Molecular Homology and Sequence Alignment

Olaf R.P. Bininda-Emonds
AG Systematics and Evolutionary Biology
Faculty V, Institute for Biology and Environmental Sciences (IBU)
Carl von Ossietzky Universität Oldenburg
D-26111 Oldenburg
Germany

Abstract. Phylogenetic analysis depends crucially on the data underlying the analysis, with issues of data quality being but one aspect of the problem. A second, slightly less appreciated aspect is that of data comparability and specifically the evolutionary homology (= derivation from a common shared ancestry) of the individual characters being examined. The goal of sequence alignment is to establish the positional homology of the individual elements (nucleotide bases or amino-acid residues) of two or more molecular sequences (DNA or proteins, respectively). Like many other problems in computational biology, the simultaneous alignment of three or more sequences is known to be NP-hard. Fortunately, many fast and reasonable heuristics to this problem do exist, however. In this lecture, I first examine the concept of molecular homology before describing efficient algorithms for the alignment of pairs of sequences. I then expand on this to discuss potential solutions and strategies to tackle the problem of multiple sequence alignment.

Contents

1 Introduction

The use of molecular sequence data has almost entirely surpassed that of morphological data in phylogenetic systematics today for several reasons. Among these are the greater ease in obtaining large numbers of molecular characters (either for specific gene sequences or ESTs), the generally greater information content of molecular sequence data, and the apparent lack of ambiguity in character definition. The advent of low-cost, high-throughout sequencing has pushed molecular sequence data even more to the forefront such that morphological phylogenetic studies are often limited to those cases where molecular data are not available (e.g., with fossil specimens).

The decreased ambiguity of molecular sequence data stems from the limited and precisely defined nature of the characters and character states, namely the four nucleotide bases for DNA (A, C, G, and T; ignoring ambiguous base calls) and the 20 amino acids for proteins. By contrast, morphological data lacks such clearly defined characters and character states, with both having to be delineated and defined subjectively by the researcher. (That being said, efforts to formalize this procedure for morphological studies using ontologies are currently receiving a lot of attention; see [8, 39].) However, to say that molecular sequence data is unambiguous is a myth because all characters (positions in the sequences) share the same limited number of character states, meaning that identifying the same character across sequences absolutely is often obscured through convergence, mutation and/or evolutionary events that have changed the lengths of some of the sequences under examination. The act of lining up analogous positions in two or more sequences is the process of sequence alignment, a crucial and non-trivial step in phylogenetic analysis.

In this lecture, I give a brief overview of sequence alignment, focusing on the nature of the problem as well as outlining two heuristic strategies to solve this computationally difficult problem. As background to the problem, I initially introduce the concept of homology and how it applies to molecular data in particular.

2 Molecular homology

A crucial requirement underlying phylogenetic analysis is that each of the individual characters has a common evolutionary origin among the species being examined. Thus, the characters must be heritable as well as homologous. At least in an evolutionary sense, we must be comparing apples with apples and not with oranges for

our results to be accurate. It makes no sense to assess the evolutionary relationships among a group of species by comparing the feet of some against the heads of the others. Thus, although the expression of the individual characters can differ greatly among the organisms (= different types of apples), each character must still be derived from the same common ancestor (i.e., are homologs).

For molecular sequence data, homology assessment occurs at two distinct levels, that of the data partition itself (usually one or more genes or gene segments) and that of the individual nucleotides within each partition. Much more attention and theoretical work has been dedicated to the latter, but the former is equally important and deserves some explanation as well.

A relatively common phenomenon in molecular evolution is that of gene duplication, where a second copy of all or part of a gene (ignoring the increasingly fuzzy notion of what a gene exactly is) is created that has an independent evolutionary fate. Often, deleterious mutations to this copy render it as a pseudogene, a stretch of DNA with clear similarity to a working gene, but that is itself non-functional. Less frequently, mutations to the copy do not disrupt its functionality, but modify it slightly. In such case, we speak of a gene family composed of two or more individual members, usually with similar functionality and a similar molecular composition. There are numerous important gene families among organisms, many of which have played key roles in the evolution of specific groups and/or morphologies [29]. Notable examples include the Hox gene family controlling development [13], the globin gene family including myoglobin and the many forms of hemoglobin [31], and the olfactory receptor gene family [5], which is the largest known gene family in vertebrates with hundreds if not thousands of members in some species [30].

The existence of gene families complicates the issue of homology on a molecular level. Although individual members of a gene family share a common origin, it is not homology in the strictest sense. As such, a distinction is made between orthology and paralogy (although this dichotomy is a vast simplification; see [25]). The former refers to individual genes whose common evolutionary origin can be traced back to a speciation event; by contrast, the latter refers to genes that ultimately arose because of a gene-duplication event (see Figure 1). Thus, whereas orthologs can only occur in different species (e.g., myoglobin in humans and chimpanzees), paralogs can exist both within the same species (e.g., myoglobin and hemoglobin A2 in humans) as well as between species (hemoglobin A in humans compared to hemoglobin F in chimps). Thus, as the first step to molecular phylogenetic analysis, we must ensure that we are comparing orthologous gene sequences. Although several tests to distinguish orthologous from paralogous gene copies exist (e.g., [3, 7, 21, 33]), the often high similarity between the gene copies combined with incomplete gene sampling means that the tests are far from foolproof.

As a second step, the homology of the individual nucleotides within the orthologous genes needs to be determined (positional homology). Ordinarily, this would not be problematic even in the face of mutations changing the nucleotide for a specific position in a gene (e.g., from an A to a C). So long as we can be assured that the gene is of a constant length across species, we know that any nucleotide mismatches

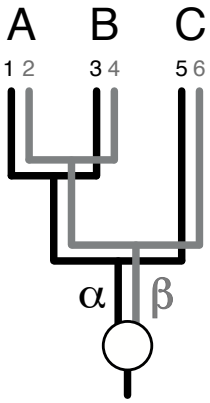

Figure 1: Distinguishing between orthology and paralogy. The ancestral gene for the species A, B, and C underwent a gene duplication event (open circle), leading to α (black; odd numbers) and β (grey; even numbers) copies of the gene, each evolving independently from one another. All genes with either odd or even numbers are orthologs of one another. Any odd-numbered gene is a paralog of any even-numbered gene (and vice versa).

derive from substitution events (a substitution simply being a mutation that has become established in a species; see [20]). Instead, the difficulty arises because of insertion-deletion events (indels) that either add or remove one or more nucleotides in a gene compared to other orthologs, thereby changing the gene length. Combined with the limited number of character states for DNA sequence data, it can often be difficult to tease apart the effects of substitution versus indels in explaining nucleotide mismatches between sequences. For instance, nucleotide A in position 160 of one gene could correspond to nucleotide T in position 160 of another orthologous copy (substitution event), to nucleotide A in position 153 (indel event), or even to nucleotide G in position 169 (substitution + indel events)! (See Figure 2 for an example of alignments based on substitutions and/or indels.) The process of establishing the positional homology between orthologous gene sequences is known as (multiple) sequence alignment.

3 Multiple sequence alignment (MSA)

3.1 Reconciling substitutions and indels

As noted above, differences between two sequences can obtain from substitution and/or indel events, with the roles played by either often being difficult to tease apart. Resolving this conflict depends on the use of a scoring function detailing (in its simplest form) the relative costs of these two events. Based on this scoring function (σ), it is possible to determine the cost of any given alignment (the score of the align-

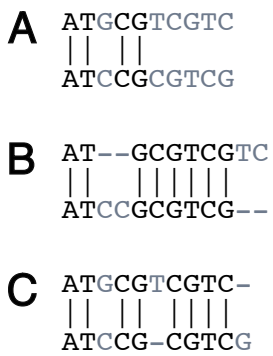

Figure 2: A set of simple alignments, with matches indicated by vertical lines and mismatches in grey text. Mismatches are accounted for either by (A) substitutions only, (B) gaps (indels) only, or (C) a combination of substitutions and gaps. Alignments (B) and (C) are equally optimal in terms of the number of matches.

ment) by counting up the numbers of each event multiplied by their relative costs. This, in turn, provides a mechanism to compare different alignments of the same set of sequences and thereby to identify the optimal alignment(s) for that scoring function.

The simplest scoring function involves giving different relative weights to substitutions (usually set to 1) and indels (specifically the gaps that they produce in some sequences), with the latter usually penalized with respect to the former. However, it is possible to expand the complexity of this scoring function enormously by assigning different costs to the types of substitution (e.g., transitions versus transversions), to opening a gap versus extending an existing gap, and to the cost of a gap in relation to the proximity of other gaps and/or the start or end of the sequence. (Note that the latter is largely an artifact of it often being computationally more optimal to place gaps near the ends of a sequence rather than these positions naturally containing more gaps. In reality, because DNA is continuous over large stretches, there are no true ends of the sequence in most cases.) In fact, so-called affine gap costs [1, 15] are commonly implemented in the form of

$$\text{gap cost} = \text{open cost} + (\text{extension cost}) \times (\text{length of gap}), \tag{1}$$

where open cost > extension cost.

The effect of incorporating affine gap costs in a scoring function is to prefer alignments where numerous, smaller gaps are preferentially fused into fewer, larger gaps, something that seems to be biologically reasonable. However, it must be said at this point that all scoring functions, regardless of how complex, remain oversimplifications that attempt to apply subjective weights more or less globally across an alignment. Moreover, the exact values used within the scoring function tend to be based on their success in achieving reasonable results rather than to any real understanding on our part of the relative frequencies of substitutions versus indels.

Until now, the implicit assumption is that we are aligning DNA sequences. How-

ever, it is equally possible to align amino-acid sequences. Indeed, the first applications of MSA were geared towards this problem given that early molecular sequences were more often amino-acid rather than nucleotide based due to the comparative ease of sequencing the former at the time. Ideally, the scoring functions here should account for the DNA substitutions underlying the observed amino-acid transitions. However, this is difficult to model accurately given the degeneracy of the genetic code, such that some amino acids can be encoded by greater than one triplet of DNA nucleotides. Instead, the most commonly used scoring functions for amino acids combine a gap penalty function with empirically observed amino-acid substitution frequencies derived from a set of proteins in closely related species and expressed as a matrix of log-odds scores. The two most common matrices are the PAM (Point Accepted Mutation) matrices of Dayhoff [9] and the BLOSUM (BLOcks of Amino Acid SUbstitution Matrix) of Henikoff and Henikoff [17]. Both matrices also exist in numerous variants designed to be applied to more closely or more distantly related sequences, the number behind each variant revealing its approximate level of application (close: low PAM, high BLOSUM; distant: high PAM, low BLOSUM). Numerous other matrices also exist as well as isolated attempts at some actual mechanistic models (e.g., [44]), but BLOSUM62 is arguably the standard one [10], generally showing the best performance. Thus, it is often used by default in many applications, including the NCBI implementation of the BLAST algorithm (Basic Local Alignment Search Tool; see below) [2]. Interestingly, an apparent error exists in the initial BLOSUM62 matrix [35], but one that has the effect of improving the performance of the method!

3.2 Alignment algorithms

3.2.1 Pairwise sequence alignment

Sequence alignment represents a difficult problem computationally. For instance, about 1060 different alignments exist for a pair of amino-acid sequences each with 100 residues. Many of these alignments will naturally be patently absurd; however, the example does illustrate the scale of the problem, even in a very simple case. Therefore, solutions for MSA derive from the computationally simpler problem of the pairwise alignment of only two sequences.

The heart of pairwise alignment algorithms lies in a dot matrix of the two sequences. Where two sequences share the same residue at a position, a dot is placed in the matrix (see Figure 3A). As such, dot matrices present a useful tool to visualize regions of similarity between two sequences as well as indels or repetitive motifs. The optimal path(s) through this matrix also represent(s) the optimal alignment(s) for the two sequences. An efficient solution to finding this path was first solved using dynamic programming by Saul Needleman and Christian Wunsch in 1970, a solution that has come to be known as the Needleman-Wunsch algorithm [28]. Not only does the algorithm find the optimal path(s), but does so quickly, running in $\mathcal{O}(nm)$ time (also representing its memory requirement), where n and m represent the lengths of the two sequences.

To implement the algorithm, the cells of the matrix are labeled initially not with

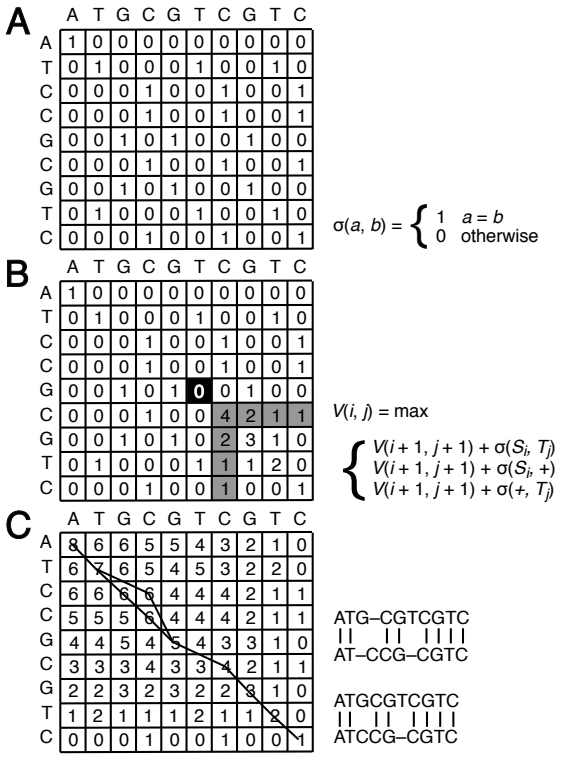

Figure 3: A worked example of the Needleman-Wunsch [28] dynamic programming algorithm for pairwise sequence alignment. **A.** Two sequences are compared in a matrix with the cells filled according to the desired scoring function (here 0 for a mismatch, 1 for a match). **B.** Starting in the lower right corner, the value of each cell (e.g., the black cell) is added to the maximal values of all cells in the next higher row and column (e.g., all grey cells). In this example, the black cell would receive a final value of 4. **C.** A path is iteratively drawn from the highest scoring cell to the highest scoring cell in the next higher row and column. If two or more cells share the highest value, multiple paths are indicated (as here). The path or paths describe the optimal pairwise alignment for the pair of sequences for the given scoring function.

dots, but with values corresponding to the desired scoring function (e.g., in the simplest case, 1 for a match and 0 for a mismatch; see Figure 3). The cells of the matrix are then rescored recursively starting from the lower right corner according to the following:

$$V(i,j) = \max\big[\, V(i+1,j+1) + \sigma(S_j, T_j), V(i+1,j+1) + \sigma(S_i, +), \\ V(i+1,j+1) + \sigma(+, T_j)\,\big], \tag{2}$$

where i, j = i-th and j-th elements of a given row (S) or column (T), respectively,

and σ = scoring function of cell $V(i, j)$.

In other words, the final score for each cell equal its initial score plus the maximal score among any of the cells found in the next row or one column of the matrix. (A slight modification also enables the algorithm to proceed from the upper left corner.) Starting from the cell(s) with the highest score, the optimal path through the matrix is traced by always moving to the cell in the next row or column with the highest score. Because more than one cell might possess the highest score, multiple, equally optimal paths (= alignments) might exist, see Figure 3C.

3.2.2 A localized digression: Smith-Waterman and BLAST

In providing a global alignment of two sequences, the Needleman-Wunsch algorithm will ignore any local regions of high similarity in favour of the globally optimal solution. However, these local regions can often be of particular interest, potentially representing conserved protein domains or motifs with specific functions that are used as building blocks for constructing proteins with different functions. Moreover, similarities between distantly related sequences, even if homologous, are often clouded through accumulated mutations in the DNA sequence. By focusing instead on local regions of high similarity, the relationship between such sequences can still be teased out. Identifying local, highly similar regions can be found using a slight variant of the Needleman-Wunsch algorithm developed by Temple Smith and Michael Waterman in 1981, the Smith-Waterman algorithm [34].

In principle, the Smith-Waterman algorithm proceeds analogously to the Needleman-Wunsch algorithm apart from three key differences. First, mismatches are now given a negative value in the scoring function instead of zero as in Needleman-Wunsch. Second, despite the difference in the scoring function, the minimum value during the reweighting step remains zero. (To achieve this, another term equal to zero is added to the reweighting function for Needleman-Wunsch given above.) Finally, the start and end points of the matrix traversal can lie anywhere in the matrix and must not proceed diagonally from the highest scoring corner. Instead, one starts with the highest scoring cell (note that there be many equally high scoring cells) and proceeds analogously to Needleman-Wunsch until a cell with value of zero is reached. The potential difference in the results that can be obtained between the Smith-Waterman and Needleman-Wunsch algorithms is illustrated in Figure 4.

One desirable property of a local versus global alignment is that only the former has an underlying statistical model based on a comparison of unrelated sequences where the optimal local alignment scores tend to follow an extreme value distribution. This property, in turn, enables an estimate (expectation value) of the degree of relatedness between two test sequences. In particular, one can assess how often two sequences from the null distribution of unrelated sequences will yield a local alignment score that is greater than or equal to that from the two test sequences. When this event is unlikely (= low expectation value), then evidence exists that the test sequences might be homologous or share homologous elements (e.g., protein domains or motifs).

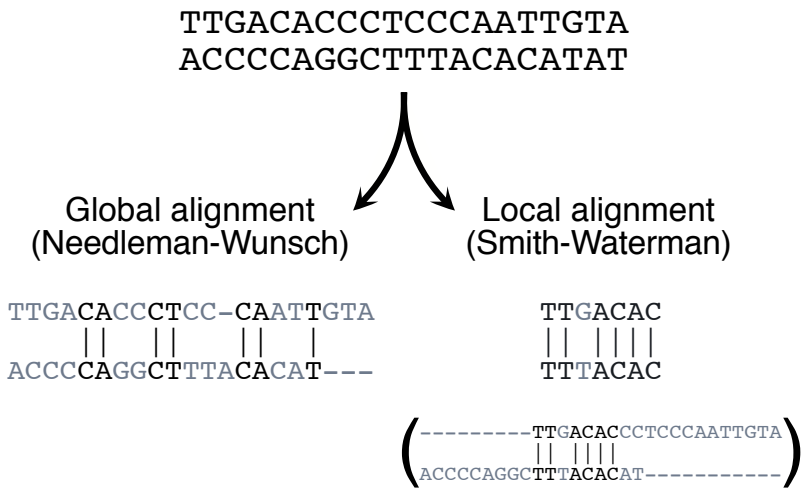

Figure 4: A comparison of the results of the Needleman-Wunsch (global) and Smith-Waterman (local) alignment algorithms. Although Needleman-Wunsch does obtain more matches between the query sequences (seven versus six), they are interspersed with more gaps. The matches found by Smith-Waterman are more concentrated, with non-matching parts of the sequences discarded. Note as well that the regions of similarity identified between the sequences differ between the two algorithms.

Even with its polynomial running time and various algorithmic improvements, Smith-Waterman remains too slow and memory intensive to enable queries of a test sequence against large sequence databases like GenBank (`www.ncbi.nlm.nih.gov/genbank/`) to determine its potential identity through highly similar sequences. Instead, a heuristic approximation of Smith-Waterman, the BLAST algorithm based on high-scoring segment pairs, is commonly used for this purpose. What BLAST lacks in accuracy compared to Smith-Waterman (it is not guaranteed to find the optimal local alignment), it makes up for in speed, being at least some 50x faster in its basic implementation (some variants are even faster). However, this tradeoff between speed and accuracy is necessary when searching through databases like GenBank with its 124,516,775,718 bases of information contained in 132,302,771 separate sequence records (in the traditional GenBank divisions as of March 2011). (See Figures 5 and 6 for screenshots of a BLAST search of GenBank conducted using the NCBI servers.) Moreover, BLAST offers great flexibility, actually existing as a family of algorithms depending in part whether the query and reference sequences are formatted as DNA (nucleotides) or amino acids (proteins). Using translated searches, it is even possible to search one format against the other. Together, BLAST and its many variants undoubtedly represent one of the most frequently used bioinformatics tools today. Typical uses include identifying the species or group from which an unknown sequence might come from (useful for environmental sampling studies where the source

organisms are unknown), identifying protein domains within the query sequence, establishing gene families or mapping gene annotations between organisms, or localizing the genomic location of the query sequence.

3.2.3 Applying pairwise sequence alignment to MSA: progressive and iterative alignment

The Needleman-Wunsch algorithm provides an efficient solution to the problem of pairwise sequence alignment. It is also flexible in the sense that it can be used for both DNA and amino-acid sequences and for a variety of scoring functions for either. It is therefore tempting to think that the NP-hard problem of multiple sequence alignment could be conquered by repeated pairwise application of Needleman-Wunsch. Unfortunately this is not the case. For MSA, recursive Needleman-Wunsch runs on the order of $\mathcal{O}(Nm)$ time, where N = length of the sequences and m = number of sequences. Consider the following. If it takes one second to align two sequences of 100 residues using the Needleman-Wunsch algorithm, it would require 100^2 seconds to align three sequences, 100^3 seconds to align four, and so on. Aligning only ten sequences would require 10^{18} seconds, which is more time than the universe has existed. Analogous to BLAST and Smith-Waterman, another, heuristic strategy is obivously required for MSA.

In developing a useful heuristic for MSA, two factors are important to account for: 1) similar sequences are easier to align than less similar ones, and 2) pairs of sequences are easier to align than multiple ones. Thus, one solution to the problem of MSA is to align the most similar sequences in a pairwise fashion, which is exactly the strategy underlying progressive alignment [12]. As exemplified by its implementation in the MSA program Clustal [18, 23], progressive alignment uses a three-step procedure: 1) a fast pairwise alignment of all possible pairs of sequences, 2) construction of a phylogenetic tree based on the matrix of pairwise alignment scores from step 1, and 3) the final pairwise alignment of sequences or clusters of sequences in their decreasing order of similarity (see Figure 7). The key to the strategy lies in steps 2 and 3. In step 2, the phylogenetic tree is typically constructed using a fast distance-based method like neighbour joining (NJ; [32]) to provide a guide tree for choosing the most similar (clusters of) sequences to be aligned in turn in step 3. Progressively dissimilar pairs of sequences or clusters thereof are continually selected and aligned until the global alignment is obtained.

The trick behind progressive alignment lies in large part in the last step, where not only individual sequences are pairwise aligned, but also clusters of sequences resulting from previous alignment operations and that the alignment of these clusters is fixed internally. In other words, once the sequences X and Y are aligned to one another to form the sequence cluster XY, the alignment of X and Y relative to one another cannot be altered. If a subsequent alignment step requires a gap to be added to X, the same gap must be added to Y (and any other members of the cluster). This restriction has the advantage of speeding up the entire alignment procedure dramatically (because not all possible pairs of sequences are being examined and adjusted), but at the cost of not being able to retroactively correct an alignment

Figure 5: Screenshots from the NCBI webpages (http://blast.ncbi.nlm.nih.gov/) showing the results of a BLAST search using the CO1 gene sequence from the rotifer *Seison nebaliae* (NCBI accession number DQ297765) as a query sequence. Searching against the entire nucleotide collection of GenBank took less than 15 seconds. **A.** Graphical representation of the results showing local regions of similarity colour coded with respect to the alignment score. **B.** Table of BLAST results showing alignment scores, degree of overlap ("query coverage") and expectation values ("E value"). The same sequence came back unsurprisingly as the best hit. However, all but one of the remaining sequences do not come from rotifers, hinting that the taxonomic affinity of the query sequence might have been misidentified originally. (Continued in Figure 6.)

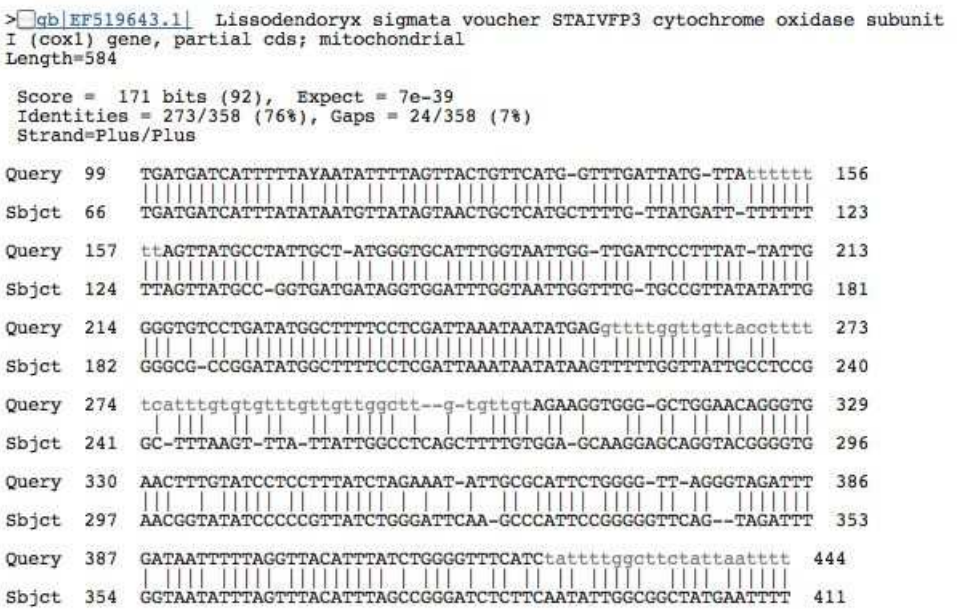

Figure 6: (Continuation of Figure 5). Sequence comparison between the query sequence and the last entry in (B) showing sequence mismatches and gaps. Note that this is clearly a local alignment in that the entire query sequence (660 nucleotides) is not represented.

based on subsequent information derived from other sequences. Alignment errors deriving from the latter are more common when the sequences being aligned are distantly related and therefore subject to random similarities. To help counteract for this shortcoming, many progressive alignment programs include a secondary weighting function to correct for the evolutionary distance between the sequences being aligned.

A derivative of progressive alignment are iterative methods that periodically revisit and realign sequences within previously aligned clusters to optimize a global objective function (e.g., the global alignment score). These methods thus directly address the one major weakness of a pure progressive alignment approach, but are not necessarily slower as a result. For instance, one the most popular iterative programs, MUSCLE [11], is demonstrably faster than Clustal and often delivers better alignments as well.

The overview I provide here is admittedly overly simplistic. There are many, many MSA programs, each sporting a variety of different options and features, including the ability to customize the scoring function for the alignment (and often in an excruciatingly detailed manner). Other programs based on hidden Markov models, genetic algorithms and simulated annealing, or motif building also exist. Even the BLAST algorithm for quickly determining regions of local similarity, which uses a fast heuristic of the Smith-Waterman algorithm [34], can be used for MSA. Each of these programs

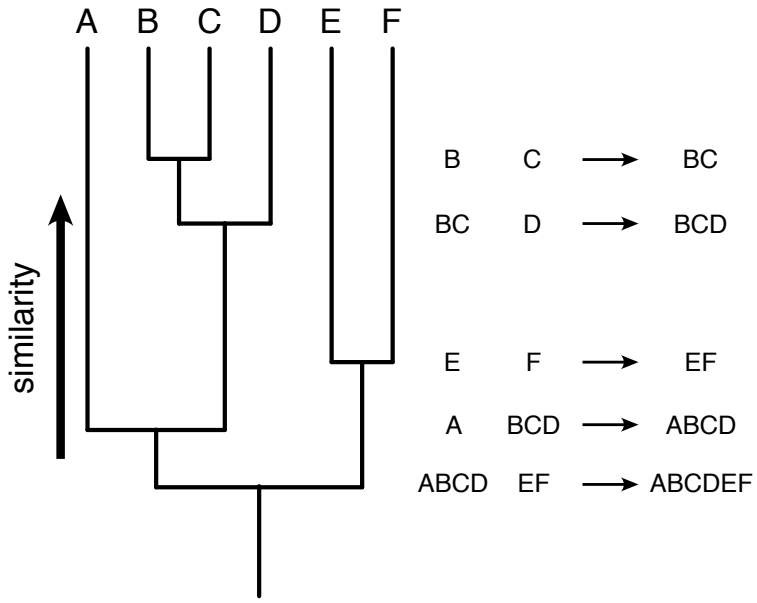

Figure 7: An example of how the guide tree is used during progressive alignment. The tree is traversed downwards from the tips (i.e., in decreasing order of similarity), with pairs of sequences or sequence clusters being aligned at each node until the root of the tree is reached and therefore the global alignment is obtained.

has their strengths and weaknesses and no clear-cut consensus exists as to the best program. My personal experience is that the performance of the different programs is often dependent on the data set and for no obvious reason. For instance, although MUSCLE typically outperforms Clustal in terms of both speed and accuracy, Clustal will occasionally deliver better alignments than MUSCLE for certain data sets. Indeed, despite being one of the oldest MSA programs available, the Clustal family of alignment programs (ClustalW and Clustal X; the important difference being that the latter sports a graphical user interface) remains one of the most popular tools for MSA and is implemented as both a standalone program as well as being found on numerous web servers.

3.2.4 A different perspective: simultaneous optimization

The traditional approach to molecular phylogenetic studies is to first determine the MSA and then to use it as a fixed data set for the phylogenetic analysis. A potential shortcoming here is that the resulting phylogenetic tree is often highly dependent on the sequence data (and therefore alignment) underlying it. However, there are many reasons to question the accuracy of any given alignment: the scoring functions used are ultimately subjective; progressive alignments are often based on NJ or UPGMA trees,

which are usually less accurate than other criteria for phylogenetic inference [19, 36]; and the NP-hard nature of MSA means that the optimal alignment is simply unlikely to be found. Thus, some have argued that basing the phylogenetic analysis on only a single, fixed alignment might not be desirable [40,43], especially given that differences arising from the use of different alignments are occasionally greater than those arising from the use of different phylogenetic optimization criteria (e.g., see [27]).

Instead, given that the analysis is already optimizing at least the tree topology, why not have it optimize the alignment at the same time? The preferred alignment is then the one yielding the most optimal tree topology. This is the idea underlying optimization alignment (also known as dynamic homology) [40,42] and the related fixed-state optimization [41], the latter being an extension of the former that essentially respect boundaries between partitions (e.g., genes) within the data set. Both methods are implemented in the comprehensive, open-source program POY [38], which includes standard substitution plus affine gap costs for the alignment scoring function as well as both maximum parsimony and maximum likelihood as phylogenetic optimization criteria. The program is also able to perform combined analyses of morphological and molecular data.

As theoretically appealing as optimization alignment is, the method has failed to find general favour among the phylogenetic community, even in light of comparative studies showing that it often yields tree topologies that are more optimal than those obtained from the phylogenetic analysis of a fixed alignment [43]. The reasons for this are unclear. In part, it may stem from the alignment procedure still being based on a subjective scoring function combined with a somewhat black-box approach in which the alignment (representing the input data) ultimately does not derive from and cannot be altered by the researcher. This situation goes against a long tradition in phylogenetic systematics (if not science in general) of ensuring that the input data are as robust as possible in terms of character definition and homology assessment.

The focus on optimization also means that the final alignment need not be entirely biologically sensible, merely optimal, a problem that afflicts any alignment procedure based solely on a scoring function. Fixed-state optimization will prevent the nucleotides of one partition from invading that of another as a result of the alignment process; however, somewhat nonsensical (but optimal) alignments within the individual partitions can still occur. The effects of these errors can be counteracted to some degree by rerunning the analysis many times using different scoring functions as a form of sensitivity analysis to find those relationships that are stable to different alignments. (This strategy is also equally applicable and desirable for progressive alignment based analyses.)

Finally, optimization alignment combines two computationally difficult problems alignment and phylogeny reconstruction (both of which are NP-hard) in the same analysis and then performs both numerous times. Thus, the method is more computationally intense than the traditional method of feeding the phylogenetic analysis a single, preferred alignment (for an empirical comparison, see [22]). Even so, POY shows surprisingly good performance even on normal desktop PCs despite this additional computational burden.

4 Conclusions

I conclude by giving four personal (and subjective) tips to aid in performing MSA.

1. Never trust automated alignments.

The simple fact is that all automated alignment programs use subjective, overly simplistic, and restrictive cost functions and so will all make mistakes to varying degrees. No alignment program is as good as the human eye, although the latter is quickly overwhelmed by large, somewhat noisy alignments. Therefore, a good general strategy is to perform an automated alignment initially (using any program and a reasonable scoring function) and then to manually improve the alignment. The exact choice of alignment program / scoring function here is somewhat secondary. They are simply being used as tools to deliver a reasonable starting alignment (although better alignments mean less work subsequently). Some researchers reject the notion of manual adjustment out of hand, feeling it to be too subjective and non-repeatable (e.g., [22]). However, it is no more subjective than the cost functions for the different programs. Ideally, if time allows, it would be best to perform a sensitivity analysis using different alignments derived from different scoring functions (e.g., [24]). However, when time is in short supply (which is usually the case), it is often preferential to use the best alignment possible.

2. Use secondary information wherever possible.

The use of additional, meta-information can often greatly ease the process of MSA. Such information can exist in the form of structural (e.g., secondary structure or amino-acid translations) or taxonomic information (realizing that orthologous sequences will usually be more similar in more closely related species).

The use of amino-acid translations for aligning protein coding DNA data (translated alignments) is particularly desirable for numerous reasons (for a summary, see [4]). First, DNA-based alignments of such sequences ignore important structural information in the form of the triplet codon organization. Thus, gaps should have lengths of multiples of three to avoid introducing frame shifts and spurious internal stop codons in the amino-acid translation. This is typically not guaranteed when performing a DNA-based alignment (although the scoring function could be modified to enforce this restriction). Second, the scoring function for amino-acid alignments are based on empirical transition frequencies rather than subjective relative weights and therefore tend to be more biologically realistic, despite lacking a clear model to explain these observed frequencies. Third, the amino-acid sequence is more conserved evolutionarily than is the DNA sequence and also utilities a larger alphabet of 20 amino acids. Thus, it is often possible to align more distantly related sequences using a translated alignment where a DNA based alignment simply fails or is more difficult or ambiguous. Finally, obtaining the translated alignment is much faster because the translated sequences are one-third as long as the associated DNA sequence. Given that the running time of the Needlemann-Wunsch algorithm is proportional to the product of the length of the two input sequences, the potential time savings can be

up to an order of nine times (see [4]). The option to perform translated alignments of a coding DNA sequence exists today in most major MSA programs and alignment editors.

3. Dont be afraid to throw out very noisy, unalignable blocks out of the alignment.

There will often be cases where MSA of specific blocks will be next to impossible: the mutation and indel rates are too high and/or the species are too distantly related to discover any apparent structure to the jumble of letters. Rather than input these questionable data into the analysis (which would contribute only noise and a possibly misleading or disruptive signal), it is arguably more desirable to delete such unalignable regions outright (but see [14]) such that the results are based solely on well supported, homologous data. Indeed, there has been a growing trend in the phylogenetic community supporting this idea and that of removing noisy data in general [37]. Yet, despite the adverse effects poorly aligned regions can have on the outcome of the phylogenetic analysis [16], surprisingly little effort has gone into ways to identify such regions. Two notable exceptions here are the programs GBlocks [6] and ALISCOR [26] (see also [14]), which attempt to identify blocks suffering both from being poorly aligned and/or containing highly divergent sites. As with most other aspects of MSA, however, the identification and delineation of noisy blocks is also subjective in nature, such that it is important not to trust these tools absolutely and to double check their results. The simultaneous optimization of the alignment and the tree topology under optimization alignment renders the exclusion of unalignable regions in such analyses as difficult, if not impossible given that the tree optimality scores will no longer be comparable with the various exclusion of some of the underlying data.

4. Alignment is more art than science and is something that needs to be learned.

A clear message throughout this lecture is that alignments, except when they are obvious, are more or less subjective. Doing good alignments requires a practiced eye and sufficient experience with different genes to know which patterns are more common than others. As such, there is no better strategy for obtaining good alignments than to do them oneself and to practice, practice, practice.

References

[1] S. F. Altschul and B. W. Erickson. Optimal sequence alignment using affine gap costs. *Bull. Math. Biol.*, 48:603-616, 1986.

[2] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215:403-410, 1990.

[3] L. Arvestad, A. C. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19 (Suppl 1):i7-i15, 2003.

[4] O. R. P. Bininda-Emonds. transAlign: using amino acids to facilitate the multiple alignment of protein-coding DNA sequences. *BMC Bioinformatics*, 6:156, 2005.

[5] L. Buck and R. Axel. A novel multigene family may encode odorant receptors: a molecular basis for odor recognition. *Cell*, 65:175-187, 1991.

[6] J. Castresana. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Mol. Biol. Evol.*, 17:540-552, 2000.

[7] J. C. Chiu, E. K. Lee, M. G. Egan, I. N. Sarkar, G. M. Coruzzi, and R. DeSalle. OrthologID: automation of genome-scale ortholog identification within a parsimony framework. *Bioinformatics*, 22:699-707, 2006.

[8] W. M. Dahdul, J. G. Lundberg, P. E. Midford, J. P. Balhoff, H. Lapp, T. J. Vision, M. A. Haendel, M. Westerfield, and P. M. Mabee. The teleost anatomy ontology: anatomical representation for the genomics age. *Syst. Biol.*, 59:369383, 2010.

[9] M. O. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. In M. O. Dayhoff (ed), *Atlas of Protein Sequence Structure*, pages 345-352, National Biomedical Research Foundation, Washington, D.C., 1978.

[10] S. R. Eddy. Where did the BLOSUM62 alignment score matrix come from? *Nature Biotechnol.*, 22:1035-1036, 2004.

[11] R. C. Edgar. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucl. Acids Res.*, 32:1792-1797, 2004.

[12] D. F. Feng and R. F. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 25:351-360, 1987.

[13] J. Garcia-Fernàndez. The genesis and evolution of homeobox gene clusters. *Nature Rev. Genet.*, 6:881-892, 2005.

[14] J. Gatesy, R. DeSalle, and W. Wheeler. Alignment-ambiguous nucleotide sites and the exclusion of systematic data. *Mol. Phylogenet. Evol.*, 2:152-157, 1993.

[15] O. Gotoh. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162:705-708, 1982.

[16] S. Gribaldo and H. Philippe. Pitfalls in tree reconstruction and the phylogeny of eukaryotes. In R. P. Hirt and D. S. Horner (eds), *Organelles, Genomes and Eukaryote Phylogeny*, pages 133-152, CRC Press, Boca Raton, 2004.

[17] S. Henikoff and J. G. Henikoff. Amino acid substitution matrices from protein blocks, *Proc. Natl Acad. Sci. USA*. 89:10915-10919, 1992.

[18] D. G. Higgins and P. M. Sharp. CLUSTAL: a package for performing multiple sequence alignment on a microcomputer. *Gene*, 73:237-244, 1988.

[19] D. M. Hillis, J. P. Huelsenbeck, and C. W. Cunningham. Application and accuracy of molecular phylogenies. *Science*, 264:671-677, 1994.

[20] S. Y. Ho and G. Larson. Molecular clocks: when times are changin'. *Trends Genet.*, 22:79-83, 2006.

[21] R. Jothi, E. Zotenko, A. Tasneem, and T. M. Przytycka. COCO-CL: hierarchical clustering of homology relations based on evolutionary correlations. *Bioinformatics*, 22:779-788, 2006.

[22] K. M. Kjer, J. J. Gillespie, and K. A. Ober. Opinions on multiple sequence alignment, and an empirical comparison of repeatability and accuracy between POY and structural alignment. *Syst. Biol.*, 56:133-146, 2007.

[23] M. A. Larkin, G. Blackshields, N. P. Brown, R. Chenna, P. A. McGettigan, H. McWilliam, F. Valentin, I. M. Wallace, A. Wilm, R. Lopez, J. D. Thompson, T. J. Gibson, and D. G. Higgins. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23:2947-2948, 2007.

[24] D. R. Maddison, M. D. Baker, and K. A. Ober. Phylogeny of carabid beetles as inferred from 18S ribosomal DNA (Coleptera: Carabidae). *Syst. Entomol.*, 24:103-138, 1999.

[25] D. P. Mindell and A. Meyer. Homology evolving. *Trends Ecol. Evol.*, 16:434-440, 2001.

[26] B. Misof and K. Misof. A Monte Carlo approach successfully identifies randomness in multiple sequence alignments: a more objective means of data exclusion. *Syst. Biol.*, 58:2134, 2009.

[27] D. A. Morrison and J. T. Ellis. Effects of nucleotide sequence alignment on phylogeny estimation: a case study of 18S rDNAs of Apicomplexa. *Mol. Biol. Evol.*, 14:428-441, 1997.

[28] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48:443-453, 1970.

[29] M. Nei and A. P. Rooney. Concerted and birth-and-death evolution of multigene families. *Annu. Rev. Genet.*, 39:121-152, 2005.

[30] Y. Niimura and M. Nei. Extensive gains and losses of olfactory receptor genes in mammalian evolution. *PLoS One*, 2:e708, 2007.

[31] A. Roesner, C. Fuchs, T. Hankeln, and T. Burmester. A globin gene of ancient evolutionary origin in lower vertebrates: evidence for two distinct globin families in animals. *Mol. Biol. Evol.*, 22:12-20, 2005.

[32] N. Saitou and M. Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4:406-425, 1987.

[33] M. J. Sanderson, A. C. Driskell, R. H. Ree, O. Eulenstein, and S. Langley. Obtaining maximal concatenated phylogenetic data sets from large sequence databases. *Mol. Biol. Evol.*, 20:1036-1042, 2003.

[34] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195-197, 1981.

[35] M. P. Styczynski, K. L. Jensen, I. Rigoutsos, and G. Stephanopoulos. BLOSUM62 miscalculations improve search performance. *Nature Biotechnol.*, 26:274-275, 2008.

[36] D. L. Swofford, P. J. Waddell, J. P. Huelsenbeck, P. G. Foster, P. O. Lewis, and J. S. Rogers. Bias in phylogenetic estimation and its relevance to the choice between parsimony and likelihood methods. *Syst. Biol.*, 50:525-539, 2001.

[37] G. Talavera and J. Castresana. Improvement of phylogenies after removing divergent and ambiguously aligned blocks from protein sequence alignments. *Syst. Biol.*, 56:564-577, 2007.

[38] A. Varón, L. S. Vinh, and W. C. Wheeler. POY version 4: phylogenetic analysis using dynamic homologies. *Cladistics*, 26:72-85, 2010.

[39] L. Vogt. The future role of bio-ontologies for developing a general data standard in biology: chance and challenge for zoo-morphology. *Zoomorphology*, 128:201217, 2009.

[40] W. C. Wheeler. Optimization alignment: the end of multiple sequence alignment in phylogenetics? *Cladistics*, 12:1-9, 1996.

[41] W. C. Wheeler. Fixed character states and the optimization of molecular sequence data. *Cladistics*, 15:379-386, 1999.

[42] W. C. Wheeler. Homology and the optimization of DNA sequence data. *Cladistics*, 17:S3-S11, 2001.

[43] W. C. Wheeler. Alignment, dynamic homology, and optimization, in V. A. Albert, ed., *Parsimony, Phylogeny, and Genomics*, Oxford University Press, Oxford, 2006, pp. 71-80.

[44] Z. Yang, R. Nielsen and M. Hasegawa. Models of amino acid substitution and applications to mitochondrial protein evolution. *Mol. Biol. Evol.*, 15:1600-1611, 1998.