Sequence alignment

Any pair of nucleic acid sequence will share a degree of similarity. For example DNA sequence are constructed from an alphabet of only four letters (A,T,G,C), so any sequence that consist of a mixture of these letters will show some similarity to any other similarly constructed sequence. We need a way of quantifying similarity.

Quantifying similarity begins with an **alignment,** such as that shown for the very short DNA sequence in Fig 1. In this figure, the two sequences are written one above and below each other are **aligned or equivalenced.** In fig 1., letters 7-9 of sequence 1 (TTG) are not aligned with any letters from sequence 2. When this happens, we say **gap** has been introduced. The point of introducing a gap here is that it enables a better alignment of the two sequences, in which more of the aligned pairs are identical letters. In this case, it enables the shared CGCAT directly after the gap to align between the two sequences.

SEQ1: AATTGATTGCGCATTTAAAGGG

SEQ2: AACTGA --- CGCATCTAAGGG

Fig 1. An alignment of two short DNA sequence

Alignment can be interpreted in evolutionary terms. When identical letters are aligned, the simplest interpretation is that these letters were part of the ancestral seqeuence and have remained unchanged. When non –identical letters are aligned, the simplest interpretation is that a mutation has occurred in one of the sequence.

**Gap** in alignments can be interpreted in terms of the **insertion or deletion** of letters in one of the sequences with respect to the ancestral sequence. In fig 1, gap could have resulted either from an insertion of three letters in sequence 1 or deletion of three in sequence 2. Again, without the ancestral sequence, these possibilities cannot distinguished, so the gap is sometimes referred to as an **indel.**

**Alignement algorithms**

Finding the best alignment between the two sequences in Fig 1 was fairly straightforward, because the sequences are short and very closely related. It is much more difficult if the sequences are longer or if they are less closely related. In these cases, computational methods are required to find best alignment of the sequences. Fortunately, there are known computational methods for this task, called **Dynamic Progmming Algorithms. These algorithms take two input sequences and produce as output the best alignment between them.**

**Sequence similarity analyses commonly use two dynamic programming algorithms, the Needleman-Wunsch Algorithm and Smith-Waterman algorithm.** These are closely related, but the main difference is that the **Needleman-Wunsch Algorithm finds global similarity between sequences, while Smith-Waterman algorithm finds local similarity finds the local similarity.**

 **An alignment from the Smith-Waterman algorithm might only cover a small (local) part of each sequence, while a Needleman –Wunsch alignment will try to cover a much of the sequences as possible, starting at left most end of one of the sequences and finishing at the right most end of one of the sequences.**