

Optimisation Alignment. 9.11.07 (60 minutes)

<http://www.stats.ox.ac.uk/~hein/lectures.htm>

Current Topics in Computational Molecular Biology

Chapter 3. 45-58 + Chapter 4.71-82

α -globin (141) and β -globin (146)

V-LSPADKTNVKAANGKVGAAHAGEYGAELERMFLSFPTTKTYFPHF-DLS--H---GSAQVKGHGKKVADAL
VHLTPEEKSAVTALWGKV--NVDEVGGGEALGRLLVVPWTQRRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAF

TNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTISKYR
SDGLAHLNLDLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH

1. It often matches functional region with functional region.
2. Determines homology at residue/nucleotide level.
3. Similarity/Distance between molecules can be evaluated
4. Molecular Evolution studies.
5. Homology/Non-homology depends on it.

Evaluating alignments & choosing the best.

V-LSPADKTNVKAANGKVGAGHAGEYGAELERMFLSFPTTKTYFPHF-DLS--H---GSAQVKGHGKKVADAL
VHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAF

1. Similarity/Distance (Parsimony):

a. Similarity

Identity scores high – difference low.

variable positions are scored less extreme than conserved sites.

Used scores: identities, structural or log-odds $\log[p_{i,j}/(p_i * p_j)]$

b. Distance

The scale is reversed: identity low – difference high.

Used scores: identities, structural, genetic code, ...

c. Distance is easier to interpret – similarity more flexible (+ & -, + only).

2. Gaps – single or many at a time.

Many is better, slightly more complicated

3. Choose the alignment that optimizes the selection criteria –

minimize/maximize

Number of alignments, $T(n,m)$

T	1	9	41	129	321	681
G	1	7	25	63	129	231
T	1	5	13	25	41	61
T	1	3	5	7	9	11
T	1	1	1	1	1	1
	C	T	A	G	G	

Parsimony Alignment of two strings.

Sequences: s1=CTAGG s2=TTGT.

Basic operations:

transitions 2 (C-T & A-G), transversions 5, indels (g) 10.

$$\text{Cost Additivity} \quad \begin{array}{ccc} \text{CTAG} & & \text{CTA} & & \text{G} \\ & = & & + & \\ \text{TT-G} & & \text{TT-} & & \text{G} \end{array}$$

$$\begin{array}{l} \{ \text{CTAG}, \text{TTG} \}_{\text{AL}} \\ \underline{12} \end{array} = \begin{array}{l} \text{(A)} \quad \{ \text{CTA}, \text{TT} \}_{\text{AL}} \\ \underline{12} \\ \text{(B)} \quad \{ \text{CTA}, \text{TTG} \}_{\text{AL}} \\ \underline{4} \\ \text{(C)} \quad \{ \text{CTAG}, \text{TT} \}_{\text{AL}} \\ \underline{22} \end{array} + \begin{array}{l} \text{GG} \\ \underline{0} \\ \text{G-} \\ \underline{10} \\ \text{-G} \\ \underline{10} \end{array}$$

Initial condition: $D_{0,0}=0$. ($D_{i,j} := D(s1[1:i], s2[1:j])$)

$$D_{i,j} = \min\{D_{i-1,j-1} + d(s1[i], s2[j]), D_{i,j-1} + g, D_{i-1,j} + g\}$$

	40	32	22	14	9	17
T	30	22	12	4	12	22
G	20	12	2	12	22	32
T	10	2	10	20	30	40
T	0	10	20	30	40	50
	C	T	A	G	G	

Alignment:

CTAGG

i v

TT-GT

Cost 17

Complexity of Accelerations of pairwise algorithm.

Dynamical Programming: $(n+1)(m+1)3=O(nm)$

Backtracking: $O(n+m)$

Recursion without memory: $T(n,m) > 3^{\min(n,m)}$

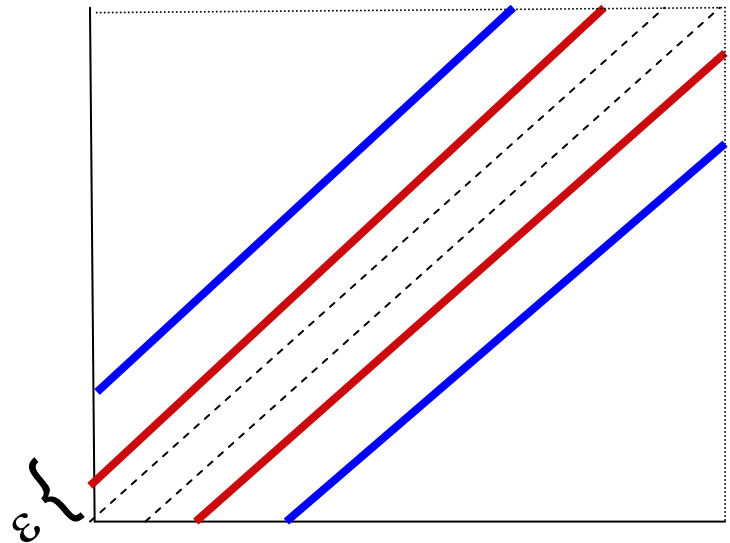
$(T(n,m)=T(n-1,m)+T(n,m-1)+T(n-1,m-1), T(0,0)=1)$

Exact acceleration (Ukkonen, Myers).

Assume all events cost 1.

If $d_\varepsilon(s_1, s_2) < 2\varepsilon + |l_1 - l_2|$, then

$d(s_1, s_2) = d_\varepsilon(s_1, s_2)$



Heuristic acceleration: Smaller band & larger acceleration, but no guarantee of optimum.

Close-to-Optimum Alignments

(Waterman & Byers, 1983)

Alignments within ϵ of optimal

Ex. $\epsilon = 2$.

	40	32	22	14	9	*	17
T				*	/		
	30	22	12	4	12		22
G			*	/			
	20	12	2	-	12		22
T		/					32
	10	2	10	20	30		40
T	/						
	0	10	20	30	40		50
	C	T	A	G	G		

C T A G G
i **i** **v** **g**
 T T G T -

Cost 19

Caveat:

There are enormous numbers of suboptimal alignments.

Hirschberg & Close-to-Optimum Alignments

(Hirschberg, 1975).

Sets of positions that are on some suboptimal alignment.

Alignments within ϵ of optimal. Ex. $\epsilon = 2$

	40 / 50	32 / 40	22 / 30	14 / 20	9 / 10	17 / 0
T						
	30 / 40	22 / 30	12 / 25	4 / 15	12 / 5	22 / 10
G						
	20 / 35	12 / 25	2 / 15	<u>12 / 5</u>	22 / 10	32 / 20
T						
	10 / 25	2 / 15	10 / 15	20 / 15	30 / 20	40 / 30
T						
	0 / 17	10 / 15	20 / 20	30 / 25	40 / 30	50 / 40
	C	T	A	T	G	G

Mid point: (3,2) and the alignment problem is then reduced to 2 smaller alignment problems: (CTA + TT) and (GG + GT)

Longer Indels

TCATGGTACCGTTAGCGT
GCA-----GCAT

g_k : cost of indel of length k .

Initial condition: $D_{0,0}=0$

$$D_{i,j} = \min \{$$

$$D_{i-1,j-1} + d(s1[i],s2[j]),$$

$$D_{i,j-1} + g_1, D_{i,j-2} + g_2, ,$$

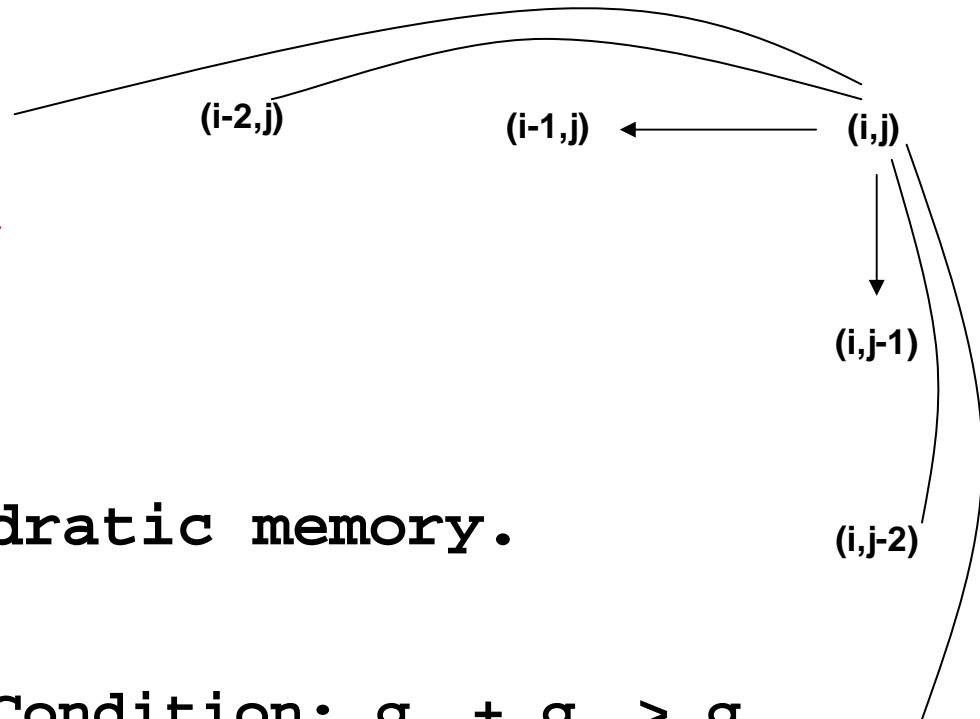
$$D_{i-1,j} + g_1, D_{i-2,j} + g_2, ,$$

}

Cubic running time. Quadratic memory.

Comment:

Evolutionary Consistency Condition: $g_i + g_j > g_{i+j}$



If $g_k = a + b*k$, then quadratic running time

Gotoh (1982) $D_{i,j}$ is split into 3 types:

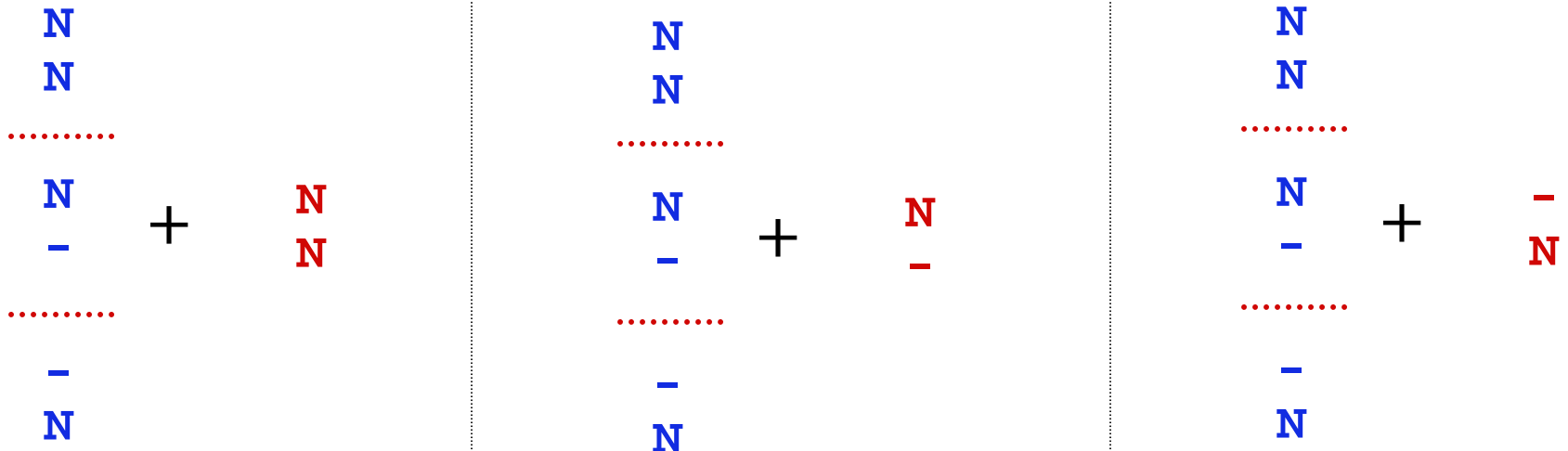
1. $D0_{i,j}$ as $D_{i,j}$, except $s1[i]$ must match $s2[j]$.
2. $D1_{i,j}$ as $D_{i,j}$, except $s1[i]$ is matched with "-".
3. $D2_{i,j}$ as $D_{i,j}$, except $s2[i]$ is matched with "-".

Then:

$$D0_{i,j} = \min(D0_{i-1,j-1}, D1_{i-1,j-1}, D2_{i-1,j-1}) + d(s1[i], s2[j])$$

$$D1_{i,j} = \min(D1_{i,j-1} + b, D0_{i,j-1} + a + b, D2_{i,j-1})$$

$$D2_{i,j} = \min(D2_{i-1,j} + b, D0_{i-1,j} + a + b, D1_{i-1,j})$$



Distance-Similarity.

(Smith-Waterman-Fitch, 1982)

$$S_{i,j} = \max\{S_{i-1,j-1} + s(s1[i],s2[j]), S_{i,j-1} - w, S_{i-1,j} - w\}$$

Similarity
 $s(n1,n2)$
 w

Distance
 $M - d(n1,n2)$
 $1/(2*M) + g$

Similarity: Transversions:0 Transitions:3 Identity:5 Indels: 10 + 1/10

Distance: Transitions:2 Transversions 5 Identity 0 Indels:10. M largest dist (5)

T	40/-40.4	32/-27.3	22/-12.2	14/0.9	9/11.0	17/2.9
G	30/-30.3	22/-17.2	12/-2.1	4/11.0	12/2.9	22/-7.2
T	20/-20.2	12/-7.1	2/8.0	12/-2.1	22/-12.2	32/-22.3
T	10/-10.1	2/3.0	10/-7.1	20/-17.2	30/-27.3	40/-37.4
	0/0	10/-10.1	20/-20.2	30/-30.3	40/-40.4	50/-50.5
	C	T	A	G	G	

1. The Switch from Dist to Sim is highly analogous to Maximizing $\{-f(x)\}$ instead of Minimizing $\{f(x)\}$.

2. Dist will based on a metric:

- i. $d(x,x) = 0$, ii. $d(x,y) \geq 0$, iii. $d(x,y) = d(y,x)$ &
- iv. $d(x,z) + d(z,y) \geq d(x,y)$.

There are no analogous restrictions on Sim, giving it a larger parameter space.

Local alignment

Smith, Waterman (1981)

Global Alignment:

$$S_{i,j} = \max\{D_{i-1,j-1} + s(s1[i],s2[j]), S_{i,j-1} - w, S_{i-1,j} - w\}$$

Local:

$$S_{i,j} = \max\{D_{i-1,j-1} + s(s1[i],s2[j]), S_{i,j-1} - w, S_{i-1,j} - w, \underline{0}\}$$

	0	1	0	.6	1	2	.6	1.6	1.6	3	2.6
C	0	0	1	0	1	.3	.6	0.6	2	3	1.6
A	0	0	0	1.3	0	1	1	2	<u>3.3</u>	2	1.6
G	0	0	.3	.3	1.3	1	2.3	<u>2.3</u>	2	.6	1.6
C	0	0	.6	1.6	.3	1.3	<u>2.6</u>	2.3	1	.6	1.6
U	0	0	2	.6	.3	<u>1.6</u>	2.6	1.3	1	.6	1
A	0	1	.6	0	1	<u>3</u>	1.6	1.3	1	1.3	1.6
C	0	1	0	0	<u>2</u>	1.3	.3	1	.3	2	.6
C	0	0	0	<u>1</u>	.3	0	0	.6	1	0	0
G	0	0	<u>0</u>	.6	1	0	0	0	1	1	2
U	0	0	1	.6	0	0	0	0	0	0	0
A	0	0	1	0	0	0	0	0	0	0	0
A	0	0	0	0	0	0	0	0	0	0	0
	C	A	G	C	C	U	C	G	C	U	U

Score Parameters:

Match: 1

Mismatch -1/3

Gap 1 + k/3

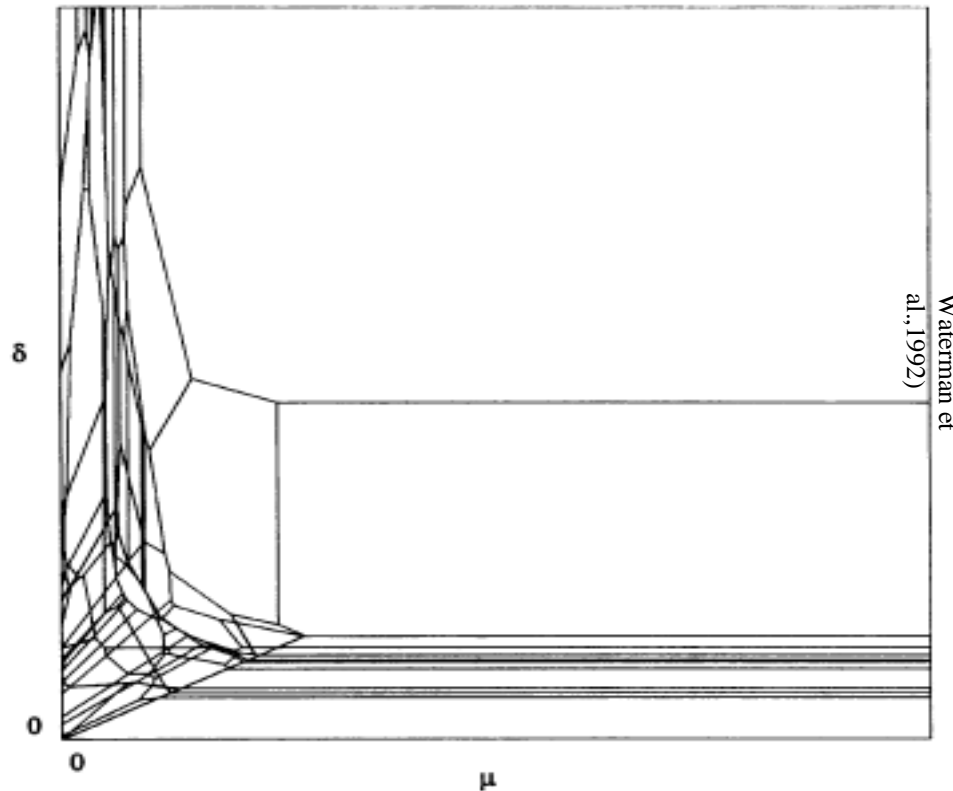
GCC-UCG

GCCAUG

Parametric Alignment

Waterman et al. 1992, Gusfield et al., 1992

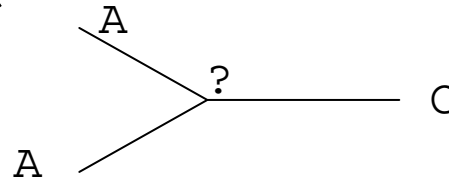
- The set of alignments is finite, while parameter space is region of Euclidian Space.
- The parameter space can be tiled into areas with the same optimal alignment.



Alignment of three sequences.

s1=ATCG s2=ATGCC s3=CTCC

Alignment: **AT-CG**
 ATGCC
 CT-CC



Consensus sequence: ATCC

Configurations in an alignment column:

-	-	n	n	n	-	n	-
-	n	-	n	-	n	n	-
n	-	-	-	n	n	n	-

Recursion: $D_{i,j,k} = \min\{D_{i-i',j-j',k-k'} + d(i,i',j,j',k,k')\}$

Initial condition: $D_{0,0,0} = 0.$

Running time: $l_1 * l_2 * l_3 * (2^3 - 1)$ Memory requirement: $l_1 * l_2 * l_3$

New phenomena: ancestral sequence.

Parsimony Alignment of four sequences

s1=ATCG s2=ATGCC s3=CTCC s4=ACGCG

Alignment: AT-CG
 ATGCC
 CT-CC
 ACGCG



Configurations in alignment columns:

-	-	-	n	-	-	-	n	n	n	-	n	n	n	n	-
-	-	n	-	n	n	-	n	-	-	n	-	n	n	n	-
-	n	-	-	n	-	n	-	n	-	n	n	-	n	n	-
n	-	-	-	-	n	n	-	-	n	n	n	n	-	n	-

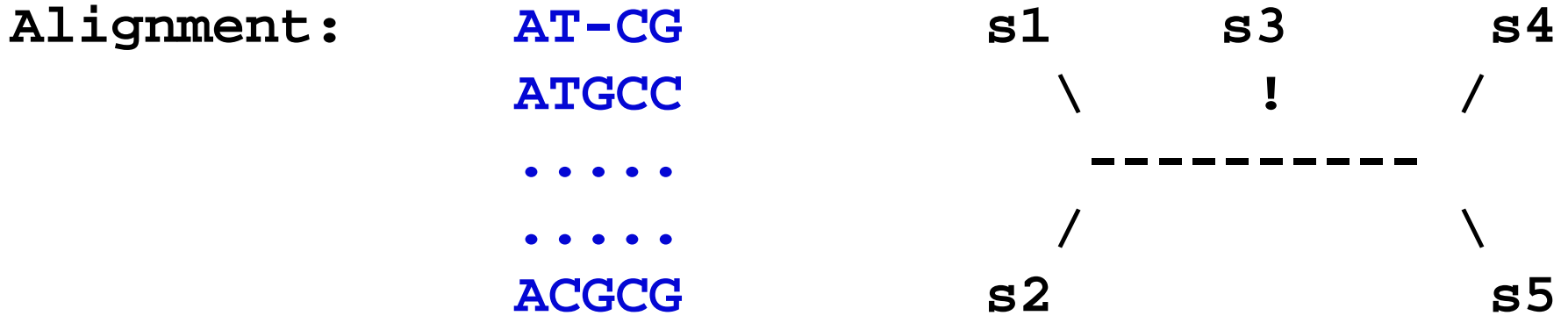
Recursion: $D_i = \min\{D_{i-\Delta} + d(i, \Delta)\} \Delta [\{0,1\}^4 \setminus \{0\}^4]$

Initial condition: $D_0 = 0.$

Computation time: $l_1 * l_2 * l_3 * l_4 * 2^4$ Memory : $l_1 * l_2 * l_3 * l_4$

Alignment of many sequences.

s1=ATCG, s2=ATGCC,, sn=ACGCG



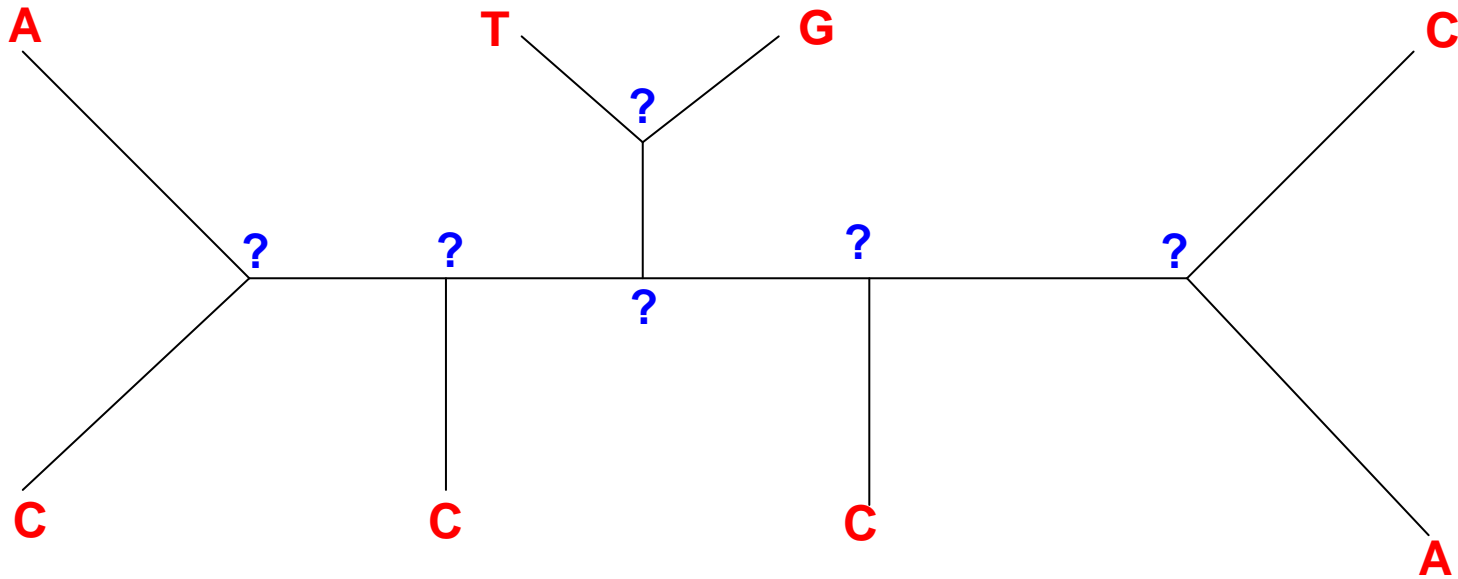
Configurations in an alignment column: $2^n - 1$

Recursion: $D_i = \min\{D_{i-\Delta} + d(i, \Delta)\} \quad \Delta \in [\{0, 1\}^n \setminus \{0\}^n]$

Initial condition: $D_{0,0,\dots,0} = 0.$

Computation time: $l^n * (2^n - 1) * n$ Memory requirement: l^n
(l: sequence length, n: number of sequences)

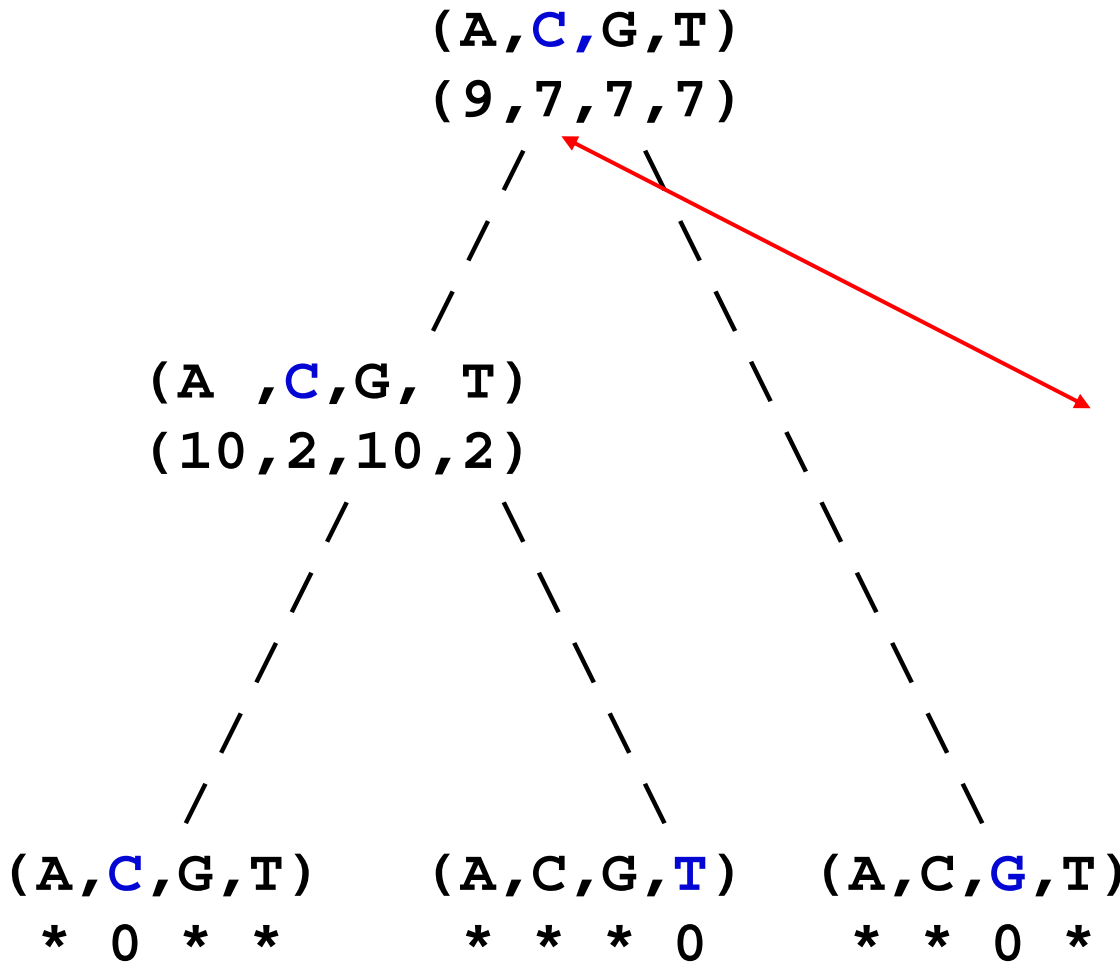
Assignment to internal nodes: The simple way.



What is the cheapest assignment of nucleotides to internal nodes, given some (symmetric) distance function $d(N_1, N_2)$??

If there are k leaves, there are $k-2$ internal nodes and 4^{k-2} possible assignments of nucleotides. For $k=22$, this is more than 10^{12} .

Fitch-Hartigan-Sankoff Algorithm



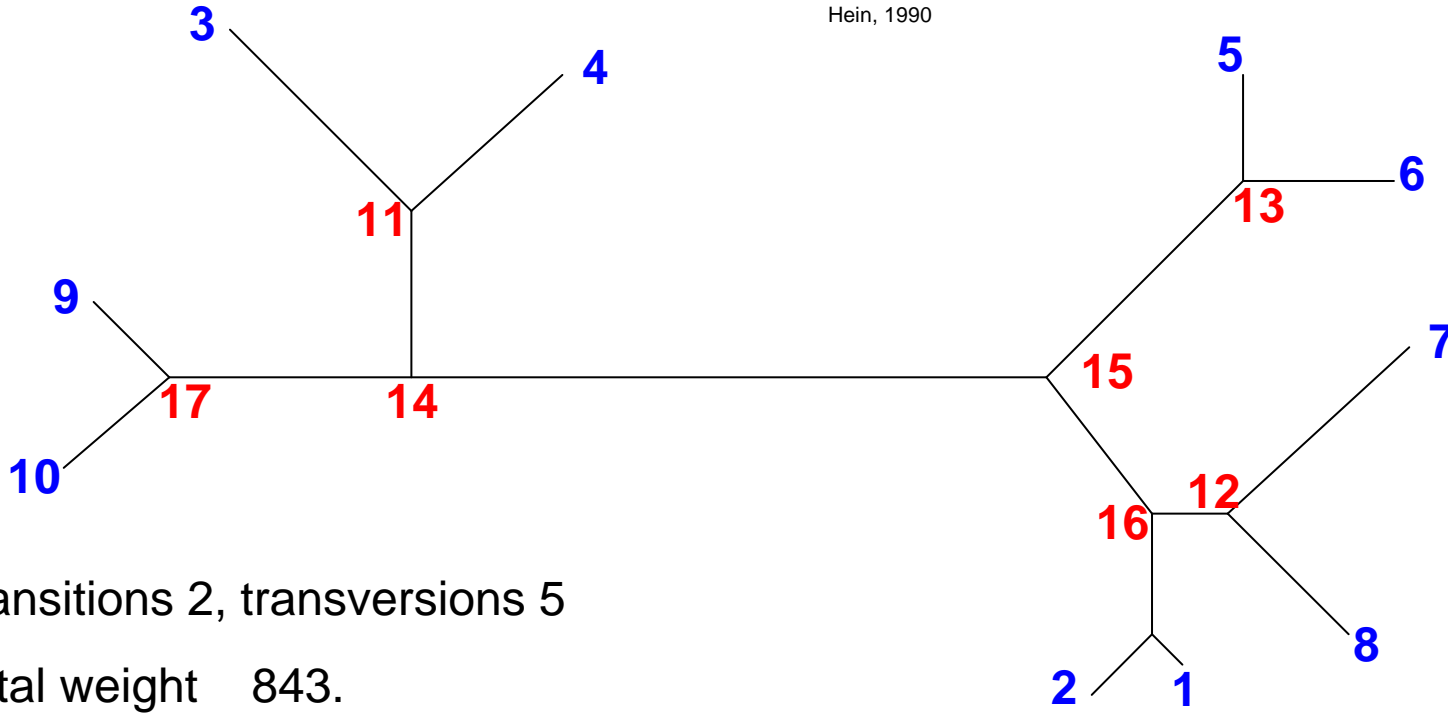
Costs: Transition 2,
Transversion 5, indel 10.

The cost of cheapest tree hanging from this node given that there is a "C" at this node

Indel Constraint: Nucleotides is connected set.

5S RNA Alignment & Phylogeny

Hein, 1990



Transitions 2, transversions 5

Total weight 843.

```

10 tatt-ctggtgtcccaggcgtagaggaaccacaccgatccatctcgaacttgggtggtgaaactctgccgcggt--aaccaatact-cg-gg-gggggccct-gcggaaaaatagctcgatgccagga--ta
17 t--t-ctggtgtcccaggcgtagaggaaccacccaatccatccogaacttgggtggtgaaactctgctgcggt--ga-cgatact-tg-gg-gggagcccg-atggaaaaatagctcgatgccagga--t-
9 t--t-ctggtgtctcaggcgtggaggaaccacccaatccatcccgaacttgggtggtgaaactctattgcggt--ga-cgatactgta-gg-ggaagcccg-atggaaaaatagctcgacgccagga--t-
14 t----ctggtggccatggcgttagaggaaacacccccatcccataccgaactcggcagttaagctctgctgcgcc--ga-tggtact-tg-gg-gggagcccg-ctgggaaaaataggacgctgccag-a--t-
3 t----ctggtgatgatggcggaggggacacacccttcccataccgaacacggccttaagccctccagcgcc--aa-tggtact-tgctc-cgcagggag-cgggagagtaggacgctgccag-g--c-
11 t----ctggtggcgtaggcgaagaggacacacccttcccataccgaacacggcagttaagctctccagcgcc--ga-tggtact-tg-gg-ggcagtcgg-ctgggagagtaggacgctgccag-g--c-
4 t----ctggtggcgtatagcgaagaaggtcacacccttcccataccgaacacggaggttaagcttctcagcgcc--ga-tggtact-tg-gg-ggctgtccc-ctgtgagagtaggacgctgccag-g--c-
15 g----cctgcgccatagcaccgtgaaagcaccatcccct-cogaactcggcagttaagcagcgttgcccagga-tagtact-tg-ggtgggagaccgctgggaaacctggatgctgcaag-c--t-
8 g----cctacggccatcccaccctggtaacgcccgatctcgt-ctgatctcggaaagctaaagcaggttgggctgggt-tagtact-tg-gatgggagaccctcctgggaaatacgggtgctgtagg-ct-t-
12 g----cctacggccataccaccctgaaagcaccatcccct-cogatctgggaagttaagcagggttgagcccagt-tagtact-tg-gatgggagaccgctgggaaacctgggtgctgtagg-c--t-
7 g----cttacgaccatatacgttgaatgcacgccatcccct-cogatctggcaagttaagcaacgttgagtcaggt-tagtact-tg-gatcggagaccgctgggaaacctggatgttgaag-c--t-
16 g----cctacggccatagcaccctgaaagcaccatcccct-cogatctgggaagttaagcagggttgcccaggt-tagtact-tg-ggtgggagaccgctgggaaacctgggtgctgtagg-c--t-
1 a----tccacggccataggactctgaaagcactgcatcccct-cogatctgcaagttaaccagagtagcggcccagt-tagtacc-ac-ggtgggggaccacgcgggaaacctgggtgctgt-gg-t--t-
18 a----tccacggccataggactctgaaagcaccgcatcccct-cogatctgcaagttaaccagagtagcggcccagt-tagtacc-ac-ggtgggggaccacatgggaaacctgggtgctgt-gg-t--t-
2 a----tccacggccataggactgtgaaagcaccgcatcccct-ctgatctgcgcagttaaacacagtgcccctagt-tagtacc-at-ggtgggggaccacatgggaaacctgggtgctgt-gg-t--t-
5 g---tggtgcggtcataccagcgtaatgcaccggatcccct-cagaactccgcagttaagcgcgcttgggcccagaa-cagtact-gg-gatgggtgacctcccgggaaagtcctggtgccgacc-c--c-
13 g---ggtgcggtcataccagcgttaatgcaccggatcccct-cagaactccgcagttaagcgcgcttgggcccagc-tagtact-ag-gatgggtgacctcctgggaaagtcctgatgctgcacc-c--t-
6 g---ggtgcgatcataccagcgttaatgcaccggatcccct-cagaactccgcagttaagcgcgcttgggtggag-tagtact-ag-gatgggtgacctcctgggaaagtcctaatattgcacc-c-t-
    
```

Progressive Alignment

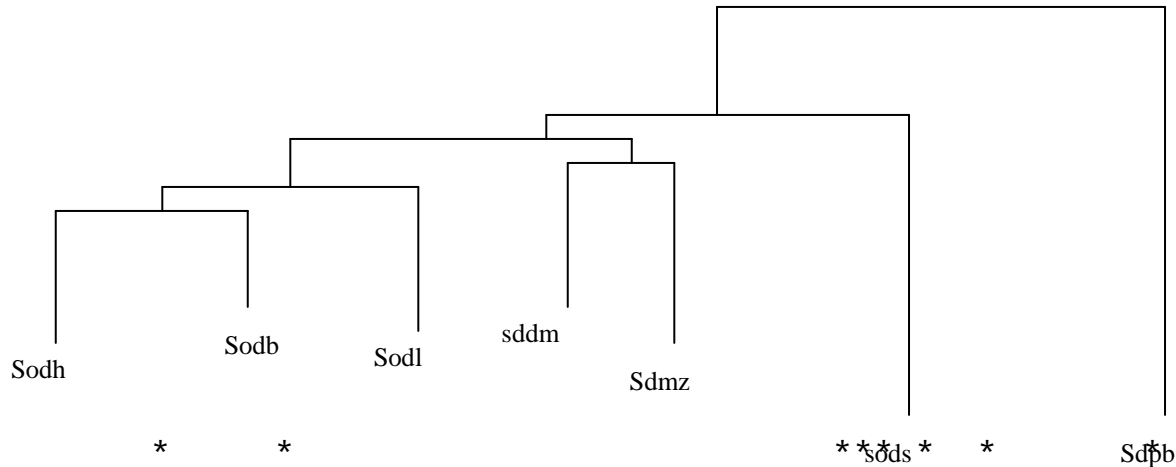
(Feng-Doolittle 1987 J.Mol.Evol.)

Can align alignments and given a tree make a multiple alignment.

```
*
alkmny-trwq
akkmdyftrwq
kkkmemftrwq

*
acdeqrt
acdehrt
```

$[P(n,q) + P(n,h) + P(d,q) + P(d,h) + P(e,q) + P(e,h)] / 6$



```
Sodh  atkavcvlkgdgpqvqgsinfeqkesdgpvkvwgsikglte-ghgfhvhqfg----ndtagct      sagphfnp  lsrk
Sodb  atkavcvlkgdgpqvqgtinfeak-gdtvkvwgsikglte--ghgfhvhqfg----ndtagct      sagphfnp  lsrk
Sodl  atkavcvlkgdgpqvqgsinfeqkesdgpvkvwgsikglte-ghgfhvhqfg----ndtagct      sagphfnp  lsrk
Sddm  atkavcvlkgdgpqvq -infeak-gdtvkvwgsikglte--ghgfhvhqfg----ndtagct      sagphfnp  lsrk
Sdmz  atkavcvlkgdgpqvq- infeqkesdgpvkvwgsikglte-ghgfhvhqfg----ndtagct      sagphfnp  Lsrk
Sods  vatkavcvlkgdgpqvq- infeak-gdtvkvwgsikgltepnglhgfhvhqfg----ndtagct      sagphfnp  lsrk
Sdpb  datkavcvlkgdgpqvq--infeqkesdgpv----wgsikgltglhgfhvhqfgscasndtagctvlggssagphfnpehtnk
```

Summary

Comparison of 2 Strings

- Minimize Distance-Maximize Similarity
- Dynamical Programming Algorithm
- Local alignment
- Close-to-Optimal Solutions
- Parametric Alignment

Comparison of many Strings

- Simultaneous Phylogeny and Alignment

History of Alignment

1953 Richard Bellman invents Dynamical Programming

1966: Levenstein formulates distance measure between sequences and introduces dynamica programming algorithm finding the distance.

1970: Needleman and Wunch compares proteins maximising a similarity score.

1972: Sankoff & Sellers reinvents the basic algorithm.

1972: Sankoff can align subject to the constraint that there must be exactly k indels.

1973: Sankoff makes multiple alignment and phylogeny - both exact & heuristic.

1975: Hirschberg gives linear memory algorithm.

1976: Waterman gives cubic algorithm allowing for indels of arbitrary length without reference to phylogeny.

1981: Waterman, Smith and Fitch shows duality of simiarity and distance.

1981 Smith and Waterman invents similarity based local alignment.

1982: Gotoh gives quadratic algorithm if gap penalty functionen is $g_k = a + b*k$ (for indel of length k). Uses 3 matrices in stead of 1.

1983: Waterman and Byers introduces close-to-optimal alignments.

1984-5: Ukkonen, Myers, Fickett accelerates algorithmen considerably.

1984: Hogeveg and Hespers introduces heuristic multiple phylogenetic alignment.

1984: Fredman introduces triple alignment generalisation of Needleman-Wunch.

1985: Lipman & Wilbur uses hashing. 1989: Myers introduces alignment with concave gap penalty function.

1987: Feng-Doolittle introducesphylogenetisk alignment: "Once a gap always a gap".

1989: Kececioglou makes strong acceleration of Sankoff's exact algorithm.

1991: Thorne, Kishino & Felsenstein makes good model for statistical alignment, partially introduced in 1986 by Thomson & Bishop.

1991: States & Botstein compares a DNA string with a protein in search of frameshift mutations.

1993-4: Gusfield, Lander, Waterman and others introduces parametric alignment.

1994: Krogh et al & Baldi et al. introduces Hidden Markov Models for multiple alignment.

1995: Mitcheson & Durbin introduces Tree-HMMs

1999 - Resurgence of interest in statistical alignment

References

- D. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 60:351-360, 1987.
- Fitch, W.(1971) "Towards defining the course of evolution: minimum change for a specific tree topology" *Systematic Zoology* 20.406-416.
- Gotoh, O. (1982). "An improved algorithm for matching biological sequences." *J. Mol. Biol.* **162**: 705-708.
- Hartigan,JA (1973) "Minimum mutation fit to a given tree" *Biometrics* 29.53-69.
- E. Myers, "[An O\(ND\) Difference Algorithm and Its Variations,](#)" *Algorithmica* 1, 2 (1986), 251-266.
- Needleman, S. B. and C. D. Wunsch (1970). "A general method applicable to the search for similarities in the amino acid sequences of two proteins." *J. Mol. Biol.* **48**: 443-453.
- Sankoff, D. (1975) "Minimal mutation trees for sequences" *SIAM journal on Applied Mathematics* 78.35-42.
- Sankoff,D. and Kruskal, J. (1983) "Time Warps, String Edits & Macromolecules" Addison-Wesley
- Smith, T. F., M. S. Waterman, et al. (1981). "Comparative Biosequence Metrics." *J. Mol. Evol.* **18**: 38-46.
- E. Ukkonen: Algorithms for approximate string matching. *Information and Control* 64 (1985), 100-118.