Alignment- an issue of homology:
As we have discussed previously, establishing an initial estimate of homology (i.e. primary homology or conjectural homology) is essential for all types of characters. Primary homology involves assessment of similarity that includes evaluation of *topographical identity* and *state identity*. However, I disagree with the Brower and Schawaroch paper (1996, one of your readings) that it is necessary or even possible to separate the implication of homology from the evaluation of these two aspects of similarity. If phylogenetic analysis is your intention, then here is no other reason to bother making a statement of similarity. On the other hand, it is important to recognize the difference between establishing the columns in the matrix (i.e. characters) via topographical identity, which are assumed background knowledge based on previous analyses, and establishing state identity (character states), which are subsequently tested by congruence. It is interesting and important to note that topographical identity for morphological characters is often (but certainly not always!) uncontroversial and state identity much more frequently problematic. For sequence data the opposite is usually the case, i.e. state identity (A,C,G,T) is a given but what are the columns (character or topographical identity) can be problematic.

The methodology of PCR and sequencing helps to establish broad-scale topographic identity by presumed primer specificity that results in a single product that is assumed to be homologous and orthologous. When this fails, often (we hope) alignment or phylogenetic analysis will show symptoms of this. Nuclear pseudogenes or other non-functional paralogs are often degenerated (for example they may include stop codons in an open reading frame) or otherwise are highly differentiated and are difficult to align. But we certainly can be fooled.

Assuming homology of the gene or region bound by conserved primer binding sites is not usually too problematic, however, in variable length regions, particularly in non-coding regions, establishing an alignment is very problematic. Typically a fixed alignment, achieved by one method or another, is treated as prior, or background knowledge to the phylogenetic analysis. In most cases the outcome of the phylogenetic analyses are influenced by the alignment in terms of topology and/or support values.

Note: In the earlier molecular literature you may see the term "percent homology", which is an incorrect use of the term homology. The correct way to refer to the difference/similarity of two sequences is percent similarity or percent identity.

Alignment, Pairwise, local and global:
Two sequences (strings of bases, amino acids, proteins, etc.) are matched in a **pairwise alignment** either **globally** (two sequences matched over their whole length) or **locally** (some subset of the sequences matched while other regions are not expected to match).

Local pairwise comparison is very useful in finding partially highly similar regions in a larger query sequence. The sequences and the local residues compared may or may not prove to be homologous. This is a strategy often used in bioinfomatic applications such as database searches.

A common and powerful example is of local pairwise alignment is BLAST (Altschul, SF, W Gish, W Miller, EW Myers, and DJ Lipman. Basic local alignment search tool. J Mol Biol 215(3):403-10, 1990). This is a very fast way to sift through a very large database of sequences if, for instance, a gene is newly identified and function understood in *Drosophila*, a researcher can BLAST the database of the human genome to look for similar gene sequences.

Very basic description of BLAST
1. Uses short segments ("words") of sequence to find other sequences that contain the same set.
2. Does "ungapped" alignment extending from the matched subsequence regions to find high-scoring matches
3. Does a rapid gapped alignment to select and rank close matches

Global pairwise alignment establishes overall sequence similarity usually by calculating a mathematical distance, i.e. minimum edit distance, between two sequences being compared. The alignment attempts to balance the number of indels (gaps) with the amount of base substitution, normally using some cost differential. It is possible to account for all differences between the pair by inserting enough gaps (trivial alignment), but this would be uninformative and

unrealistic. In the simplest model the "Edit distance" is the minimal number of events required to transform one sequence into another using some scheme of insertions, deletions and substitutions.

Go from acctga to agcta:
accgta <<[substitution]>> agctga <<[deletion]>> agcta
The edit distance = 2.

OTU1 ACTTCCGAATTTGGCT
OTU2 ACTCGATTGCCT

Minimize ind/dels

```
ACTTCCGGAATTTGGCT
|||*      **|||*||
ACTC-----GATTGCCT
```

Minimize substitutions

```
ACTTCCGAATTTGG-CT
|||  |||  |||  ||
ACT--CGA--TTG-CCT
```

Dynamic Programming and global alignment: (Needleman-Wunsch) underlies or is part of most alignment methods. Check out tutorial at http://www.avatar.se/molbioinfo2001/dynprog/dynamic.html

Global multiple alignment: Two problems- how to find alignments and how to choose.
For phylogenies, pairwise comparison is not sufficient. What must be done is **multiple sequence alignment**, a global solution for the whole data matrix or primary homology for the characters (columns) in the entire matrix.

Practical issues. For two sequences, i.e. pairwise alignment, of length n, if no gaps are allowed then there is one or few optimal alignment(s). If gaps are allowed, i.e. there is sequence length variation, then...

$(2n)!/(n!)^2$ e.g. n=50 then 10^{29} alignments. For global multiple alignment, where N= the number of sequences, an N-dimensional matrix implementing the dynamic programming is need. *Enumeration is not an option!* We need heuristic searches based on optimality and scoring.

Various methods and programs have been/are used to tackle this problem. Here are some.....

>>**Manual, by hand or by eye-** For very simple cases it may be sufficient to simply look at the matrix and make adjustments. This is not problematic for aligning the ends of coding sequences or when a "known" reference sequence is used.

However, for complex and variable length sequence data by-eye alone would seem to violate criteria of repeatability. There are no obvious or explicit costs for inserting gaps or making mis-matches. The counter argument is that the aligned matrix can be made available. However, what if I want to add or subtract OTUs? This would influence the alignment, but how? This is subject to individual pattern recognition abilities for thousands of bases and hundreds of sequences. It is also likely to increase the number of editing errors because of additional "handling" of sequences. On the other hand, all alignment programs are known to fail in some situations so checking a machine alignment to manually correct or realign subregions is a good idea. Any matrix with manual adjustments should be saved and made available for reanalysis.

>>**Manual alignments informed by consideration of secondary structure-**
There is reasonable to expect selective pressures to apply to secondary structure interactions (that is, requirements of compensatory changes), though it is unclear just how relevant those interactions are compared to selective pressures applied at other structural levels. It does allow for more information to be used to help define the topographical identity in a way similar to what is used for standard morphological characters.

This does not solve the problem of nucleotide homology. It does attempt to place constraints on changes by establishing putative limits between loop and stem regions. Nucleotides within each of those units must still be homologized and all the problems still apply.

Determination of secondary structure is not simple and not unambiguous. Generally the actual pattern of bonding is probabilistic and depends on the minimization of free energy and the thermodynamic stability of the resulting structure. In rare cases we have known structures from x-ray crystallography. Programs explicitly designed to model

secondary structure are not very realistic (yet) in terms of the actual cell environment and might find multiple, equally probable models. In phylogenetic studies, secondary structure is typically inferred by aligning with a sequence of "known" or implied secondary structure, although the basis of that knowledge remains uncertain and applicability to the study taxa is unclear in many cases, but this is heading in the right direction.

>>Automated methods:
Progressive alignment- As in Clustal W and X the most commonly used program for progressive alignment strategies.
1. All sequences are compared to each other (pairwise alignments)
2. A NJ tree is constructed, describing the approximate groupings of the sequences by similarity.
3. Final multiple alignment uses the guide tree so produced

Basically, the alignment is created by iteratively aligning sequences from the input to an already partially constructed solution. Obviously, the order is a crucial point in this method as uses an NJ or UPGMA tree-based alignment. Some argue that it doesn't make sense to determine alignment order with one optimality criterion (e.g., phenetics) and then analyze the alignment later with another (e.g., parsimony, ml) but to re-align on a parsimony tree derived from the first alignment to get an "improved" alignment may be circular.

More on the Clustal alignment strategy for variable length sequences: Given that Clustal basically sets up a cost ratio between a base mismatch and inserting a gap, and each of those can be further broken down into relative costs of transitions - transversion and gap opening – gap extensions for any given Clustal alignment there are four ratios of costs between inserting a gap and mismatching a base.

transition = substitution of a purine for another purine (A <-> G) or a pyrimidine for another pyrimidine (C <-> T)
Ts:Gap open
Ts:Gap extension
transversion = substitution of a purine for a pyrimidine or vice versa (A <-> T, C <-> G)
Tv:Gap open
Tv:Gap extension

For Clustal when the transition weight is set to 1.0 any transition is considered a match (cost = 0) and when it is set to 0.0 the costs is the same as a transversion (cost = 1). For example if I chose to use two weights, 0.0 and 0.5. This makes the cost equal to or half that of transversions.

For example to have this: Ts=0.5 Tv=1.0 GO=15.0 GE=6.66 [30:1, 13.3:1, 15.0:1, 6.66:1]

The command line is:
-PWGAPOPEN=15.0 -PWGAPEXT=6.66 -GAPOPEN=15.0 -GAPEXT=6.66 -ITERATION=alignment -NUMITER=100 -MAXDIV=40 -TRANSWEIGHT=0.5

Notice "-ITERATION=alignment -NUMITER=100" the recent version of Clustal allows tree or alignment iteration that attempts to improve the initial global multiple alignment. I have found that this eliminates most (but not all) of the segments that needed manual editing.

Progressive, consistency-based alignment- The genre of consistency-based multiple alignments are newer. These strategies are incorporating "local signals" into global alignment construction. See various program documentation (online) for more.

Simultaneous alignment- Simultaneous multiple alignments synchronise the information of all input sequences in a hyperspace lattice, e.g. so-called exact alignment algorithms using the divide-and-conquer (DCA) strategy (Tönges, U., Perrey, S.W., Stoye, J. and Dress, A.W.M. 1996. A general method for fast multiple sequence alignment.*Gene* **172**GC33-GC41). In part it cuts down the input sequences at carefully chosen positions to align in segments. Current algorithms cannot handle large/complex data sets.

Iterative, segment-based alignment- One example is DIALIGN, which iteratively collects local similar segments,

which can be merged into a common multiple alignment. Iteration continues until no more local signals can be found or until all positions are aligned. Recent benchmarks have shown that this strategy can even handle long sequences of a low overall similarity. No explicit gap cost or input trees.

Iterative and various.
Mafft- Fast, iterative with many options for similarity based alignments. Available online.
Opal- Another iterative approach. This can be run via Mesquite.
Muscle- Heuristic as implemented in the program iterating with parsimony guide-trees. Also can be run via Mesquite.
Malign- Uses tree swapping algorithms to imply alignments and then progressively attempts to find improved alignments.

Direct optimization- POY (W.Wheeler)- POY does phylogenetic analysis using sequence and/or morphological data, a s partitions or as a whole, using parsimony or maximum likelihood. The correspondences among homologues are determined and evaluated simultaneously with transformations and trees. The methods used assess directly the number of DNA sequence transformations, evolutionary events, required by a tree topology without the use of a prior static multiple sequence alignment (although it can do that too). Insertion and deletion events are by default counted as "real events" (transformations with costs) as apposed to being implied by the pattern as in multiple sequence alignments and (usually) treated as missing data.

Promoters say-

1. Eliminates inconsistent treatment of data between alignment and tree construction steps. Alignment and phylogeny can be inferred under the same parsimony or maximum likelihood framework.
2. "Dynamic homology" (Wheeler, 2001) is preferable to "static homology" where predetermined correspondences and putative homologies are established prior to analysis and applied uniformly throughout tree search.
3. Multiple sequence alignments are not "real" and can't found or observed in nature. The goal of DO is an optimal tree and there may be various sequence alignments constructs consistent with a given tree.

The program uses several methods- Direct optimization and iterative-pass optimization strive to construct HTU sequences such that the overall cladogram is of minimal length. This is done through modified two and three dimensional string-matching, respectively. Fixed-states optimization and search-based optimization draw optimal HTU sequences from a pool of predetermined sequences. This can be a small or large collection of possible sequences. Dynamic programming is used to identify the best HTU sequences and determine cladogram length.

Setting costs and implementing models. It is a normal part of most of these many of these methods to set costs as a ratio of gap:TS:TV. Determining what the best settings are is problematic. Results vary and there is no clear best way to choose. It has been suggested that comparing measures of congruence across a broad range of parameters is the best way to choose among the possible parameter sets or to determine that altering the parameters changes the results minimally. The obvious problem is that there is no objective measure to use or standard to compare against.

This is most commonly referred to as **Sensitivity analysis**- Use a range of costs and look for character congruence and/or topological congruence. Again when is it good enough? (Terry, M. & M. Whiting. 2005. Comparison of two alignment techniques within a single complex data set: POY versus Clustal. Cladistics. 21:272-281.)

ILD= (length of combined data set on MPT - sum of the lengths of the individual data set's MPT)/length of combined data set on MPT (Mickevich & Farris.1981. The implications of congruence in *Mendia*. Syst. Zool. 30:351-370)

RILD= (length of combined data set on MPT - sum of the lengths of the individual data set's MPT)/(maximum length of the combined data set- sum of the lengths of the individual data set's MPT)) (Wheeler &Hayashi. 1998 Phylogeny of extant chelicerate orders. Caldistics. 14:173-193)

What to do with those regions of "ugly" alignment?

>>**Manual purging "bad" data**
A method sometimes used get around problems in hard to align sections is the elimination of gap heavy regions in alignments. Conjectural homology is considered to be too uncertain so the segment of residues is down-weighted or removed (i.e. given a weight of 0). Exactly which columns should be eliminated and where the left and right boundaries fall is a subjective. Obviously keeping the data in may have an impact on the results otherwise why bother removing it.

>> **Coding gaps as information:**
Alternatively, the variable region can be converted into a character in each taxon and scored. This has all the problems above and adds another layer of difficulty in determining how to code the states. However, if indels are viewed as events that may mark history they can be informative. One method is to consider a gap (-) as a fifth character. In some cases where gaps are rare this may make sense, however, there are problems in more variable sequences. For one every lost base counts even if a region of many bases was lost/added in a single event. It is also possible that a sister group could be supported by partially overlapping gap regions, which would be biologically difficult to understand. The GapCoder program is one way of applying so-called affine gap costs of identical gap regions in a static alignment. The assumption in this is that an event (insertion/deletion) occurred creating a region in a sequence that is recognizable by its starting point and length. This reduces the problem of overweighting a single event (likely an underestimate) and eliminates the partial overlap support problem.

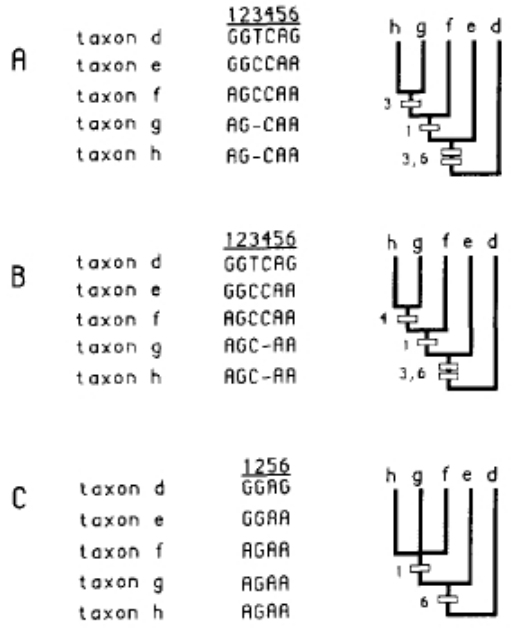

FIG. 1. Two equally costly alignments, A and B, and alignment-invariant sites, C, for hypothetical taxa (d–h). Nucleotide positions are labeled 1–6 in each alignment. Cladograms derived from each alignment are shown to the right with character changes numbered by position. The alignment-ambiguous sites, 3 and 4, support the clade g + h in the initial alignments. Support for g + h is lost by excluding the alignment-ambiguous positions.

Gatesy, J., R. de Salle, and W. Wheeler, 1993. Alignment-ambiguous nucleotide sites and the exclusion of systematic data. Molecular Phylogenetics and Evolution, 2:152-157

#	#Taxa	Data	Alignment method	# Char	# p.i. Char	# Trees	Length	CI	RI	Sister to *Hypherpes* complex	*Hypherpes* cmplx	Anill	Lepto	Hyp	mta
1	157	28S	Clustal	1356	544	12150	3126	31	77	*Cryobius*	yes	yes	yes	yes	yes
2	157	28S	Dialign	1723	571	3420	2924	32	76	*Ecsteropus*	yes	yes	no	no	yes
3	**157***	**28S**	**Clustal-GC**	**1912**	**885**	**72**	**3849**	**34**	**80**	***Cryobius***	**yes**	**yes**	**yes**	**yes**	**yes**
4	157	28S	Dialign-GC	2221	905	768	4171	31	76	*Ecsteropus*	yes	yes	no	yes	yes
5	140	28S	Clustal	1349	499	24	2684	33	77	*Gastrosticta*	yes	yes	yes	no	yes
6	140	28S	Dialign	1656	515	216	2480	34	75	*Pseudoferonina*	yes	yes	yes	yes	yes
7	140	28S	Clustal-GC	1833	801	4800	3306	36	80	large clade	yes	yes	yes	yes	yes
8	140	28S	Dialign-GC	2079	798	1080	3487	32	75	*Pseudoferonina*	yes	yes	yes	yes	yes
9	140	COI&COII	manual	1569	568	46512	3840	23	67	large clade	yes	yes	yes	yes	yes
10	140	COI&COII pos.3 off	manual	1045	184	>100K	830	29	80	*Cyclotrachelus*	yes	yes	no	no	yes
11	140	COI&COII pos.3 off. TNT Estimated Consensus	manual	1045	184	na	na	na	na	*unresolved*	yes	yes	no	no	yes
12	**140***	**28SCOICOII**	**Clustal-GC**	**3402**	**1369**	**15504**	**7283**	**28**	**73**	***Pseudoferonina+ Cryobius***	**yes**	**yes**	**yes**	**yes**	**yes**
13	140	28SCOICOII Bayesian Analysis	Clustal-GC	3402	1369	na	na	na	na						
14	15	18S	Clustal	1994	42	6	65	72	80	na	yes	na	na	na	no
15	15	CAD	manual	2231	235	8	523	55	53	na	yes	na	na	na	yes
16	15	wg	Clustal	463	56	26	126	60	53	na	yes	na	na	na	no
17	15	All sequences	Clustal-GC	3346	528	1	1214	55	57	na	yes	na	na	na	yes
18	15	All sequences Bayesian Analysis	Clustal-GC	3346	528	na	na	na	na	na	yes	na	na	na	yes

Table 1.


```
                        [00000000000000000000]
                        [00000000000000000000]
                        [00000000011111111222]
                        [12345678901234567890]

TaxonA                   AAAAAAAAAAAAAAAA000000
TaxonB                   AA-----AAA--AAAA1--100
TaxonC                   AAA--AAAAAA--AAA010010
TaxonD                   AAAGAA-AAAAGAA-A001001
TaxonE                   ACGTACGTACGTACGT000000
TaxonF                   AAAAAAAAAAAAAAAA000000
TaxonG                   AA-----AAA--AAAA1--100
TaxonH                   AAA--AAAAAA--AAA010010
TaxonI                   AAAGAA-AAAAGAA-A001001
TaxonJ                   ACGTACGTACGTACGT000000
;
END;
```

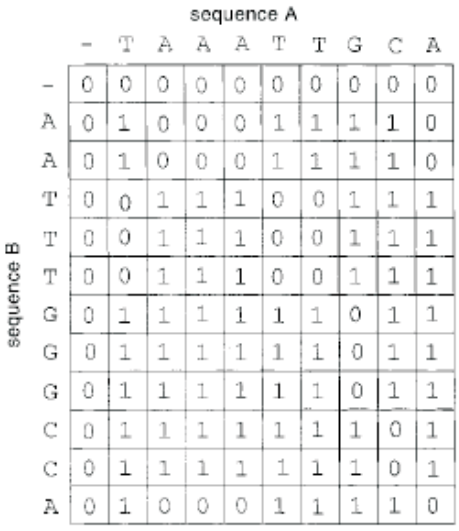

FIG. 1. An initialized matrix of a pairwise nucleotide sequence comparison with an assigned mismatch cost of 1.

matching state between sequences, regardless of position. All nonidentical states are assigned a value of one. More elaborate schemes of mismatch-scoring functions can be instituted by referring to a predefined mismatch-scoring matrix.

During the wavefront update of the matrix, each cell in the matrix is assigned a new value (Fig. 2a–2d). This value results from a comparison of three neighbors of matrix cell (i, j): the cell to the immediate left $(i, j - 11)$, directly above $(i - 1, j)$, and diagonal above and to the left $(i - 1, j - 1)$. A diagonal path implies a correspondence between sequence elements whether there is a match or a mismatch. A gap is inserted into the alignment by moving across a row or down a column. Gaps are assigned a cost (>0) or a trivial alignment will be generated with a gap at every potential mismatch (total score = 0). The new value of cell (i, j) will be the minimum path cost of the three possible across the row to the right. Cell (0, 1) has only one neighbor cell, (0, 0). Therefore, cell (0, 1) is assigned a value of 10, that being the gap cost of 10 plus the value of the previous cell (0, 0), which is zero. Cell (0, 2) is assigned a value of 20, a gap cost of 10 plus the cell value of its only neighbor (0, 1) also 10. This process continues across the row, sequentially adding the gap cost to the next cell. The procedure is repeated for column 0 as each cell in that column only has one neighbor. Cell (1, 1) is the first cell where the optimal determination of path cost is performed (Fig. 2a). For cell (1, 1), the path cost from cell (0, 1) is 20, 10 from the gap cost and 10 from the cell value of (0, 1). The same

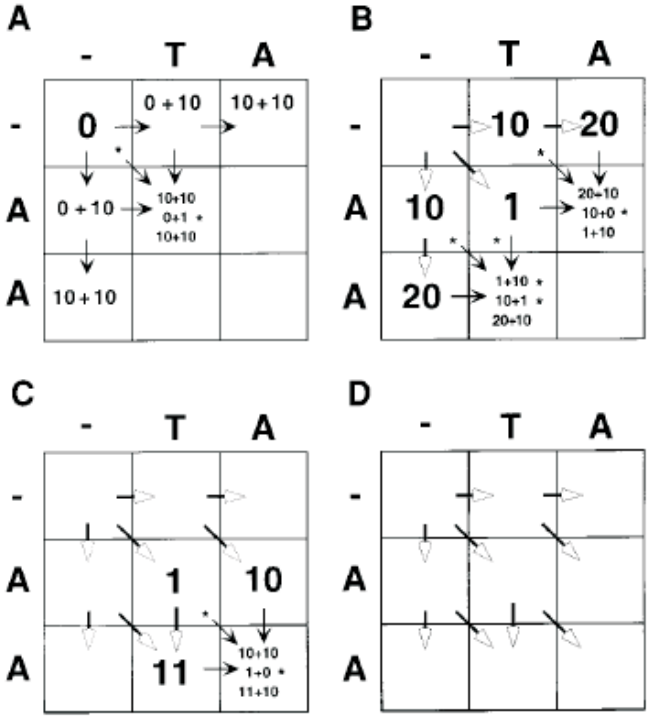

FIG. 2. Wavefront updating of the matrix elements correspond-

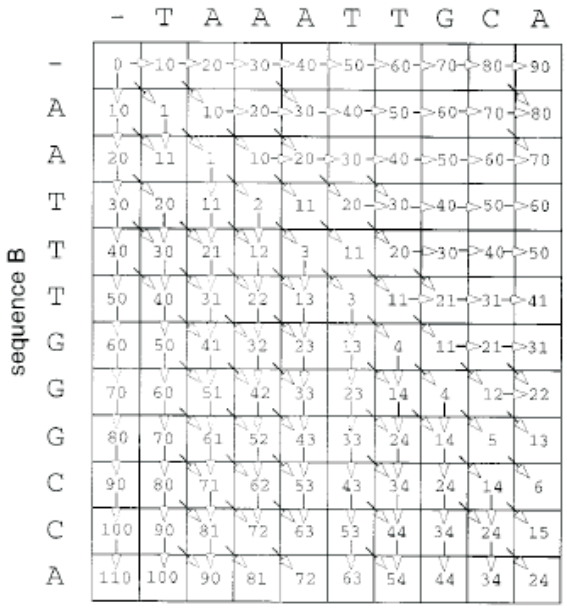

FIG. 3. A fully updated pairwise matrix of the complete sequences with the optimal paths. The cell values are retained only for

is true of the path cost from cell (1, 0), 10 from the gap cost and 10 from the cell value. The path cost from (0, 0) to (1, 1) is 1; there is no gap cost since it is on the diagonal and the cell value of cell (0, 0) is 0, but a mismatch cost is applied since cell (1, 1) represents a correspondence of an A and a T (Fig. 2a). The new cell value entered into cell (1, 1) is 1. At this time, the path from which that value was derived is logged. If two path costs are equal, an arbitrary choice is usually made, but both paths can be retained in memory. Each cell in turn undergoes this process until all cells are updated (Figs. 2b–2d, 3).

All possible alignments, whether optimal or suboptimal, are represented as pathways through the array. The traceback begins at the terminal (bottom right) element of the matrix. Any previously retained lowest path cost notation that can be connected consecutively through $(i - 1, j)$, $(i, j - 1)$, or $(i - 1, j - 1)$ is traced back through the matrix (Fig. 4). This trajectory represents a sequence of edit operations which transforms sequence A into sequence B. This edit path forms the alignment. An uninterrupted diagonal through the array would represent no gap assignments. There may be more than one optimal pathway (Fig. 5); with this particular parameter set there are six possible pathways through the matrix. The traceback procedure

the wavefront update and it is at this point that positional homology is established.

Cost functions. The gap cost and mismatch cost associated with the N-W algorithm are in a dynamic relationship; increasing mismatch cost will create more gaps in the alignment and increasing gap cost will increase the number of mismatches. Accordingly, an alignment may only be optimal for a particular combination of mismatch and gap costs. Alter these values and the optimal alignment may alter as well yielding a different phylogenetic data set. How, then, does one decide which combination of parameter sets to use? In general these choices are arbitrary. The following is a discussion of various implementations of cost functions in the N-W algorithm. There is a myriad of variations on the implementation of cost functions. Most of these implementations are attempts to mimic biological processes or constraints, which are thought to regulate the evolution of DNA or protein sequences.

Mismatch cost functions. There are many variations on the type of mismatch costs one can assign when laying out the N-W matrix. Aside from binary cost functions (0 = nucleotide match or 1 = mismatch),

alignment has six types of mismatches in a symmetri-

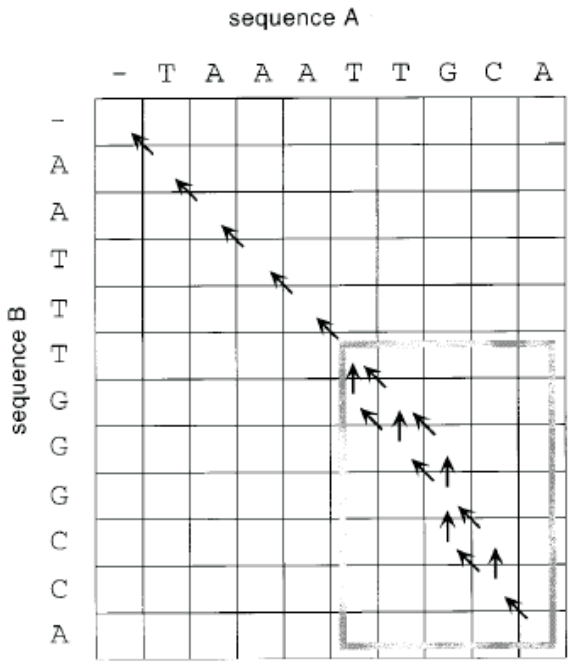

FIG. 4. The traceback procedure begins at the terminal cell