

Lecture 14: Chaining of Local Alignments, Protein Profile HMMs and Classification

Lecturer: Serafim Batzoglou
Scriber: Pegah T Afshar
March 2nd, 2009

This lecture was mainly about Multiple Sequence Alignment.

➤ Evolution at the DNA level

Here we want to see how we can construct multiple sequence alignment when we have multiple genome or protein sequences. As it was mentioned in previous lectures, evolutions at the DNA level can be categorized as microscopic or large scale evolutions. In microscopic evolution, we have basepair mutations, deletions or insertions of few letters (figure 1) and in larger scale, we have sequence rearrangements that its size can vary from inversion or duplication of a few hundreds of letters to much longer length such as the whole chromosome duplication. Also each type of the evolution (e.g. duplication, inversion) has different frequency and can happen in the body or when inheriting them from parents. Therefore both species divergence and duplications can evolve proteins and genes.

The protein and gene duplication is one of the most important evolutionary processes as one of the duplicated genes or proteins has the opportunity to evolve and take a different function while the other one keeps its original function that is still important for the organism.

Phylogenetic tree depicts these different evolutionary events. In the tree, duplications are shown by rectangular nodes and species divergences by circular nodes (figure 2).

- Orthology and Paralogy

In talking about the relationship between two sequences we assume that they have a common ancestral sequence at some point in their evolutionary history. This is called sequence homology (sequence similarity).

We can define two different type of homology based on the actual history that led to this similarity. We check the first common ancestor of two the proteins in the phylogenetic tree, if the first common ancestor is referred to species divergence we call the proteins Ortholog. If the first common ancestor is referred duplication we call them paralog. We can have paralogy between two types of a protein in the same species or different ones. For instance in figure 3, HA1 Human and HA2 Human are paralog but Yeast is ortholog with all the other proteins. Besides duplication and species divergence, Xenology is another way of acquiring DNA, such as integration of retroviruses to organism's DNA (e.g. HIV). In this case we have xenolog DNA between virus and that organism.

In paraology we have two different types: Inparalogy and Outparalogy (figure 4) which depends on the phylogenetic scope that we compare the sequences. In Outparalogy, duplication happened before any species divergence but in inparalogy there is no species divergence after the duplication. For instance in figure 3 Human proteins are inparalog but between HB Human and worm, we have duplication before human-worm species divergence, so it is out-paralogy.

In orthology, sequences after species divergence have their own duplications and some copies keep their positions relative to the entire region, while the other copies move to random locations. These fixed copies are called toportholog and If there is only one pair that keep its original position, it is called monotoortholog.

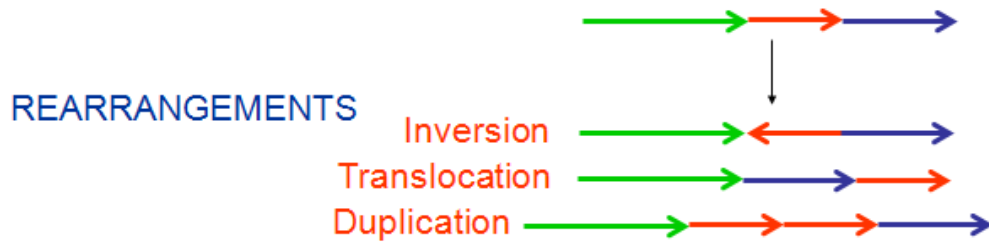


Figure 1: evolutions in DNA level

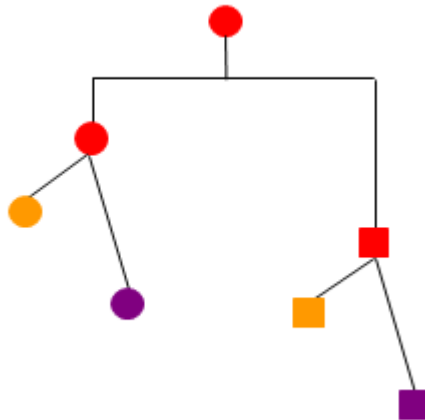


Figure 2: Duplication and speciation in the phylogenetic tree

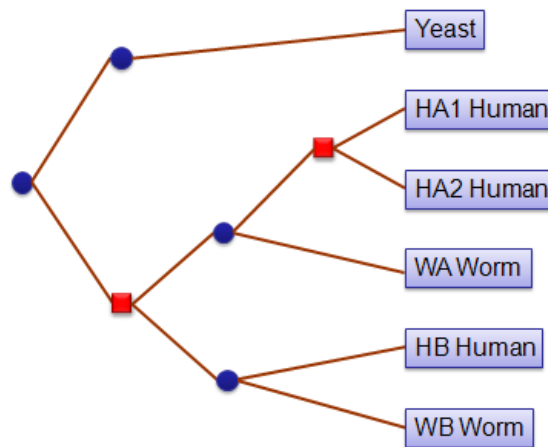


Figure 3: An example for phylogenetic tree

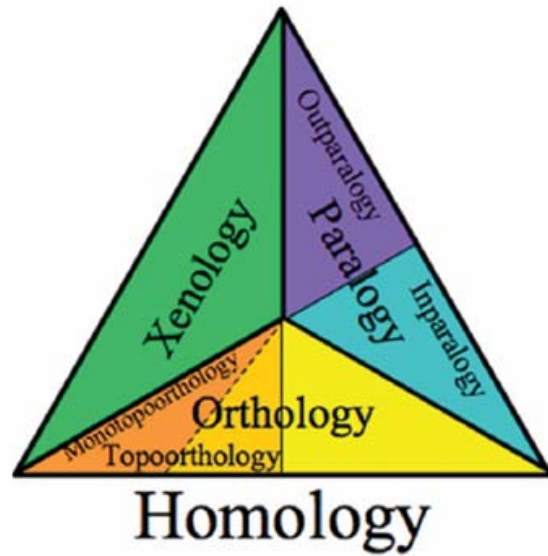


Figure 4 : Refinements of homology

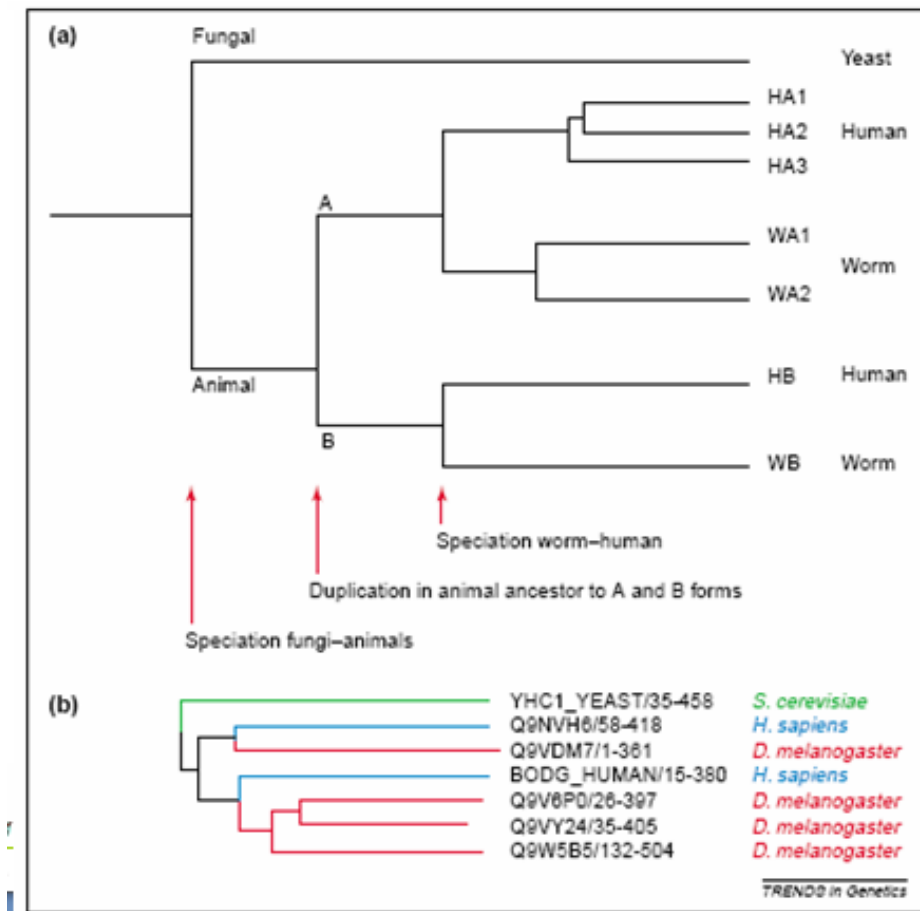


Figure 5: Different kinds of Orthology and Paralogy

➤ Multiple Sequence Alignment

Definition of multiple sequence alignment:

Given N sequences x_1, x_2, \dots, x_N ; insert gaps in each sequence, such that

- All sequences have the same length L
- Score of the global map is maximum

Multiple alignments uncover the elements that are conserved among a class of organisms, which are important in their common biology. These patterns of conservation can help us find out the function of the elements.

- Sum of Pairs

Definition of Induced Pairwise Alignment:

Induced pairwise alignment is pairwise alignment induced by the multiple alignment i.e. the induced pairwise alignment of two sequences is obtained by removing gaps inserted in the same column.

For example :

x: AC-GCGG-C
y: AC-GC-GAG
z: GCCGC-GAG

induces:

x: ACGCGG-C; x: AC-GCGG-C; y: AC-GCGAG
y: ACGC-GAC; z: GCCGC-GAG; z: GCCGCGAG

where we have removed the gaps in the same column when aligning x and y.

It is not easy to find a scoring function for multiple alignments which is computationally feasible and also satisfies the evolution of the sequences.

The common scoring function which is currently used is the sum-of-pairs.

For a given multiple alignment, we use pairwise scoring to find score of its all induced pairwise alignment and then we compute a weighted sum of them. In choosing the weights, if there are species of large population, we prefer to weight them less as they have more redundant information.

$$S(m) = \sum_{k < l} w_{kl} S(m^k, m^l)$$

- Profile Representation

In profile representation, we compress data and its size remains unchanged as we increase the number of sequences. Here, each column is represented by the percentage of every letter and gap. Also we can keep extra information such as number of gap openings, extensions which can be used for affine gap scoring function of induced pariswise alignments. (figure 6)

		-	A	G	G	C	T	A	T	C	A	C	C	T	G
	T	A	G	-	C	T	A	C	C	A	-	-	-	-	G
	C	A	G	-	C	T	A	C	C	A	-	-	-	-	G
	C	A	G	-	C	T	A	T	C	A	C	-	G	G	
	C	A	G	-	C	T	A	T	C	G	C	-	G	G	
A			1					1		.8					
C	.6				1			.4	1	.6	.2				
G			1	.2						.2			.4	1	
T	.2					1	.6							.2	
-	.2			.8						.4	.8	.4			

Figure 6: Profile representation for a multiple alignment

➤ Multidimensional Dynamic Programming

Here we extend pairwise alignment algorithm to multiple alignment.

Generalization of Needleman-Wunsh:

$$S(m) = \sum_i S(m_i) \quad (\text{sum over column scores})$$

Also we find the optimal alignment up to a given point (i_1, i_2, \dots, i_N) by maximizing over its neighbors' score.

$$F(i_1, i_2, \dots, i_N) = \max_{(\text{all neighbors of cube})} (F(nbr) + S(nbr))$$

An example for aligning 3 sequences is given in figure 7.

One drawback of this extended method is its computational complexity which is exponential in the number of sequences. Assuming we have N sequences of length L therefore the size of matrix is L^N and every cell have $2^N - 1$ neighbors so the running time is of order $O(L^N 2^N)$.

Furthermore if we want to use affine gape scoring function we would increase the running time up to $O(L^N 4^N)$. As a result this generalized algorithm is not employed in practice.

$$F(i,j,k) = \max \{ \begin{array}{l} F(i-1, j-1, k-1) + S(x_i, x_j, x_k), \\ F(i-1, j-1, k) + S(x_i, x_j, -), \\ F(i-1, j, k-1) + S(x_i, -, x_k), \\ F(i-1, j, k) + S(x_i, -, -), \\ F(i, j-1, k-1) + S(-, x_j, x_k), \\ F(i, j-1, k) + S(-, x_j, -), \\ F(i, j, k-1) + S(-, -, x_k) \end{array} \}$$

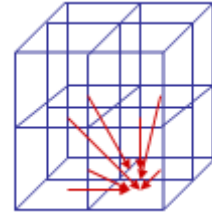


Figure 7 : An example for extended DP

➤ Progressive Alignment

A common technique to do multiple alignment with less time complexity is progressive alignment. If we know the evolutionary tree (e.g. figure 8) we can build the alignment hierarchically by using pairwise alignment at each internal node.

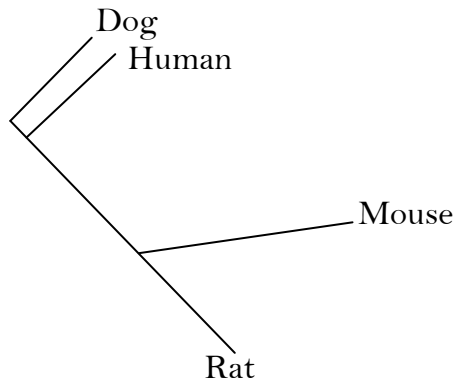


Figure 8: An example for evolutionary tree

The algorithm is as follows:

- Align closest first, in the order of the tree
- In each step, align two sequences x, y , or profiles p_x, p_y , to generate a new alignment with associated profile p_{result} .

Also we can have a model with edge weights which are proportional to the divergence in the corresponding edges. The resulting model is a weighted average of two previous profiles.

As an example, in figure 9, for the internal node of X and Y we do a pairwise alignment and summarize it to a profile representation. Then we align internal nodes using substitution scoring function.

Example

Profile: (A, C, G, T, -)

$$p_x = (0.8, 0.2, 0, 0, 0)$$

$$p_y = (0.6, 0, 0, 0, 0.4)$$

$$s(p_x, p_y) = 0.8*0.6*s(A, A) + 0.2*0.6*s(C, A) \\ + 0.8*0.4*s(A, -) + 0.2*0.4*s(C, -)$$

Result: $p_{xy} = (0.7, 0.1, 0, 0, 0.2)$

$$s(p_x, -) = 0.8*1.0*s(A, -) + 0.2*1.0*s(C, -)$$

Result: $p_{x-} = (0.4, 0.1, 0, 0, 0.5)$

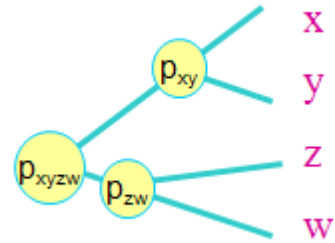


Figure 9: An example for progressive alignment

But when the structure of the evolutionary is unknown, we can change our algorithm as follows:

- Perform all pairwise alignments
- Define distance matrix D , where $D(x, y)$ is a measure of evolutionary distance based on pairwise alignment
- Construct a tree (UPGMA / Neighbor Joining / Other methods)
- Align on the tree

One drawback in this technique is that the initial alignments can be 'frozen' and do not alter despite new evidence coming in. Hence it won't correct an earlier bad decision.

For example:

x: GAAGTT

y: GAC-TT

z: GAACTG

w: GTACTG

If we first align X,Y and Z,Y and then align the results, we see that the optimal alignment is to swap C and gap in Y sequence. In order to solve such problems, we use heuristic algorithms such as iterative refinement, A*-based search, simulated annealing and consistency.

- Iterative refinement

In this algorithm we iteratively remove a sequence and realign it to the rest of the sequences. Also to make reduce the time complexity we allow the sequence to vary in an arbitrary band around the sequence.

Algorithm (Barton-Stenberg):

- For $j = 1$ to N , Remove x_j , and realign to $x_1 \dots x_{j-1} x_{j+1} \dots x_N$
- Repeat until convergence

For instance in figure 10, we have removed y and realign it with the projection of X, Z in an arbitrary variation band.

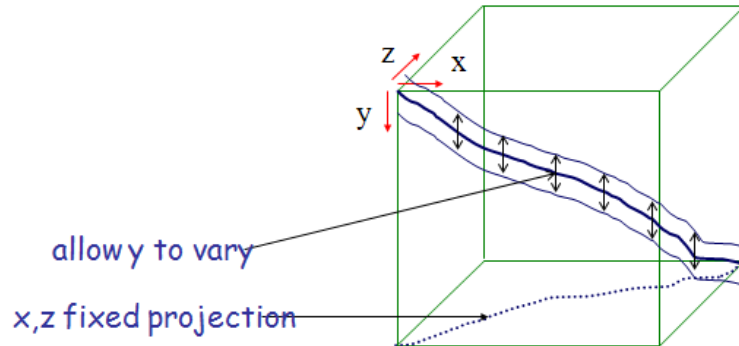


Figure 10 : Iterative refinement

For example, consider following alignment.

X: GAAGTTA
 Y: GAC - TTA
 Z: GAACTGA
 W: GTACTGA

If we realign y we can get a better alignment with 3 more matches.

X: GAAGTTA
 Y: G - ACTTA
 Z: GAACTGA
 W: GTACTGA

However there are some situations that iterative alignment cannot handle well.

For example, in the example below:

X: GAAGTTA
 Y1: GAC - TTA
 Y2: GAC - TTA
 Y3: GAC - TTA
 Z: GAACTGA
 W: GTACTGA

We should shift ACs' in all Y_i 's by one position to the right. But it cannot be done by removing and realigning one Y_i at a time.

To fix this problem there are many proposed techniques but all are some sort of local optimum search, for instance, instead of removing one sequence at a time we can iterate across branches of the tree.

➤ Consistency

The consistency technique enables us to find more accurate alignments by avoiding making bad choices in pairwise alignment between two sequences with the help of information from other multiple sequences.

Basic method for applying consistency:

- Compute all pairs of alignments xy, xz, yz, \dots
- When aligning x, y during progressive alignment
 - For each (x_i, y_j) , let $s(x_i, y_j) = \text{function_of}(x_i, y_j, a_{xz}, a_{yz})$
 - Align x and y with DP using the modified scoring function

For example in figure 11 we can align x_i to both y_j and $y_{j'}$, now if we check that in the other sequence Z , z_k aligns with x_i and y_j we would choose y_j .

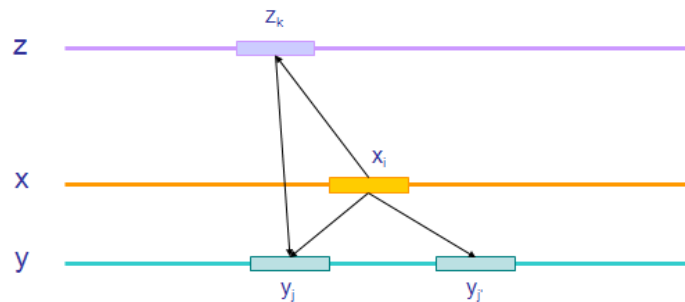


Figure 11: an example for consistency technique