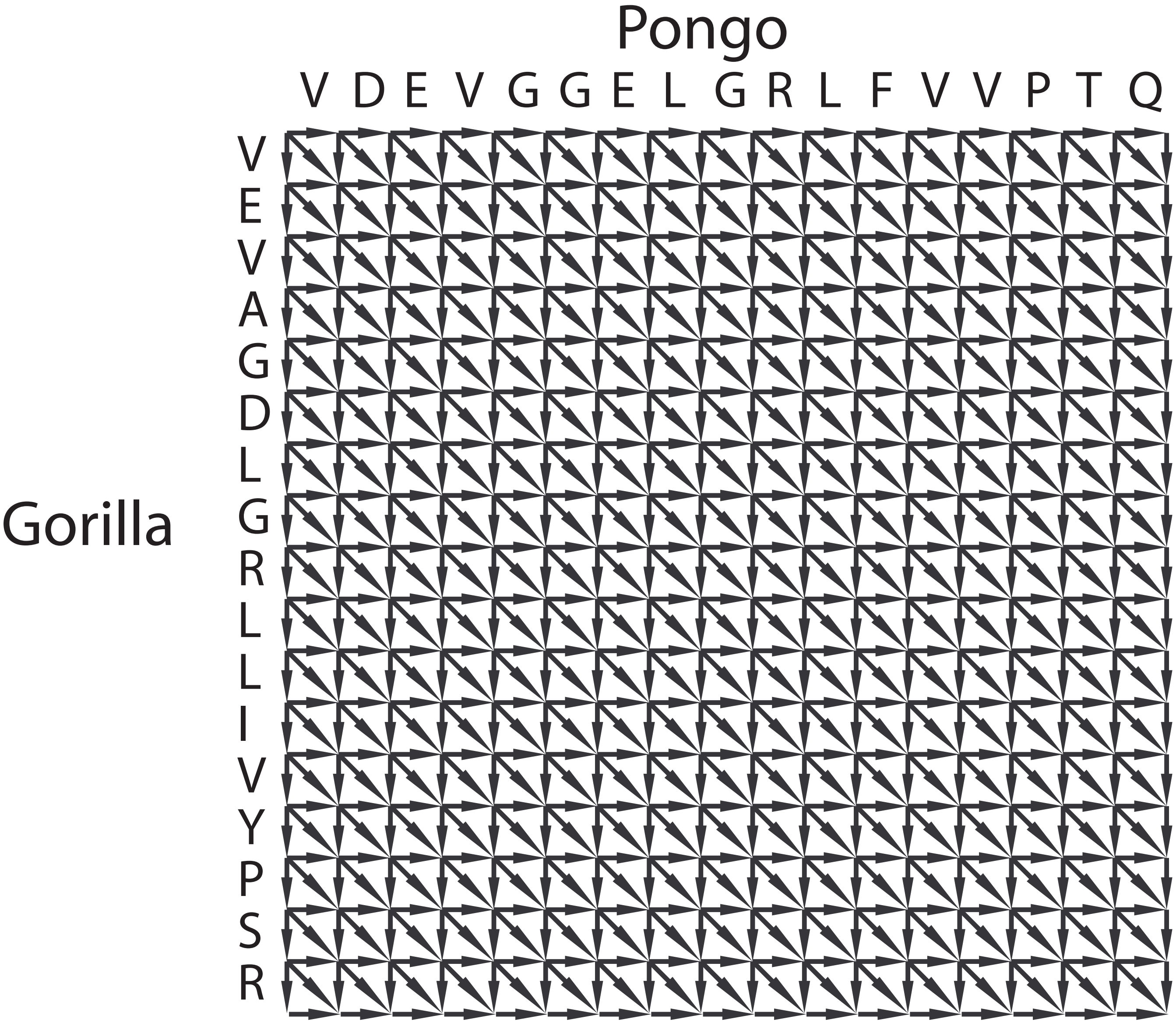
Affine gap penalties are often used to accommodate multi-site indels:

$$\text{GP} = \text{GO} + (l)\text{GE}$$

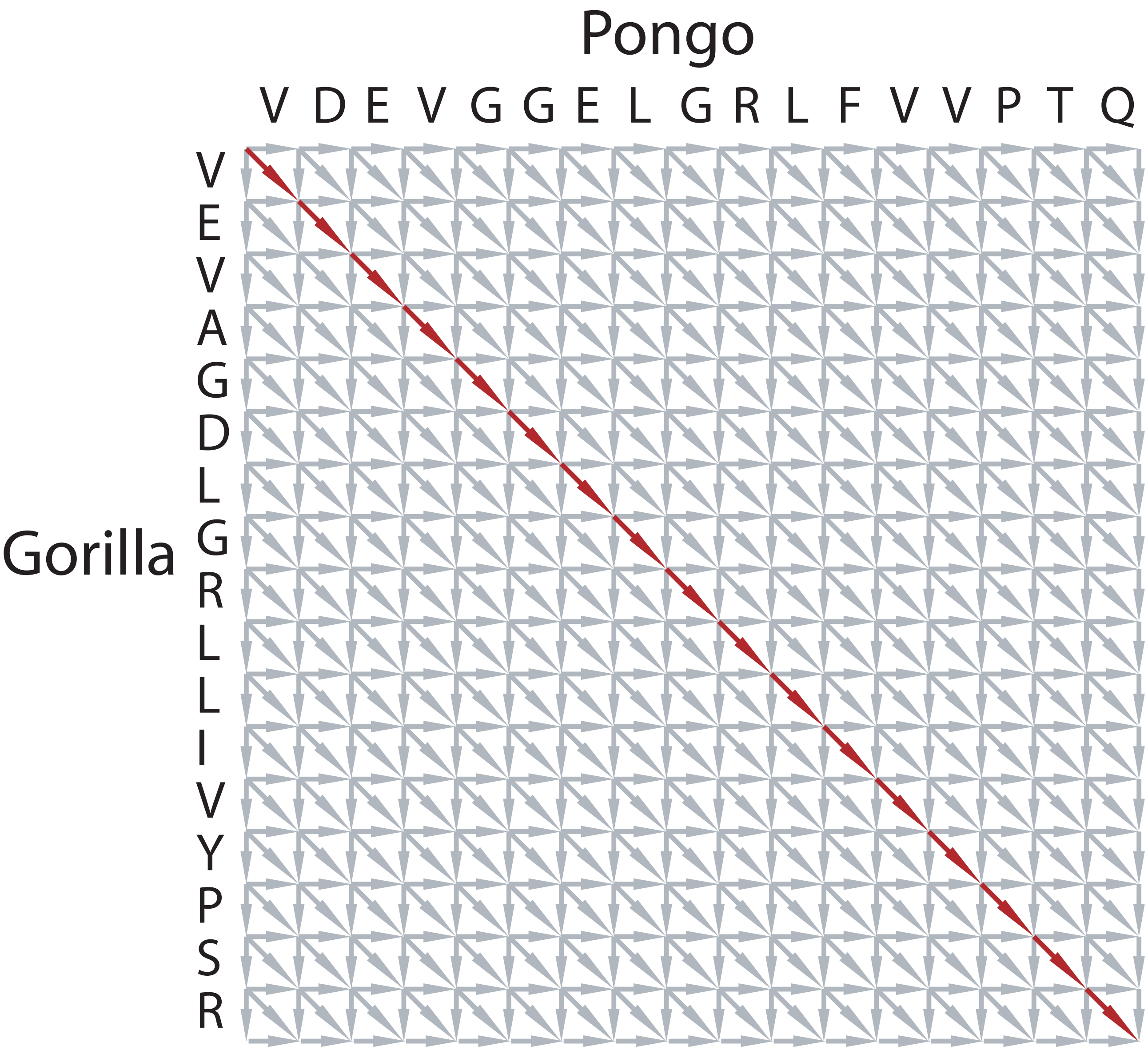
where:

- GP is the gap penalty.
- GO is the “gap-opening penalty”
- GE is the “gap-extension penalty”
- l is the length of the gap

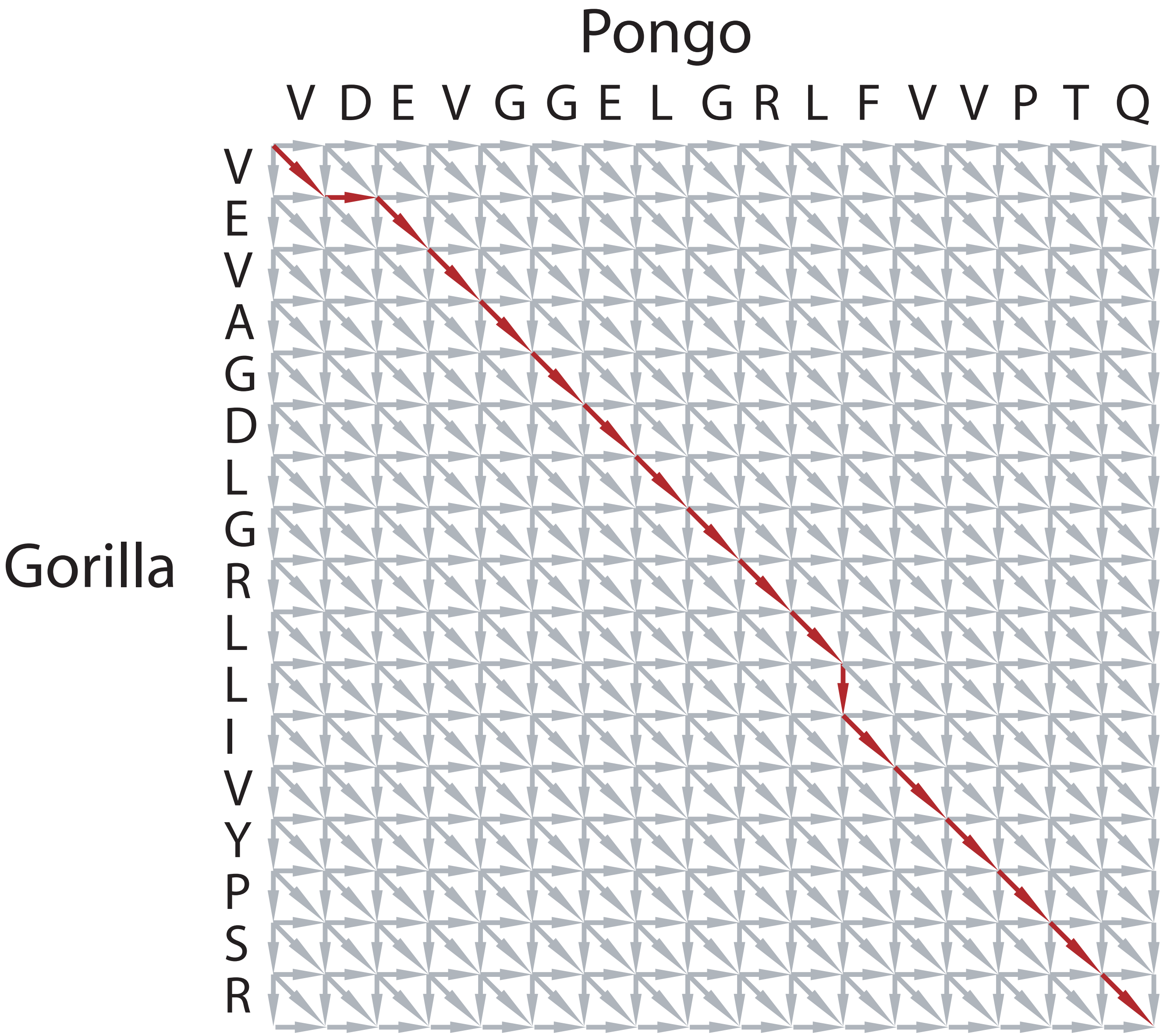
Finding an optimal alignment



pongo	V	D	E	V	G	G	E	L	G	R	L	F	V	V	P	T	Q
gorilla	V	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
score	4	2	-2	0	6	-6	-3	-4	-2	-2	4	0	4	-1	7	4	1



Pongo	V	D	E	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	V	-	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1



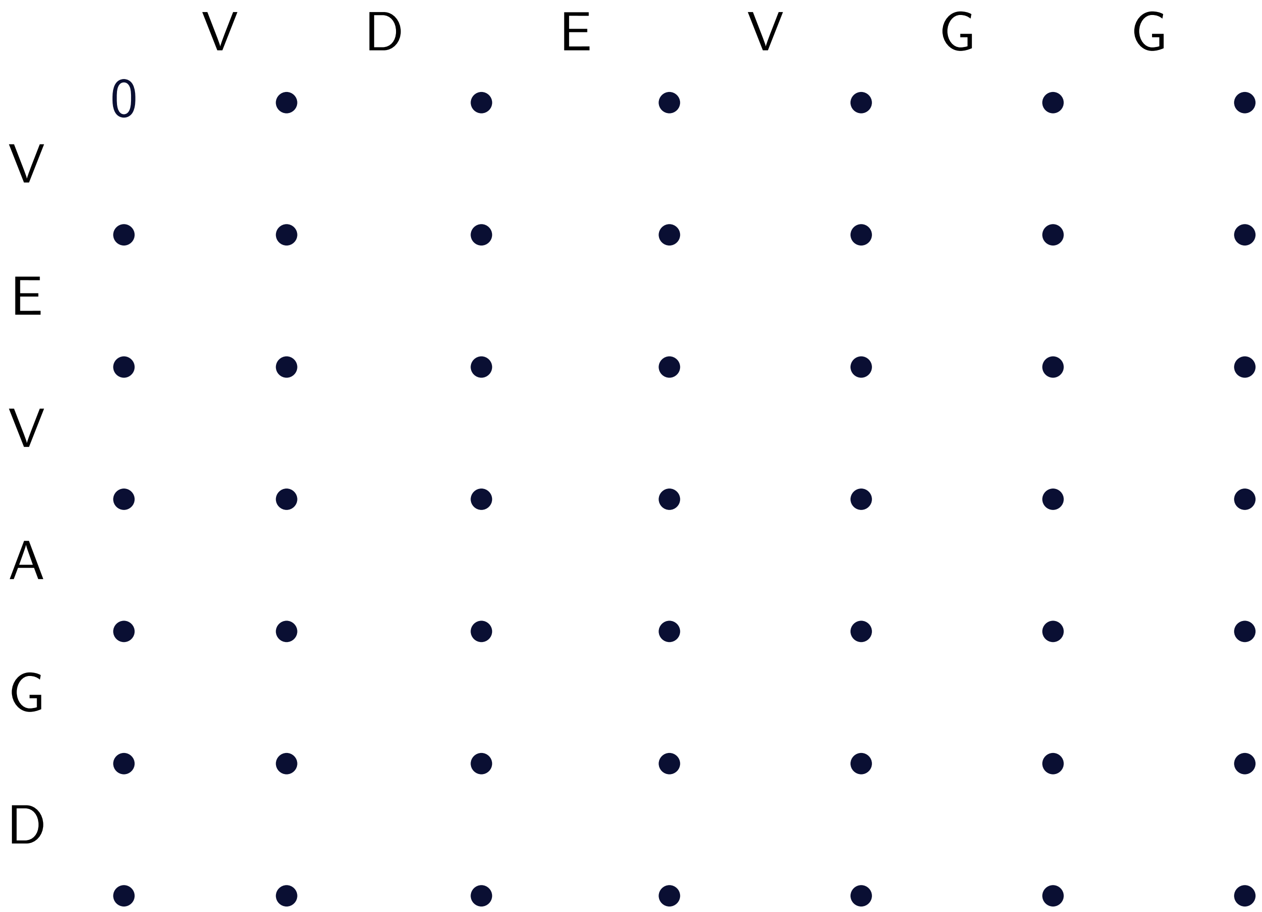
Needleman-Wunsch algorithm (paraphrased)

- Work from the top left (beginning of both sequences)
- For each cell store the highest score possible for that cell and a "back" pointer to tell point to the previous step in the best path
- When you reach the lower right corner, you know the optimal score and the back pointers tell you the alignment.

The highest score calculation at each cell only depends on the cell's 3 possible previous neighbors.

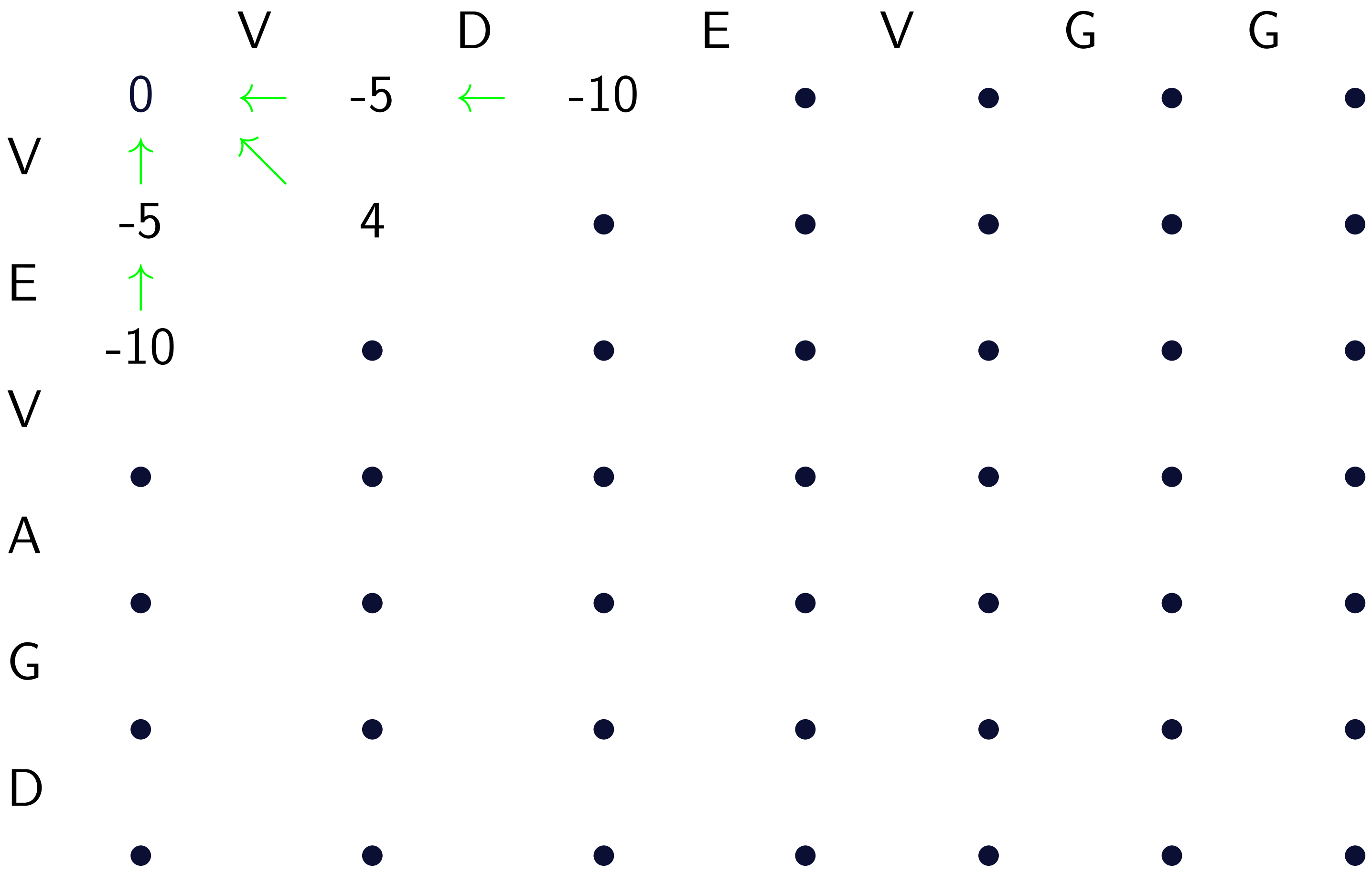
If one sequence is length M_1, and the other is length M_2, then Needleman-Wunsch only takes $\mathcal{O}(M_1 M_2)$ calculations.

But there are a **much** larger number of possible alignments (# alignments grows exponentially).

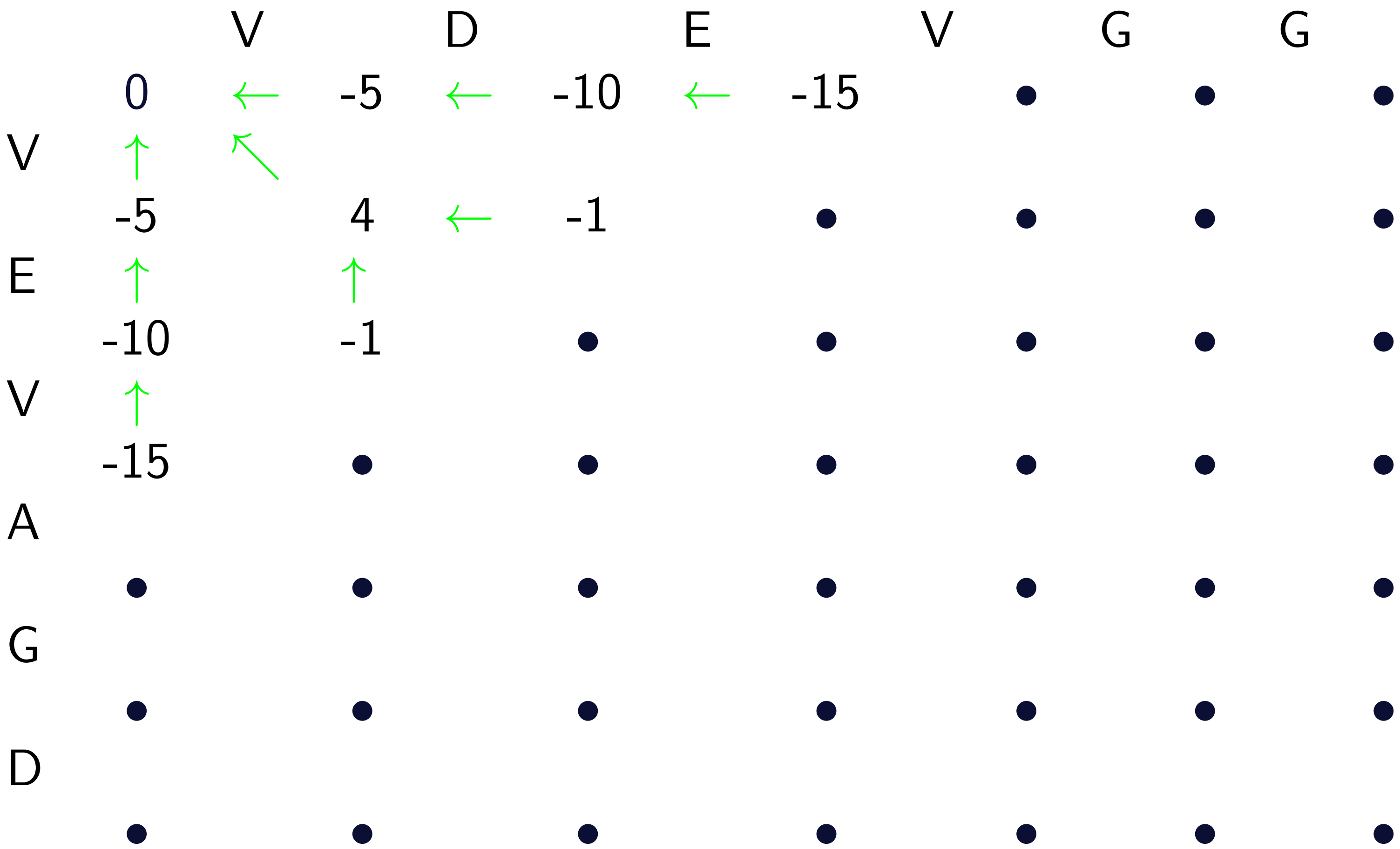

V
D
E
V
G
G
0
V
E
V
A
G
D

Using a gap penalty of -5

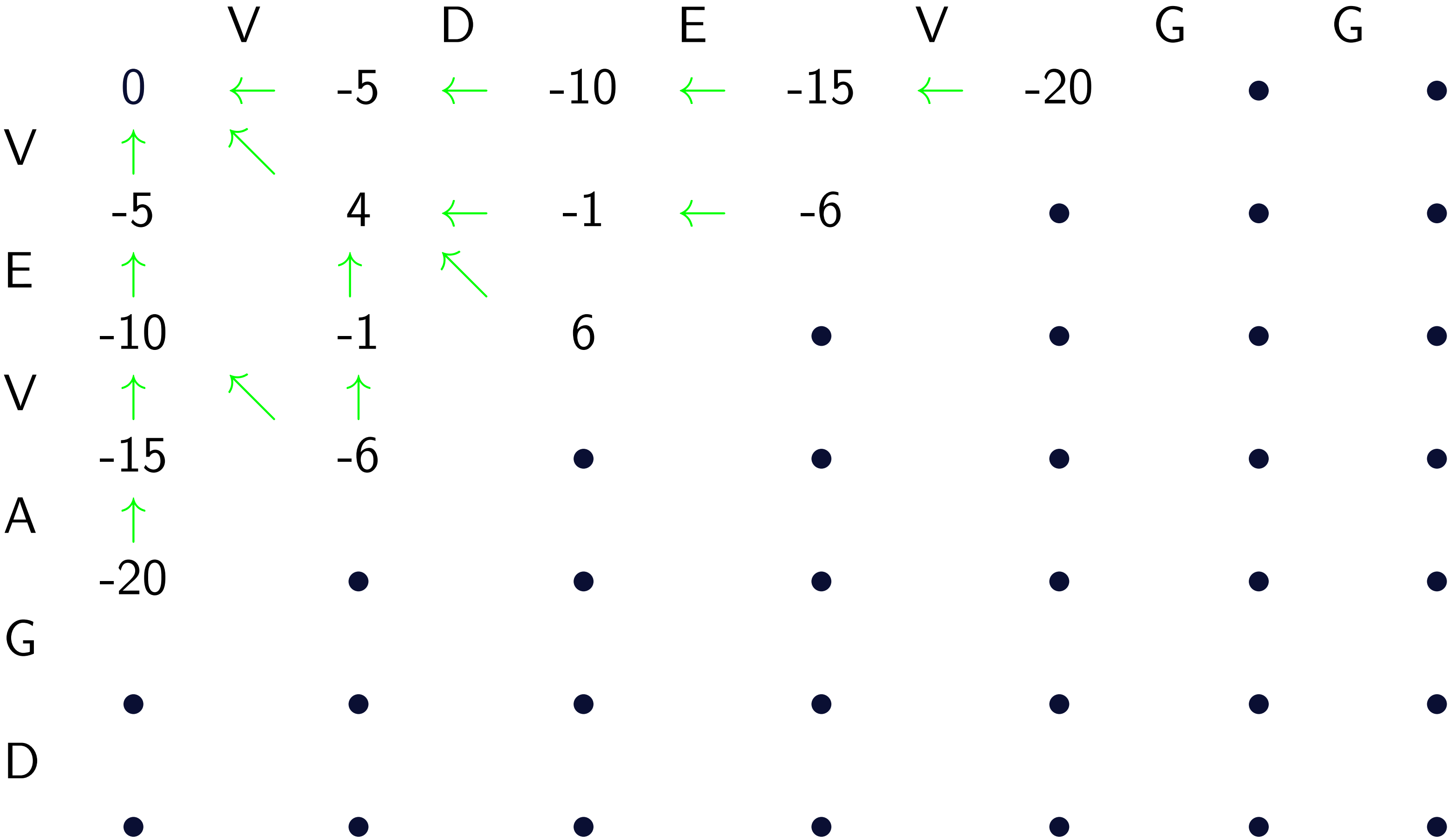
		V	D	E	V	G	G
	0	← -5	●	●	●	●	●
V	↑ -5	●	●	●	●	●	●
E	●	●	●	●	●	●	●
V	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●
G	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●



		V	D	E	V	G	G
	0	← -5	← -10	●	●	●	●
V	↑ -5	↖ 4	●	●	●	●	●
E	↑ -10	●	●	●	●	●	●
V	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●
G	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●



		V	D	E	V	G	G
	0	-5	-10	-15	●	●	●
V	-5	4	-1	●	●	●	●
E	-10	-1	●	●	●	●	●
V	-15	●	●	●	●	●	●
A	●	●	●	●	●	●	●
G	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●



		V	D	E	V	G	G
	0	← -5	← -10	← -15	← -20	●	●
V	↑ -5	↖ 4	← -1	← -6	●	●	●
E	↑ -10	↑ -1	↖ 6	●	●	●	●
V	↑ -15	↖ ↑ -6	●	●	●	●	●
A	↑ -20	●	●	●	●	●	●
G	●	●	●	●	●	●	●
D	●	●	●	●	●	●	●

Pairwise alignment is a beautiful topic in bioinformatics

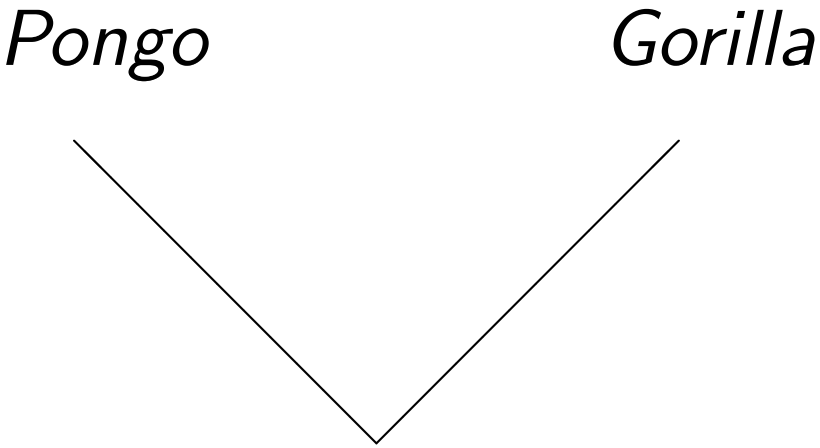
- Clever programming tricks let us find the best-scoring pairwise alignment quickly

- The additive scoring system:
  - can incorporate biological knowledge (via empirically-based substitution matrices)
  - can almost be justified in terms of powerful statistical methodology (maximum likelihood).

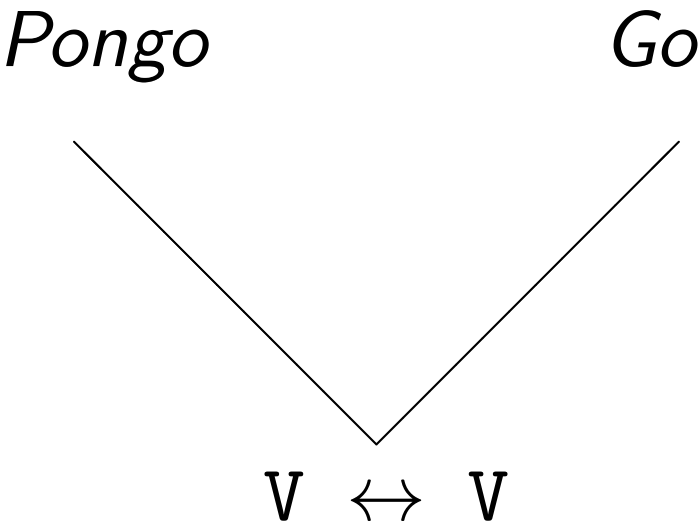
Pairwise alignment costs

- Paul Lewis will explain likelihood tomorrow,
- Additive costs can be justified as approximations to the log of likelihoods if:
  - we can identify the events that must have occurred in generate the data, and
  - we can assign (relative) probabilities based on whether these events are rare or common.

Pongo	V	D	E	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	V	-	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1

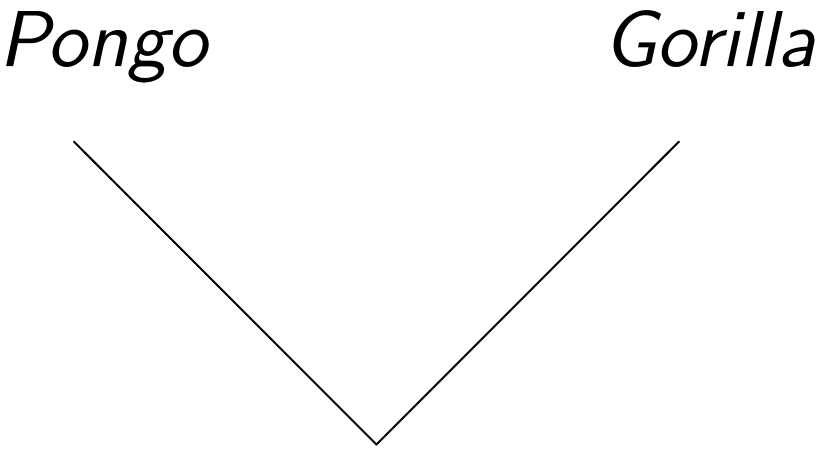


Pongo	**V**	D	E	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	**V**	-	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1



$$\mathbb{P}(\text{pos. } 1) = \mathbb{P}(V \leftrightarrow V)$$

Pongo	**V**	**D**	E	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	**V**	-	E	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1

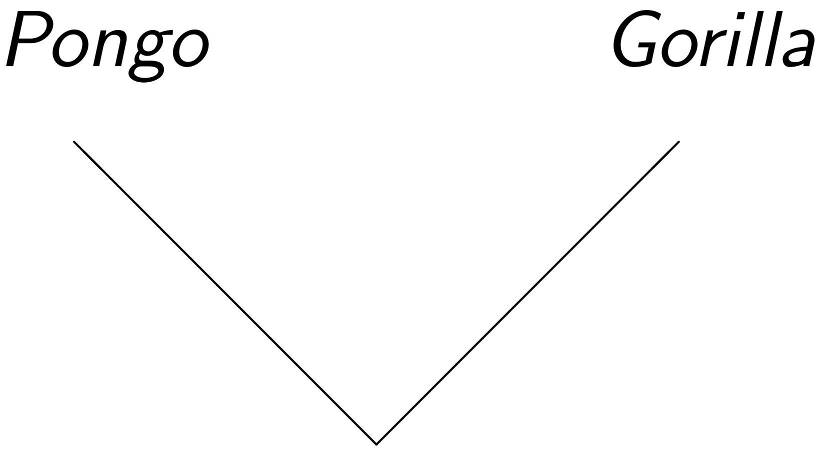


Pongo *Gorilla*

$$\begin{array}{lcl} \texttt{V} & \leftrightarrow & \texttt{V} \\ \texttt{D} & \leftrightarrow & \texttt{-} \end{array}$$

$$\mathbb{P}(\text{pos. } 1-2) = \mathbb{P}(V \leftrightarrow V)$$
$$\times \mathbb{P}(D \leftrightarrow -)$$

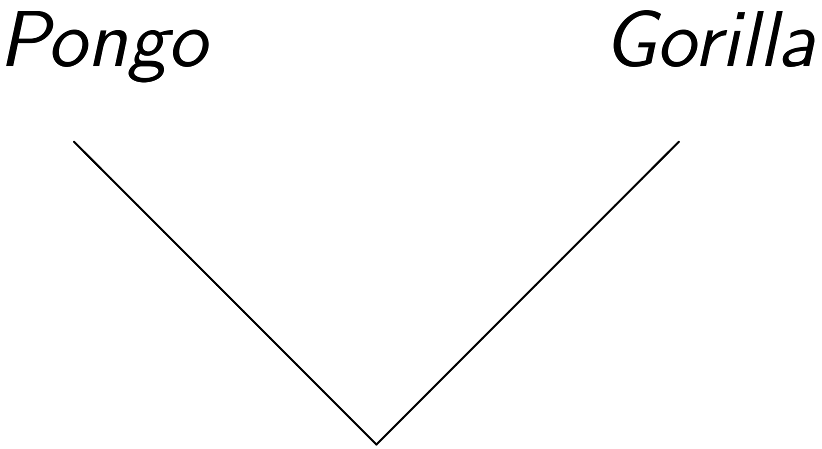
Pongo	**V**	**D**	**E**	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	**V**	-	**E**	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1



`V` $\leftrightarrow$ `V`
`D` $\leftrightarrow$ `-`
`E` $\leftrightarrow$ `E`

$$\begin{aligned} \mathbb{P}(\text{pos. } 1-3) = \mathbb{P}(V \leftrightarrow V) \\ \times\mathbb{P}(D \leftrightarrow -) \\ \times\mathbb{P}(E \leftrightarrow E) \end{aligned}$$

Pongo	**V**	**D**	**E**	V	G	G	E	L	G	R	L	-	F	V	V	P	T	Q
Gorilla	**V**	-	**E**	V	A	G	D	L	G	R	L	L	I	V	Y	P	S	R
Score	4	-8	5	5	0	6	2	4	6	5	4	-8	0	4	-1	7	4	1



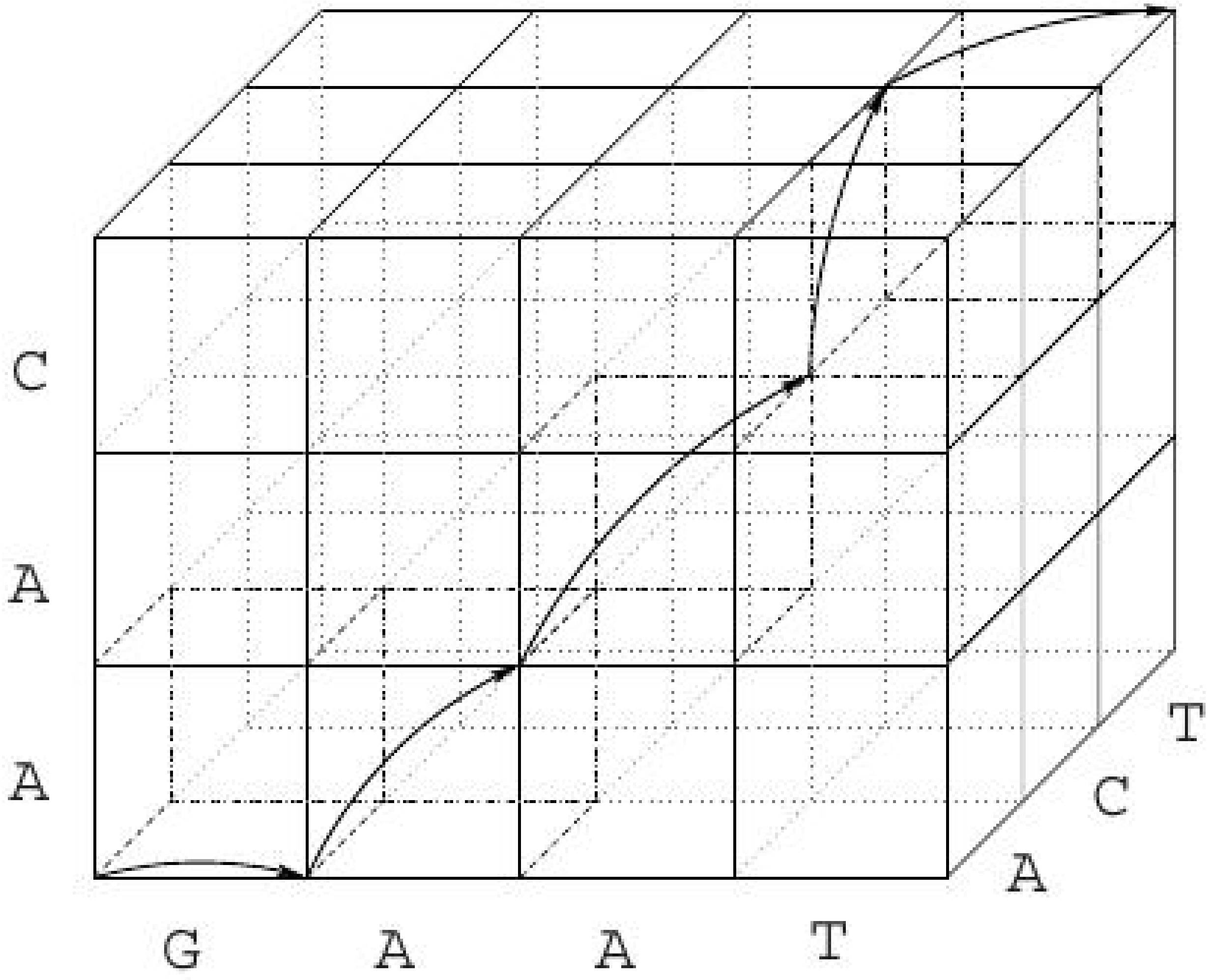
$$\begin{array}{ccc} \mathtt{V} & \leftrightarrow & \mathtt{V} \\ \mathtt{D} & \leftrightarrow & \mathtt{-} \\ \mathtt{E} & \leftrightarrow & \mathtt{E} \end{array}$$

$$\begin{aligned} \ln \mathbb{P}(\text{pos. } 1-3) = \ln \mathbb{P}(V \leftrightarrow V) \\ + \ln \mathbb{P}(D \leftrightarrow -) \\ + \ln \mathbb{P}(E \leftrightarrow E) \end{aligned}$$

Pairwise alignment summary

- The sum of the substitution and gap cost can serve as a proxy for the log-likelihood under a reasonable model (the **LR** or **AIF** model of Rivas and Eddy, preprint).
- Dynamic programming can let us find the alignment that has the highest likelihood.

from (Rausch and Reinert, 2011)



```
GAA-T
-AAC-
--ACT
```

Multiple sequence alignment is an ugly topic in bioinformatics

- Clever programming tricks help, but we still have to rely on *heuristics* – approaches that provide good solutions, but are not guaranteed to find the best solution.

- The additive scoring system suffers from the fact that we do not observe ancestral sequences.

- U is the set of unaligned sequences
- T is the genealogy tree that describes the ancestry of the sequences
- H is an indel history (specification of where all inserstions and deletions in the history of U occur on the tree T).
- A is an alignment of the sequences U

We might want:

- $\mathbb{P}(T \mid U)$ or $\mathbb{P}(A \mid U)$ or $\mathbb{P}(T, A \mid U)$. BaliPhy approximates these quantities, but it is tough to do for large datasets.

- $\hat{H}$ which maximizes $\mathbb{P}(H \mid U, T)$. ProtPal (Westesson et al., 2012) approximates this (but we have to know T)

Sum of Pairs scoring

Most MSA programs optimize a fairly strange score:

$$SP = \sum_i \sum_j w_{ij} d(A_i, A_j)$$

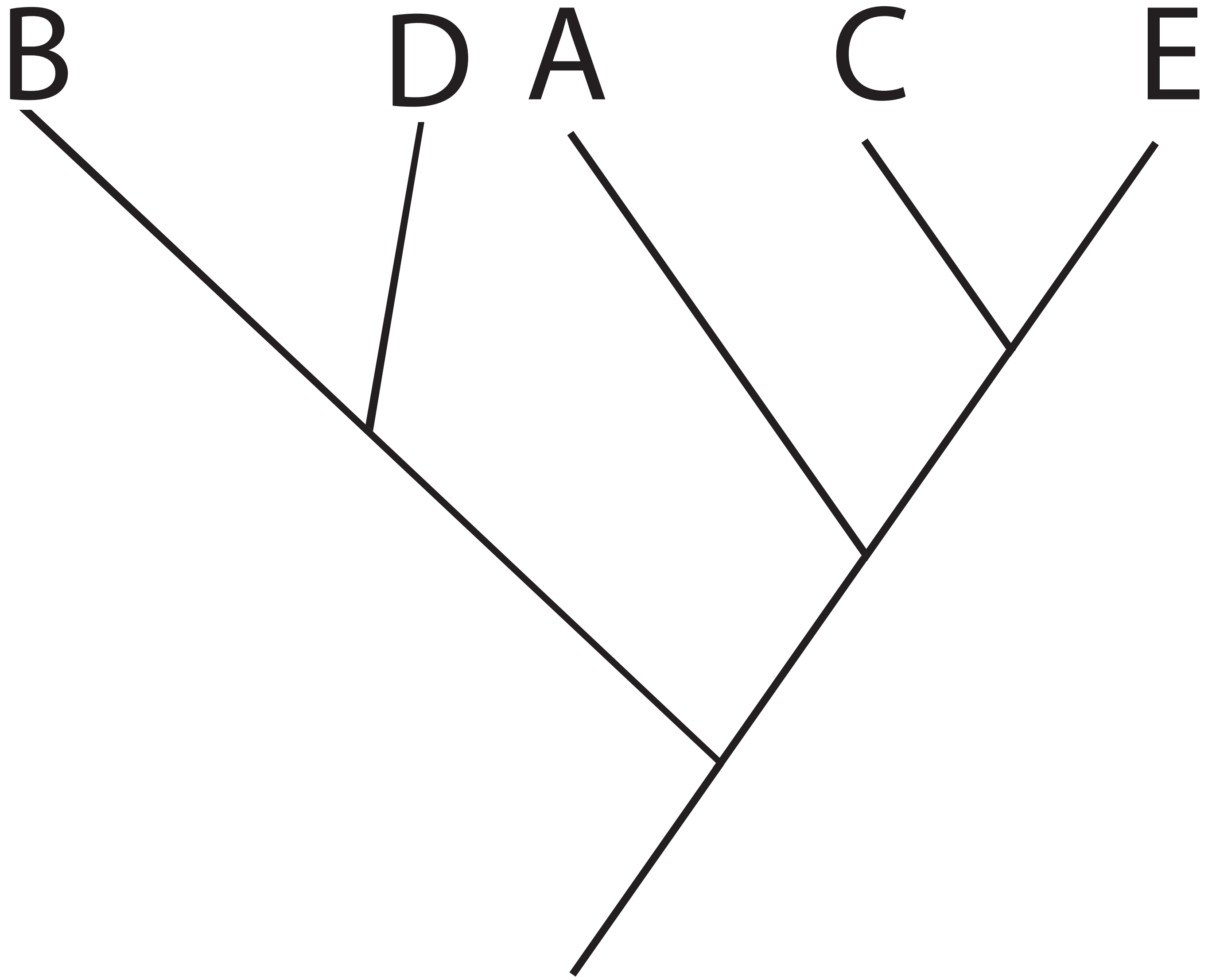
where A_i is the alignment pruned down to just sequence i, and w_{ij} is a weight for the comparison between sequences i and j.

$d(A_i, A_j)$ can be:

- a measure of the distance from A_i to A_j or
- a measure of the consistency of the alignment A with a pairwise alignment of i to j.

We usually cannot guarantee that we have found the alignment that optimizes the sum of pairs score.

Aligning more than two sequences



Progressive alignment

An approximate method for producing multiple sequence alignments using a guide tree.

- Perform pairwise alignments to produce a distance matrix
- Produce a guide tree from the distances
- Use the guide tree to specify the ordering used for aligning sequences, closest to furthest.

Feng and Doolittle 1987 and Higgins and Sharp, 1988

```
A   PEEKSAVTALWGKVNVDEVGG
B   GEEKAAVLALWDKVNEEEVGG
C   PADKTNVKAAWGKVGAHAGEYGA
D   AADKTNVKAAWSKVGGHAGEYGA
E   EHEWQLVLHVWAKVEADVAGHGQ
```

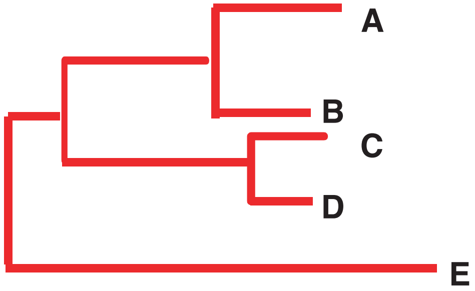
pairwise alignment

A	-				
B	.17	-			
C	.59	.60	-		
D	.59	.59	.13	-	
E	.77	.77	.75	.75	-

tree inference

```
A   PEEKSAVTALWGKVNVDEVGG
B   GEEKAAVLALWDKVNEEEVGG
C   PADKTNVKAAWGKVGAHAGEYGA
D   AADKTNVKAAWSKVGGHAGEYGA
E   EHEWQLVLHVWAKVEADVAGHGQ
```

+



alignment stage

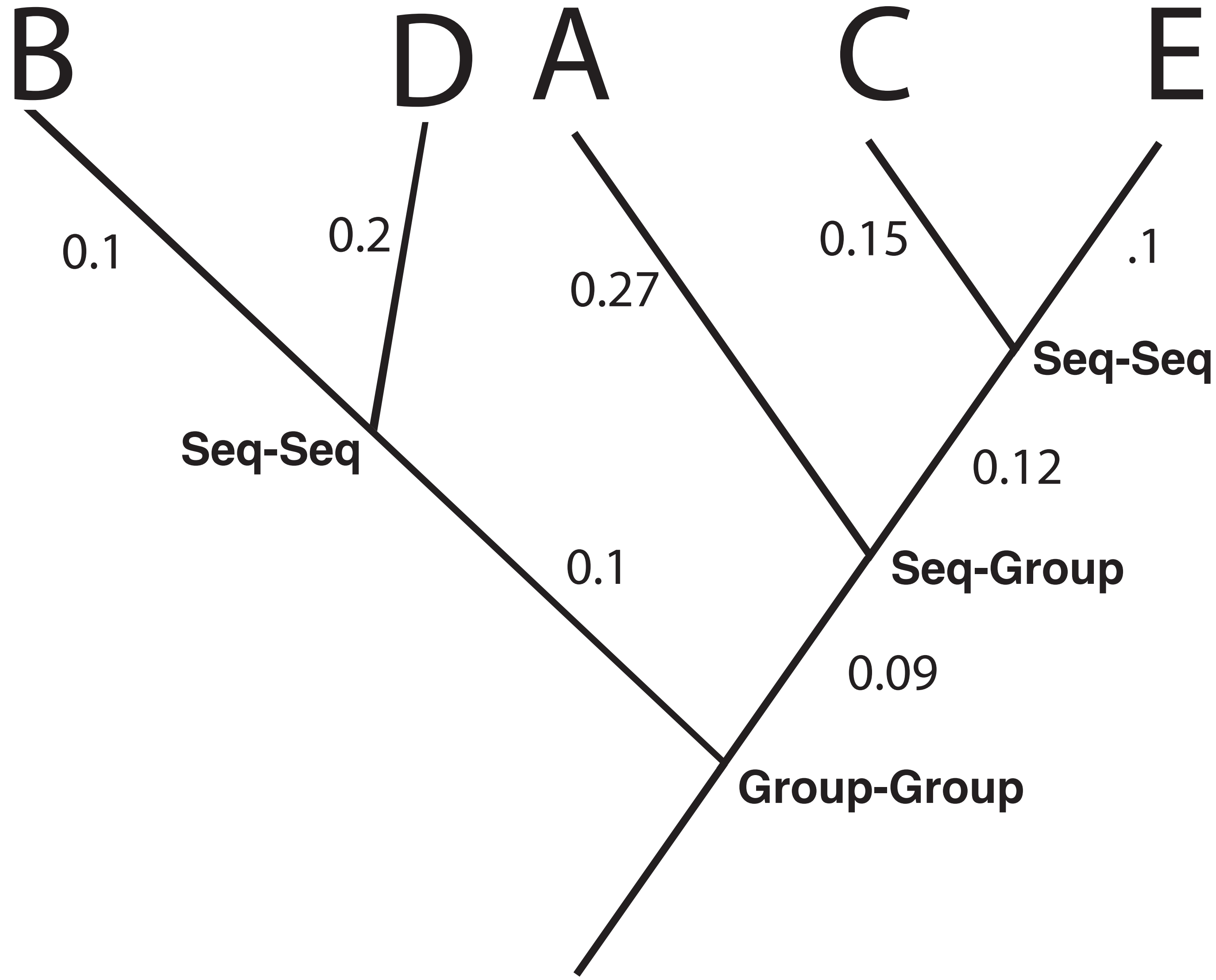
```
A   PEEKSAVTALWGKVN--VDEVGG
B   GEEKAAVLALWDKVN--EEEVGG
C   PADKTNVKAAWGKVGAHAGEYGA
D   AADKTNVKAAWSKVGGHAGEYGA
E   EHEWQLVLHVWAKVEADVAGHGQ
```

Alignment stage of progressive alignments

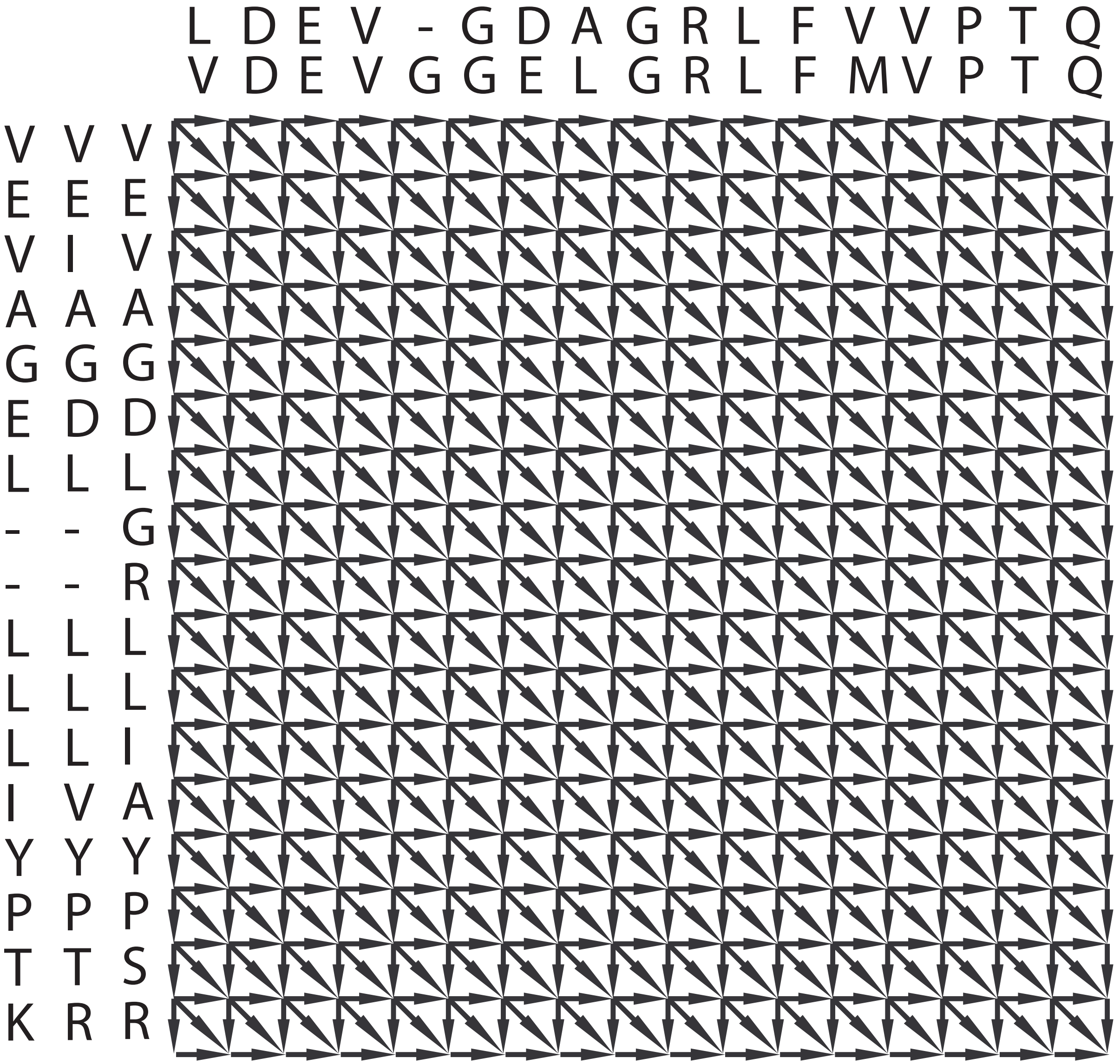
Sequences of clades become grouped as the algorithm descends the tree. Alignment at each step involves

- Sequence-Sequence,
- Sequence-Group, or
- Group-Group

Aligning multiple sequences

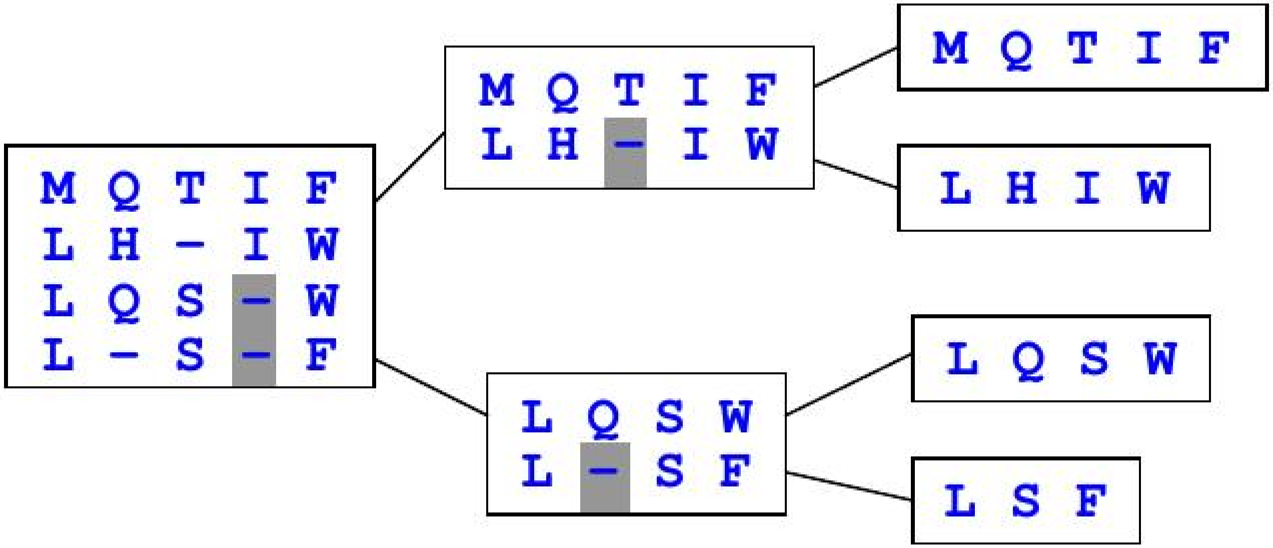


Group-to-Group alignment


L D E V - G D A G R L F V V P T Q
V D E V G G E L G R L F M V P T Q
V E V A G E L - - L L L I Y P T K
V E I A G D L - - L L L V Y P T R
V E V A G D L G R L L I A Y P S R

Group-to-group alignments

Adding a gap to a group means that every member of that group gets a gap at that position.



```
M Q T I F
L H - I W
L Q S - W
L - S - F
```

```
M Q T I F
L H - I W
```

```
M Q T I F
```

```
L H I W
```

```
L Q S W
L - S F
```

```
L Q S W
```

```
L S F
```

from Edgar (2004)

Group-to-group alignment

Usually the scores for each edge in the Needleman-Wunsch graph are calculated using a “sum of pairs” scoring system.

Many tools[1] uses weights assigned to each sequence in a group to down-weight closely related sequences so that they are not overrepresented - this is a weighted sum-of-pair scoring system.

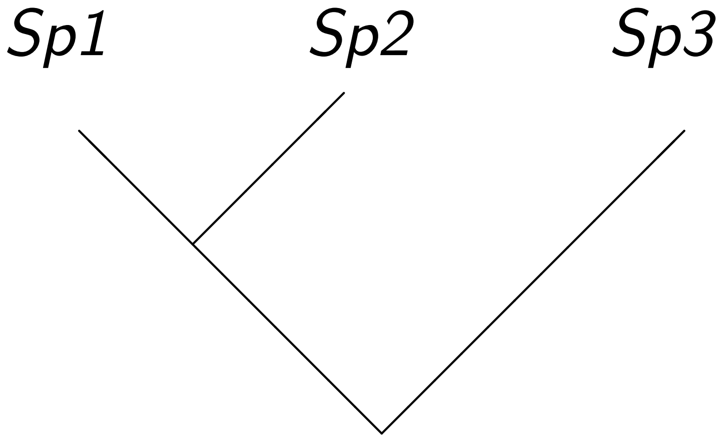
[1] e.g. Clustal and MAFFT

Greedy choices leading to failure to find the best alignment

Consider the scoring scheme:

match = 0   mismatch = -3   gap = -7

Guide Tree:



Sequences:

Sp1	`GACCGTG`
Sp2	`GCCGTAG`
Sp3	`GACCGTAG`

Greedy choices leading to failure to find the best alignment

match = 0 mismatch = -3 gap = -7

ungapped1*vs*2

Sp1	G	A	C	C	G	T	G		
Sp2	G	C	C	G	T	A	G		
Score	0	−3	0	−3	−3	−3	0	**Total=**	−12

would be preferred over gapped1*vs*2:

Sp1	G	A	C	C	G	T	-	G		
Sp2	G	-	C	C	G	T	A	G		
Score	0	−7	0	0	0	0	−7	0	**Total=**	−14

Adding a *Sp3* to ungapped1*vs*2:

Sp1	G	-	A	C	C	G	T	G
Sp2	G	-	C	C	G	T	A	G
Sp3	G	A	C	C	G	T	A	G

This implies 1 indel, and 4 substitutions. Score = -19 *

If we had been able to use gapped1*vs*2 then we could have:

Sp1	G	A	C	C	G	T	-	G
Sp2	G	-	C	C	G	T	A	G
Sp3	G	A	C	C	G	T	A	G

score = -14 *

* = “sort of...”

Polishing (aka “iterative alignment” can correct some errors caused by greedy heuristics)

1. break the alignment into 2 groups of sequences (often by breaking an edge in the merge tree).

2. realign those 2 groups to each other

3. keep the realignment if it improves the score

Opal also uses random 3-group polishing.

Weighted sum of pairs scoring system

Assigns a score for a group-to-group by averaging (or summing) the scores all of the implied pairwise alignments.

Frequently $w_{ij} = w_i w_j$

Group 1

Seq	weight	AA
taxon A	0.3	V
taxon C	0.24	A
taxon E	0.19	I

Group 2

Seq	weight	AA
taxon B	0.15	V
taxon D	0.25	M

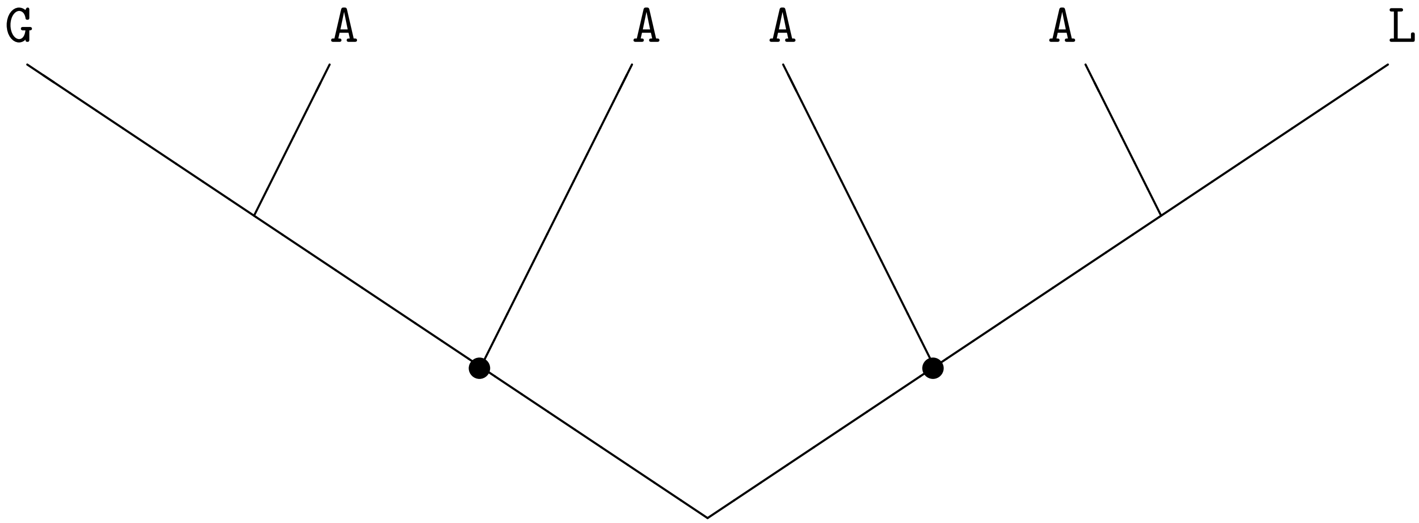
$$D_{G1,G2} \;=\; \frac{\sum_i \sum_j w_i w_j d_{ij}}{n_i n_j}$$

Group 1		
Seq	weight	AA
taxon A	0.3	V
taxon C	0.24	A
taxon E	0.19	I

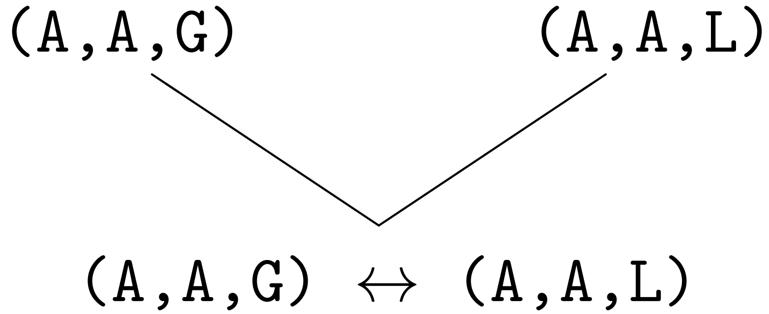
Group 2		
Seq	weight	AA
taxon B	0.15	V
taxon D	0.25	M

$$
\begin{aligned}
D_{G1,G2} &= \frac{\sum_i \sum_j w_i w_j d_{ij}}{n_i n_j} \\
&= \frac{1}{6}[d(V,V)w_A w_B + d(V,M)w_A w_D + d(A,V)w_C w_B \ldots \\
&\quad \ldots d(A,M)w_C w_D + d(I,V)w_E w_B + d(I,M)w_E w_D] \\
&= \frac{1}{6}(\mathbf{4} \times 0.3 \times 0.15 + \mathbf{1} \times 0.3 \times 0.25 + \mathbf{0} \times 0.24 \times 0.15 \ldots \\
&\quad \ldots -1 \times 0.24 \times 0.25 + \mathbf{3} \times 0.19 \times 0.15 + \mathbf{1} \times 0.19 \times 0.15) \\
&= 1.46225
\end{aligned}
$$

Sum-of-pairs is an odd scoring system



Imperfect scoring system. Consider one position in a group-to-group alignment:



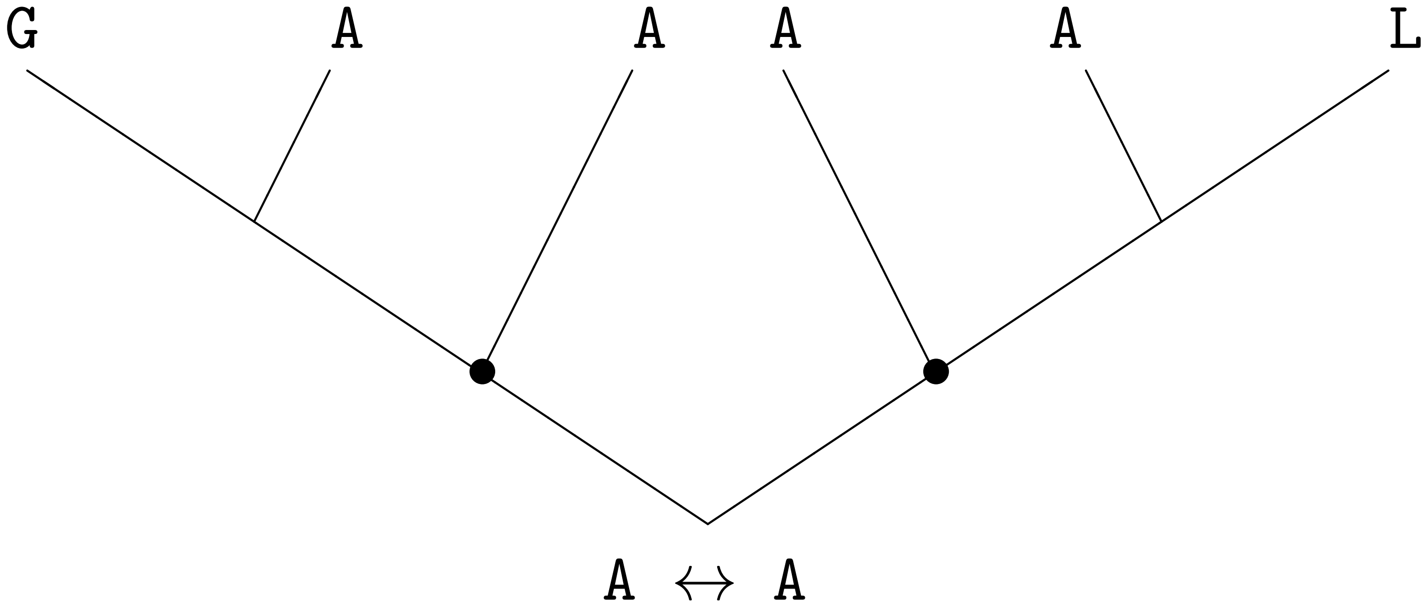
(A,A,G) (A,A,L)

(A,A,G) ↔ (A,A,L)

The sum-of-pairs score for aligning would be:

$$\frac{4}{9}(A \leftrightarrow A) + \frac{2}{9}(A \leftrightarrow L) + \frac{2}{9}(G \leftrightarrow A) + \frac{1}{9}(G \leftrightarrow L)$$

But in the context of the tree we might be pretty certain of an A↔A event



Note: weighted sum-of-pairs would help reflect the effect of ancestry better (but still not perfectly; sum-of-pairs techniques are simply not very sophisticated forms of ancestral sequence reconstruction).

“Consistency” based alignment

In our sum of pair scoring, we could just check how often each pair of sequences (one from G_h and one from G_v) display the same alignment that they did in pairwise alignment.

T-Coffee (Notredame et al., 2000) introduced the idea of performing group-to-group alignments during progressive alignment using this sense of consistency.

Indirect consistency arguments

In the pairwise alignments,

- *if* $h_{10} \sim g_{12}$
- *and* $p_{17} \sim g_{12}$
- *then* h_{10} should align with p_{17}

Probabilistic measures of “consistency”

The simplest assessment of the consistency of an MSA to pairwise alignments uses just the optimal pairwise alignment of each pair.

Opal (Wheeler and Kececioglu, 2007) uses some suboptimal alignments.

Do et al. (2005) made an important advance by proposing the use of the probability that two residues are aligned during consistency-based alignment (ProbCons).

The same sort of dynamic programming traversal of the alignment grid can give us a probability that 2 residues are aligned.

Sequence annealing

AMAP (Schwartz and Pachter, 2007) and FSA (Bradley et al., 2009) use a sequence annealing approach:

- start with trivial alignments (all residues opposite gaps),
- anchor regions by aligning long matches,
- merge columns as long as the score keeps improving.

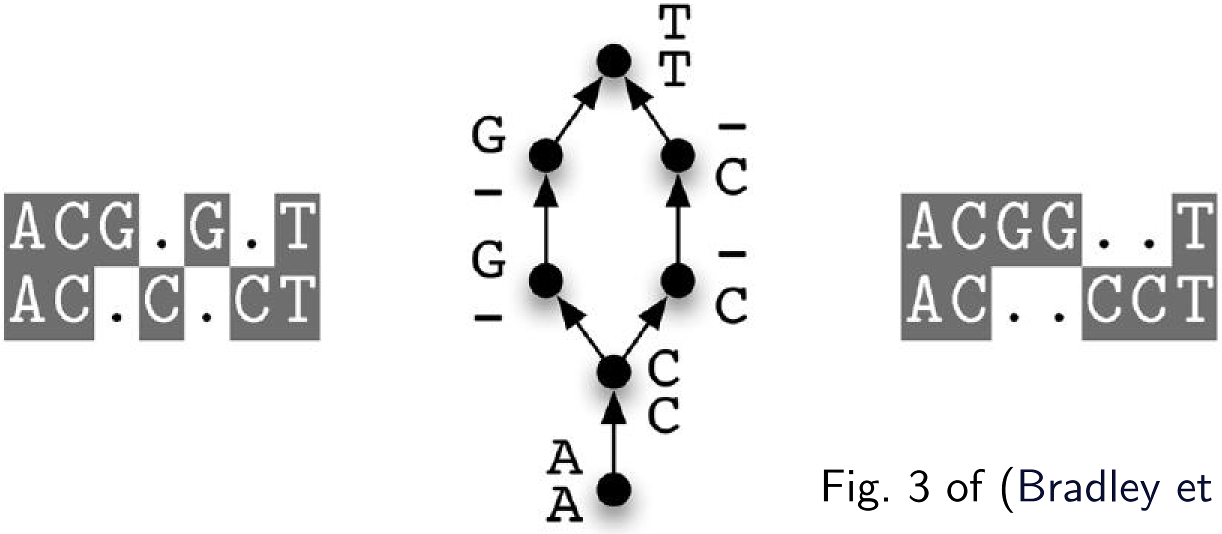


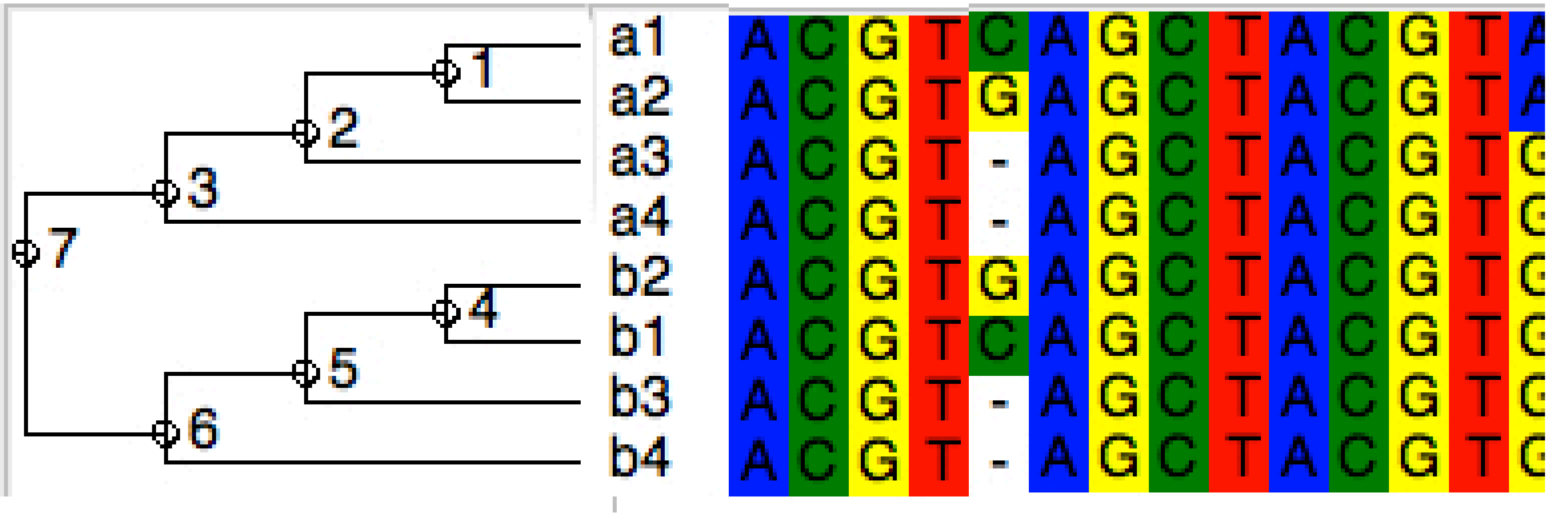
Fig. 3 of (Bradley et al., 2009)

Progressive alignment

- Uses a guide tree to change the MSA problem into a series of pairwise alignment problems;
- May not return the alignment with the best weighted sum-of-pairs scores. Early alignment decisions get "locked in" Most aligners try to polish the alignment, but we cannot guarantee that we have found the optimal alignment;
- Reconstruction of ancestral sequences is usually done in a quick-and-dirty, implicit fashion or is not done at all;

PRANK

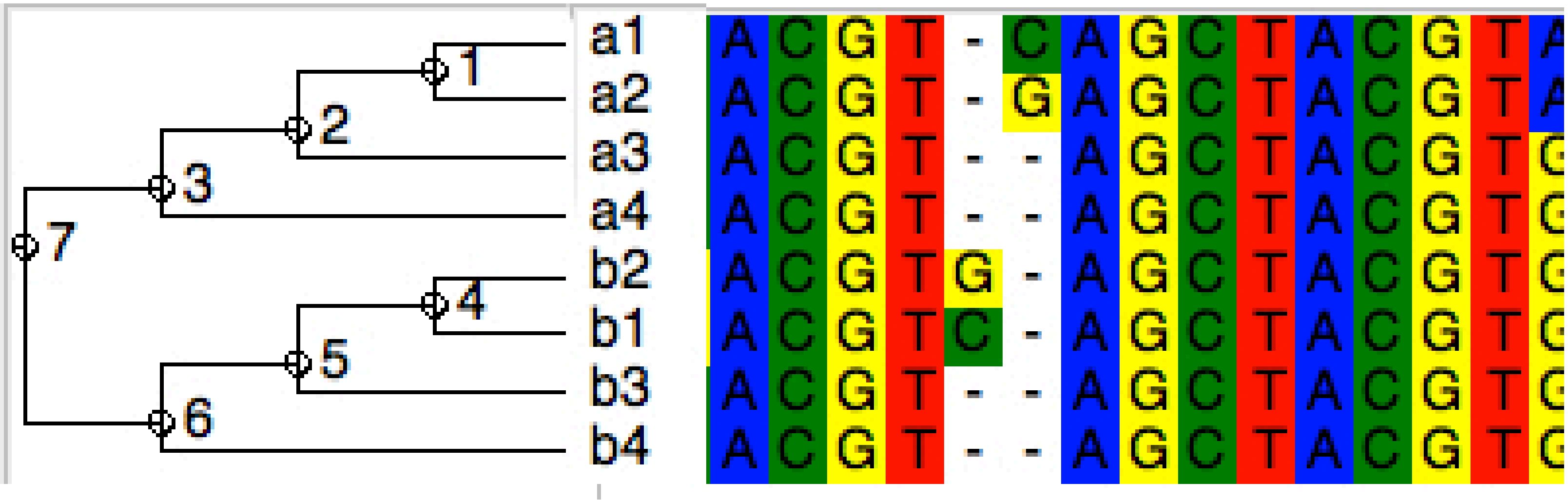
Löytynoja and Goldman (2005) showed most progressive alignment techniques were particularly prone to compression because of poor ancestral reconstruction:



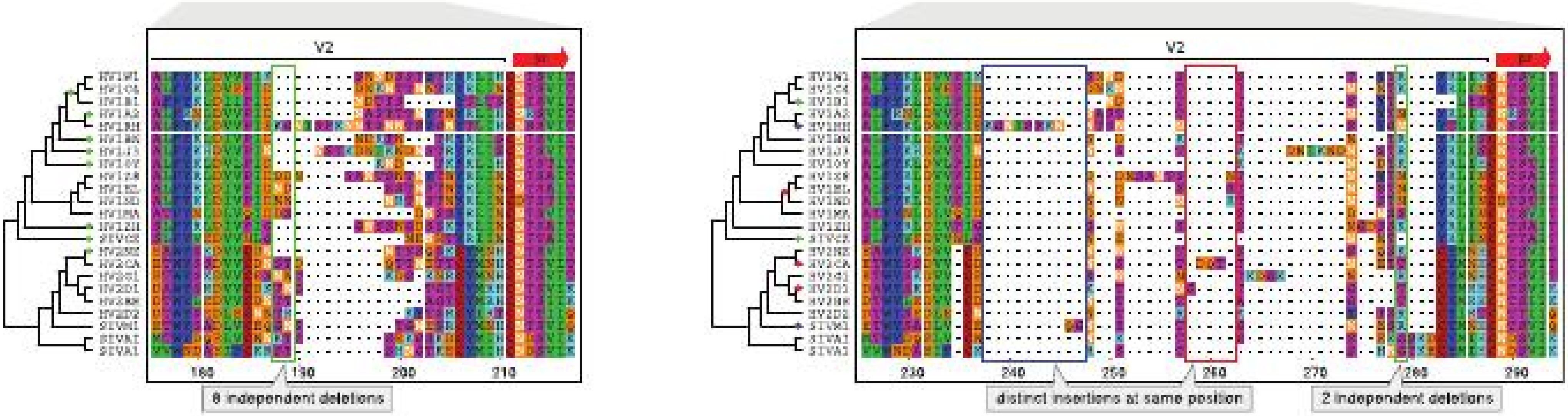
```
a1  A C G T C A G C T A C G T A
a2  A C G T G A G C T A C G T A
a3  A C G T - A G C T A C G T G
a4  A C G T - A G C T A C G T G
b2  A C G T G A G C T A C G T G
b1  A C G T C A G C T A C G T G
b3  A C G T - A G C T A C G T G
b4  A C G T - A G C T A C G T G
```

PRANK

Flagging inserted residues allows PRANK to effectively skip over these positions in the ancestor, producing more phylogenetically-sensible alignments:



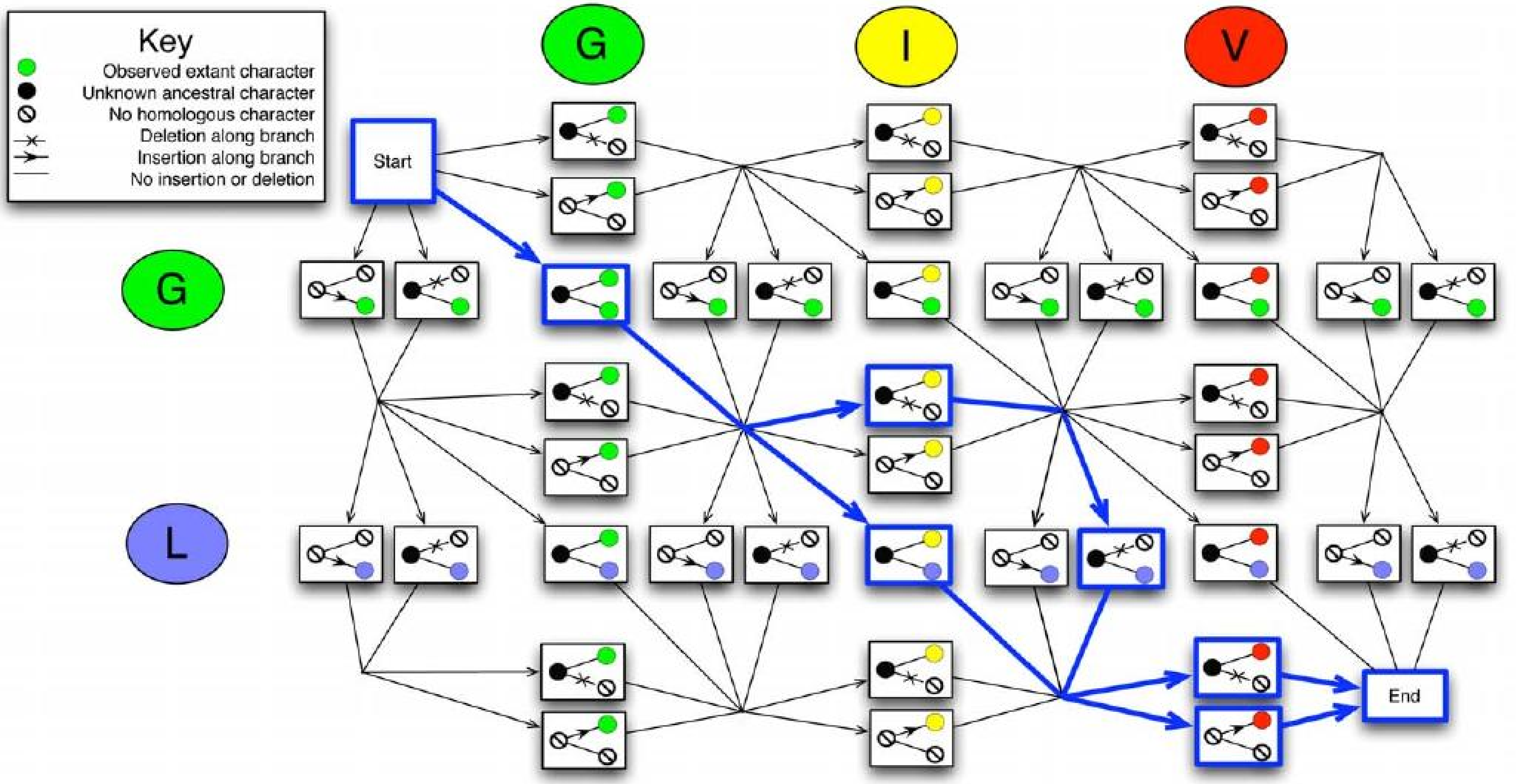
```
a1  A C G T - C A G C T A C G T A
a2  A C G T - G A G C T A C G T A
a3  A C G T - - A G C T A C G T G
a4  A C G T - - A G C T A C G T G
b2  A C G T G - A G C T A C G T G
b1  A C G T C - A G C T A C G T G
b3  A C G T - - A G C T A C G T G
b4  A C G T - - A G C T A C G T G
```



8 independent deletions

distinct insertions at same position

2 independent deletions



ProtPal is similar to PRANK, but is retains a set of inferred ancestral seqs.
Fig. 5 of Westesson et al. (2012)

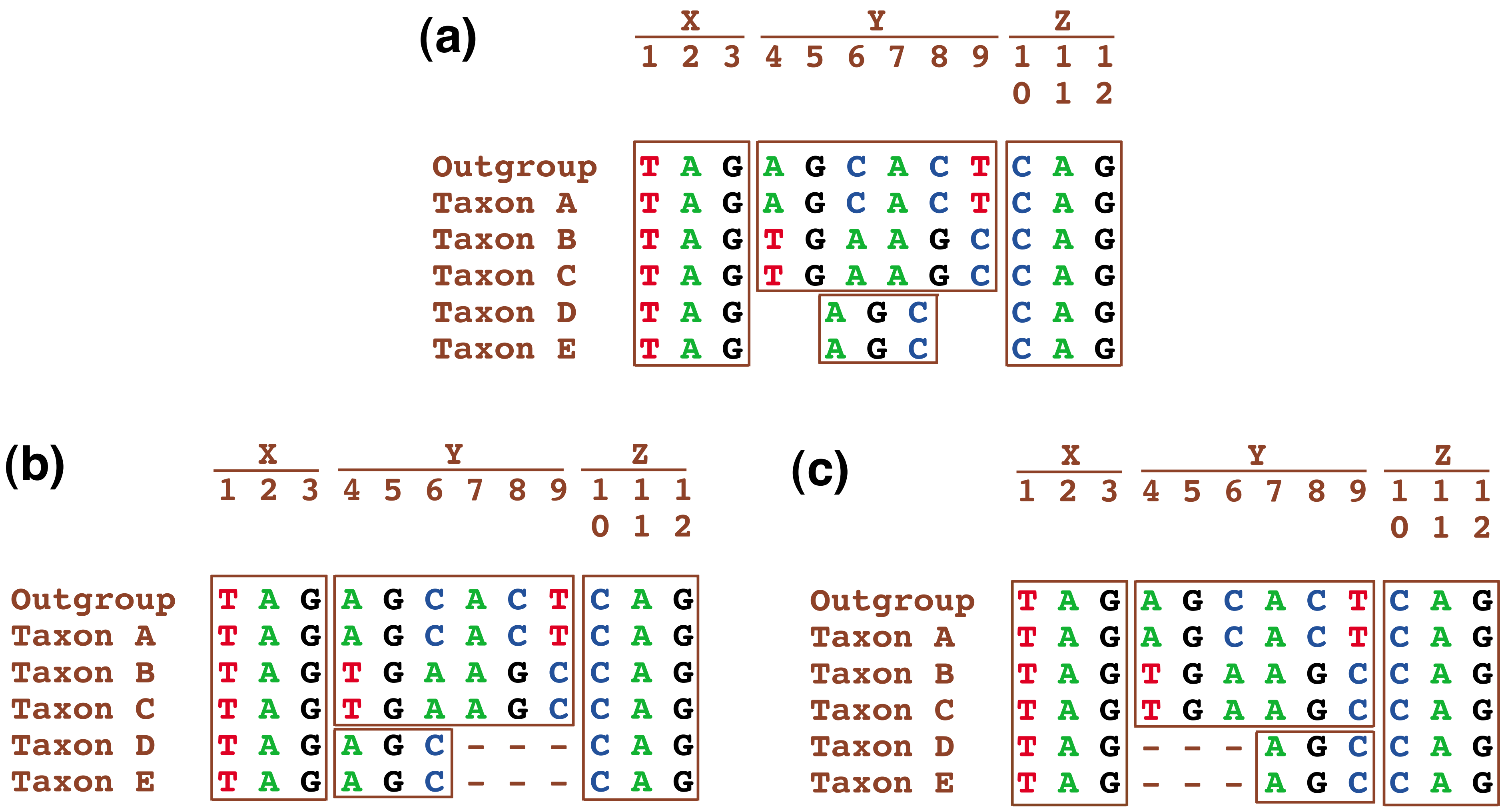
Impact of the guide tree

Using a guide tree can bias subsequent tree inference toward the guide tree.

This can also cause inflated support.

Ironically, this effect may be more of a problem for a more evolutionarily-sensible aligner such as PRANK or ProtPal!

Dealing with alignment ambiguity



(a)

	X			Y						Z		
	1	2	3	4	5	6	7	8	9	10	11	12
Outgroup	T	A	G	A	G	C	A	C	T	C	A	G
Taxon A	T	A	G	A	G	C	A	C	T	C	A	G
Taxon B	T	A	G	T	G	A	A	G	C	C	A	G
Taxon C	T	A	G	T	G	A	A	G	C	C	A	G
Taxon D	T	A	G		A	G	C			C	A	G
Taxon E	T	A	G		A	G	C			C	A	G

(b)

	X			Y						Z		
	1	2	3	4	5	6	7	8	9	10	11	12
Outgroup	T	A	G	A	G	C	A	C	T	C	A	G
Taxon A	T	A	G	A	G	C	A	C	T	C	A	G
Taxon B	T	A	G	T	G	A	A	G	C	C	A	G
Taxon C	T	A	G	T	G	A	A	G	C	C	A	G
Taxon D	T	A	G	A	G	C	–	–	–	C	A	G
Taxon E	T	A	G	A	G	C	–	–	–	C	A	G

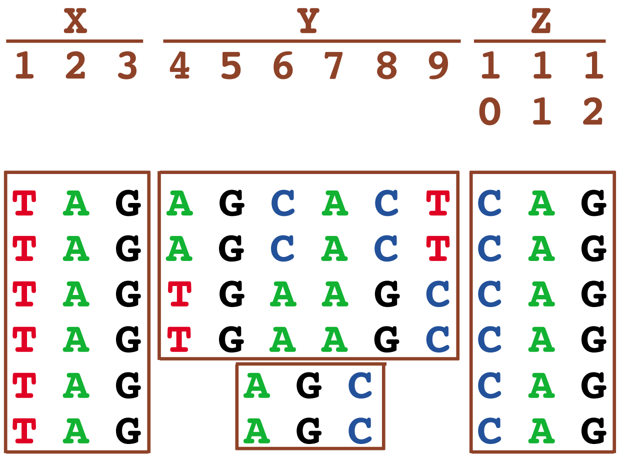
(c)

	X			Y						Z		
	1	2	3	4	5	6	7	8	9	10	11	12
Outgroup	T	A	G	A	G	C	A	C	T	C	A	G
Taxon A	T	A	G	A	G	C	A	C	T	C	A	G
Taxon B	T	A	G	T	G	A	A	G	C	C	A	G
Taxon C	T	A	G	T	G	A	A	G	C	C	A	G
Taxon D	T	A	G	–	–	–	A	G	C	C	A	G
Taxon E	T	A	G	–	–	–	A	G	C	C	A	G

from M. S. Y. Lee, *TREE*, 2001

Dealing with alignment ambiguity - deletion

(a)



	X			Y						Z		
	1	2	3	4	5	6	7	8	9	10	11	12
Outgroup	T	A	G	A	G	C	A	C	T	C	A	G
Taxon A	T	A	G	A	G	C	A	C	T	C	A	G
Taxon B	T	A	G	T	G	A	A	G	C	C	A	G
Taxon C	T	A	G	T	G	A	A	G	C	C	A	G
Taxon D	T	A	G		A	G	C			C	A	G
Taxon E	T	A	G		A	G	C			C	A	G

	X			Z		
	1	2	3	10	11	12
Outgroup	T	A	G	C	A	G
Taxon A	T	A	G	C	A	G
Taxon B	T	A	G	C	A	G
Taxon C	T	A	G	C	A	G
Taxon D	T	A	G	C	A	G
Taxon E	T	A	G	C	A	G

	X			Y						Z		
	1	2	3	4	5	6	7	8	9	10	11	12
Outgroup	T	A	G	A	G	C	A	C	T	C	A	G
Taxon A	T	A	G	A	G	C	A	C	T	C	A	G
Taxon B	T	A	G	T	G	A	A	G	C	C	A	G
Taxon C	T	A	G	T	G	A	A	G	C	C	A	G
Taxon D	T	A	G	?	?	?	-	-	-	C	A	G
Taxon E	T	A	G	?	?	?	-	-	-	C	A	G

from M. S. Y. Lee, *TREE*, 2001

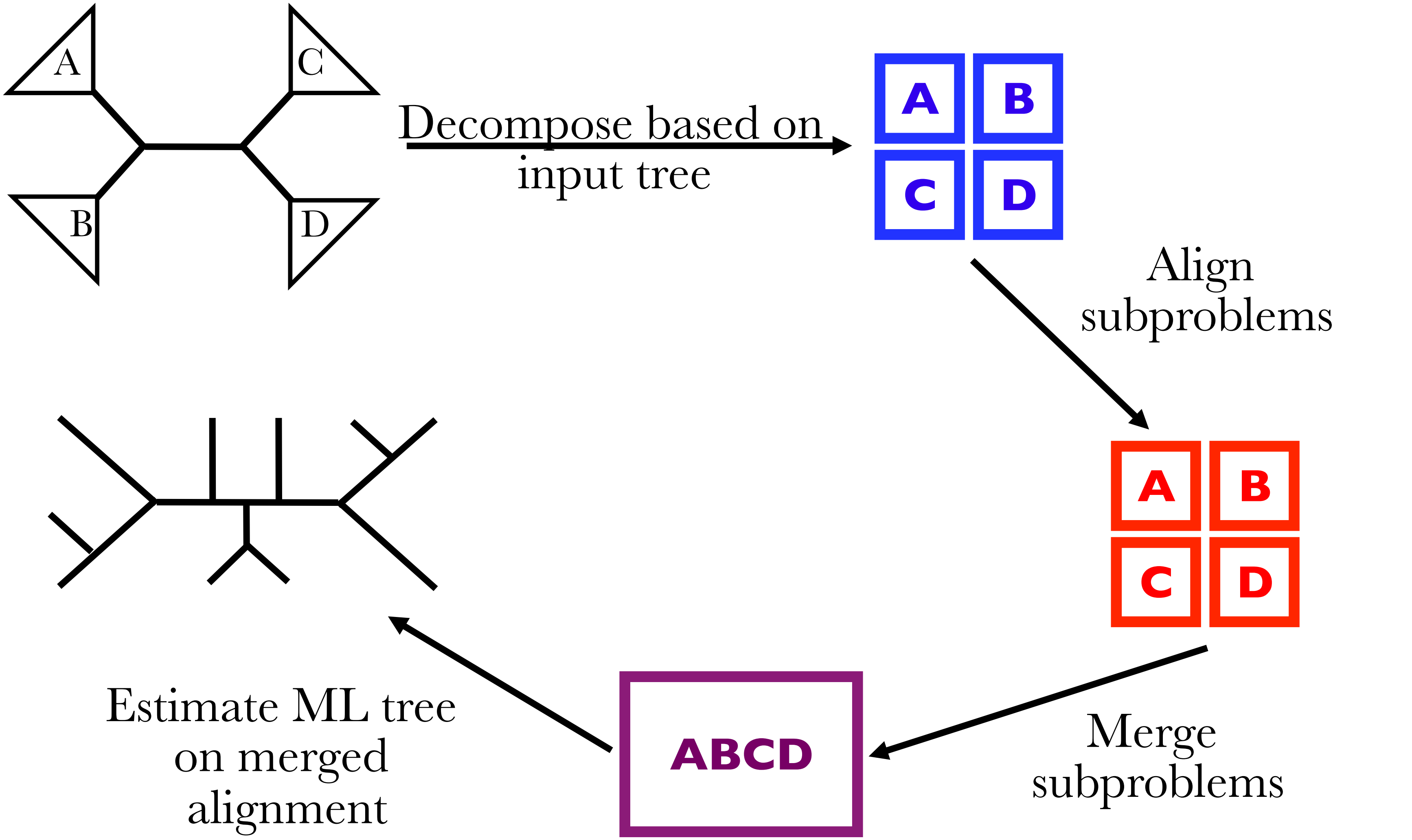
Filtering

- GUIDANCE score (Penn et al., 2010) which reflects sensitivity of the alignment to poorly supported parts of the guide tree; and
- Head-or-Tails algorithm (Landan and Graur, 2007) which compares the alignment of the sequences to the alignment that you would get from reversed forms of the sequences.
- SOAP (Löytynoja and Milinkovitch, 2001) looks for parts of the alignment that are sensitive to alignment parameters
- trimAl (Capella-Gutiérrez et al., 2009) uses column properties and consistency
- TCS (Chang et al., 2014) uses consistency and provides column scores
- Gblocks, examine properties of the columns in the matrix.

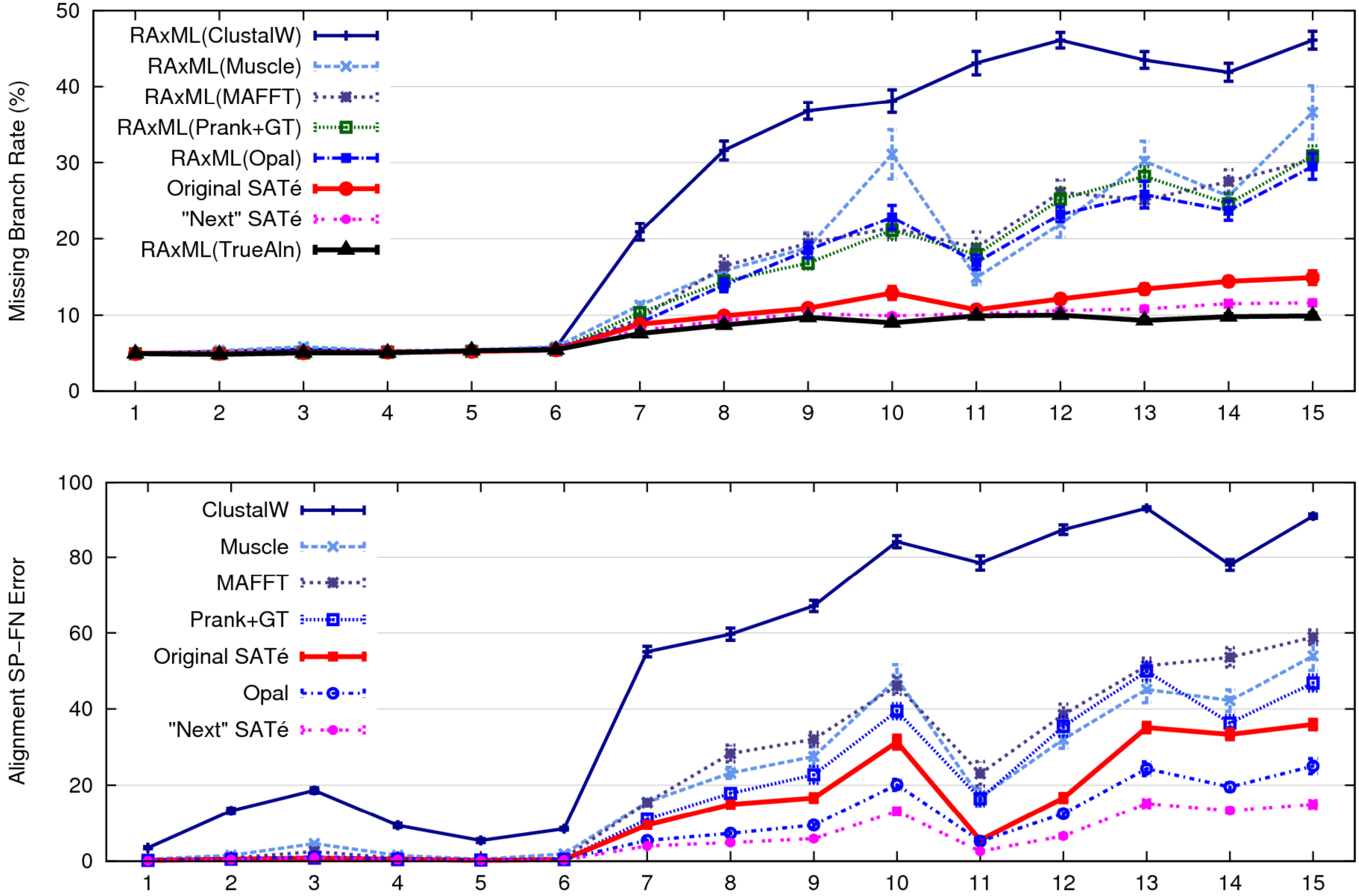
Simultaneous tree inference and alignment

- Ideally we would jointly estimate with uncertainty
- Allows for application of statistical models to improve inference and assessments of reliability
- Just now becoming feasible: `BAliPhy` (Redelings and Suchard, 2005)
  See also: `POY` (Wheeler, Gladstein, Laet, 2002), `Handel` (Holmes and Bruno, 2001) , and `BEAST`(Lunter et al., 2005; Drummond and Rambaut, 2007).
- SATé (Liu et al., 2009) and PASTA (Mirarab et al., 2014) are iterative (back and forth between tree and alignment estimation)

SATé repeats the following steps until termination


A
C
B
D
Decompose based on input tree
A
B
C
D
Align subproblems
A
B
C
D
Merge subproblems
ABCD
Estimate ML tree on merged alignment

SATé simulation results



Conclusions

- Evolutionary multiple sequence alignment is still a very active area of research.
- We are hampered by:
  - lacking a good criterion to optimize (when T is unknown).
  - being forced to use rough heuristics to optimize the sum of pairs scores.
- Filtering throws away information, but may be helpful
- Most phylogenetic inference tools ignore information from the indel process (but see Rivas et al. (2008); Rivas and Eddy (2013) and GARLI's DIMM)

Recommendations

- See if BaliPhy can run on your data! If not, try FSA
- Look at your alignments
- Be aware that the standard aligners only consider substitutions and indels. If your data shows inversions, duplications, . . . there can be serious artifacts.
- if you have protein-coding sequences, use amino acid or codon-aware aligners
- if you have sequences from an RNA with important secondary structure, check out Infernal

References

Bouchard-Côté, A. and Jordan, M. I. (2013). Evolutionary inference via the poisson indel process. *Proceedings of the National Academy of Sciences*, 110(4):1160–1166.

Bradley, R. K., Roberts, A., Smoot, M., Juvekar, S., Do, J., Dewey, C., Holmes, I., and Pachter, L. (2009). Fast Statistical Alignment. *PLoS Computational Biology*, 5(5).

Capella-Gutiérrez, S., Sillla-Martínex, J. M., and Gabaldón, T. (2009). trimal: a tool for automated alignment trimming in large-scale phylogenetic analyses. *Bioinformatics*.

Chang, J.-M., Di Tommaso, P., and Notredame, C. (2014). Tcs: a new multiple sequence alignment reliability measure to estimate alignment accuracy and improve phylogenetic tree reconstruction. *Molecular Biology and Evolution*, 31:1625–1637.

Do, C. B., Mahabhashyam, M. S., Brudno, M., and Batzoglou, S. (2005). Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome research*, 15(2):330–340.

Drummond, A. J. and Rambaut, A. (2007). Beast: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.*, 7:214.

Edgar, R. (2004). MUSCLE: a multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics*, 5:113.

Holmes, I. and Bruno, W. J. (2001). Evolutionary HMMs: a Bayesian approach to multiple alignment. *Bioinformatics (Oxford, England)*, 17(9):803–820.

Landan, G. and Graur, D. (2007). Heads or tails: a simple reliability check for multiple sequence alignments. *Molecular Biology and Evolution*, 24(6):1380–1383.

Liu, K., Raghavan, S., Nelesen, S., Linder, C., and Warnow, T. (2009). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324(5934):1561–1564.

Löytynoja, A. and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(30):10557–10562.

Löytynoja, A. and Milinkovitch, M. C. (2001). Soap, cleaning multiple alignments from unstable blocks. *Bioinformatics*, 17(6):573–574.

Lunter, G., Miklós, I., Drummond, A., Jensen, J. L., and Hein, J. (2005). Bayesian coestimation of phylogeny and sequence alignment. *BMC Bioinformatics*, 6:83.

Mirarab, S., Nguyen, N., and Warnow, T. (2014). Pasta: ultra-large multiple sequence alignment. In *Research in Computational Molecular Biology*, pages 177–191. Springer.

Notredame, C., Higgins, D. G., and Heringa, J. (2000). T-coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205 – 217.

Penn, O., Privman, E., Landan, G., Graur, D., and Pupko, T. (2010). An alignment confidence score capturing robustness to guide tree uncertainty. *Molecular Biology and Evolution*, 27(8):1759–1767.

Rausch, T. and Reinert, K. (2011). *Problem Solving Handbook in Computational Biology and Bioinformatics*, chapter Practical Multiple Sequence Alignment, pages 21–44. Springer.

Redelings, B. and Suchard, M. (2005). Joint bayesian estimation of alignment and phylogeny. *Systematic Biology*, 54(3):401–418.

Rivas, E. and Eddy, S. (2013). A probabilistic evolutionary model compatible with standard affine gap cost sequence alignment. In Review.

Rivas, E., Eddy, S. R., and Haussler, D. (2008). Probabilistic phylogenetic inference with insertions and deletions. *PLoS Computational Biology*, 4(9):e1000172.

Schwartz, A. S. and Pachter, L. (2007). Multiple alignment by sequence annealing. *Bioinformatics*, 23(2):e24–e29.

Westesson, O., Lunter, G., Paten, B., and Holmes, I. (2012). Accurate reconstruction of insertion-deletion histories by statistical phylogenetics. *PLoS ONE*, 7(4):e34572.

Wheeler, T. J. and Kececioglu, J. D. (2007). Multiple alignment by aligning alignments. *Bioinformatics*, 23(13):i559–i568.

Some of the tricks clustal uses to produce better alignments

- Chooses substitution matrix (PAM or BLOSUM series for amino acids) based on sequence similarity

- AA residues in a neighborhood affect gap opening penalty (easier to have gaps in hydrophilic loops)

- Gap penalties are raised if a column has no gaps, but there are gaps nearby.

- Low scoring alignments may be postponed until a later stage.

Terminal gaps

Using normal gap costs causes problems if one sequence is missing the starting or ending residues:

```
Pongo    VDEFKLIVEGELGRLFVVPTQ
Gorilla  VD-------GELGRLFVVPTQ
```

instead of :

```
Pongo    VDEFKLIVEGELGRLFVVPTQ
Gorilla  -------VDGELGRLFVVPTQ
```

(Methods that utilize local alignment information are more appropriate in these cases).

Using free terminal gaps to avoid this problem, but you have to watch out for:

```
Pongo    VDEPFRFKLTNRGTSHIILVAPR
Gorilla  ------VDEPNRGTSHIILVAPR
```