## *Gappiness value*

A truly divergent sequence, evolutionarily separate from the rest, can introduce large gaps in a multiple alignment. So that, we want to identify the sequences that potentially introduce gaps in the rest of the sequences. To look for these sequences, we firstly identify the columns that have many more gaps than amino acid/nucleotide positions. For each sequence, we account for the number of columns having amino acid/nucleotide when most of the sequences have a gap. By calculating the number of gaps that each sequence generate in the rest we could rank them by gap-generation capacity.

To evaluate the capacity of each sequence to introduce gaps in the alignment, a gappiness value (*gpp*) is calculated. The higher the *gpp* value the higher the number of gaps this sequence can introduce in the set of remaining sequences of the alignment. In consequence, as the *gpp* value increases the probability that the corresponding sequence is strongly divergent from the rest increases as well.

In a given sequence the *gpp* value is initialized at 0 and is incremented when a position is found to have amino acid/nucleotide and the majority of the remaining sequences have a gap. This increment is inverse to the number of sequences having amino acid/nucleotide in this position. For each sequence, a *gpp* value is calculated and then is compared with the rest. The sequence with the largest *gpp* value is considered as the sequence that introduces more gaps in the MSA. This value is calculated as follows:

- Given an alignment with *K* sequences and *L* positions.
- For each column in the alignment, a C*n* value which indicates the number of positions that are amino acid/nucleotide is defined (no gap).
- *f(s,n)* is defined as the number of no-gap positions with Cn=*n* , in sequence *s*.

Then

$$gpp_s = \frac{\sum_{n=1}^{k} (K - Cn) * f(s,n)}{L * K}$$

As an example, let us apply this formula to a simple case, as could be a MSA with 4 sequences and 5 columns.

| | **1** | **2** | **3** | **4** | **5** |
|---|---|---|---|---|---|
| **Seq 1** | A | A | T | - | T |
| **Seq 2** | A | - | T | - | A |
| **Seq 3** | - | - | C | T | T |
| **Seq 4** | - | - | G | C | - |

Firstly, the *Cn* values for each column are calculated: $Cn_1 = 2$, $Cn_2 = 1$, $Cn_3 = 4$, $Cn_4 = 2$, $Cn_5 = 3$,

After that, let's calculate *f(s,n)*

f(1,1)=1, f(2,1)=0, f(3,1)=0, f(4,1)=0

f(1,2)=1, f(2,2)=1, f(3,2)=1, f(4,2)=1

f(1,3)=1, f(2,3)=1, f(3,3)=1, f(4,3)=0

f(1,4)=1, f(2,4)=1, f(3,4)=1, f(4,4)=1

Finally, the *gpp* value for each sequence is calculated

$gpp_1$ = ((4-1)*1 + (4-2)*1 + (4-3)*1 + (4-4)*1) / 4*5 = 0.3

$gpp_2$ = ((4-1)*0 + (4-2)*1 + (4-3)*1 + (4-4)*1) / 4*5 = 0.15

$gpp_3$ = ((4-1)*0 + (4-2)*1 + (4-3)*1 + (4-4)*1) / 4*5 = 0.15

$gpp_4$ = ((4-1)*0 + (4-2)*1 + (4-3)*0 + (4-4)*1) / 4*5 = 0.1

The *gpp* values show that sequence 1 is the one introducing more gaps in the alignment. In this example, an *n* value was calculated for each column and all were used to calculate the *gpp* value. If we consider column 3, it does not seem logical to use it to calculate the *gpp* value because it does not include any gap. On the other hand, in column 5 there is only 1 gap so it is unlikely that 4 insertion events have occurred. The most likely explanation is that a deletion has occurred in sequence 4. To account for these cases, a gap threshold is defined in the algorithm. As a result, only those columns with a proportion of gaps larger than the threshold are considered when calculating *gpp* values. By default the threshold is set as 0.66 indicating that when at least 66% of the positions in a column are gaps this column would be considered for the *gpp* value calculation. However, the users could also define this threshold.

## *Evaluating the original alignment score*

For each alignment, the program calculates a score that will be a quality measure. To calculate this score we use the Sum-of-Pairs (SP) method. This is one of the most popular, simple and used methods used for scoring a MSA. The SP score is calculated as follows:

Given *N* sequences with length *L*, aligned forming a MSA matrix M = N x L.

- Given a scoring matrix which provides a score *s(x,y)* for the alignment of characters *x* and *y*.

Then, the *SP* score, for the column $m_i$ of the M matrix is given by

$$SP(m_i) = \sum_{k<l} s(m_i^k, m_i^l)$$

where $m_i^k$ is the *k*-th element from column *i* and $m_i^l$ is the *l*-th element from column *i*. The final *SP* score for the whole alignment in *M* results from adding all the scores calculated in the columns $SP(m_i)$.

$$SP(M) = \sum_i SP(m_i)$$

Let us apply this formula to a simple case, such as a MSA with 3 sequences and 5 columns. We define a simple scoring matrix with values +1 for coincidences, -1 for mismatches and -2 for gaps. When both positions have gaps, we will score 0 in order to avoid a double gap penalty. We have the *M* matrix:

| m1 | m2 | m3 | m4 | m5 | |
|---|---|---|---|---|---|
| T | G | C | - | G | |
| A | | G | C | T | G |
| A | | G | C | - | G |

The score for column 1 will be:

s(m1)= s(T,A) + s(T,A) + s(A,A) = -1+ -1 + 1 = - 1

We must calculate the score for each column. After that, we add together the scores from all columns in order to obtain the final *SP* score:

SP(M) = s(m1) + s(m2) + s(m3) + s(m4) + s(m5) = -1 + 3 + 3 - 4 + 3 **= 4**

The same system can be used for assessing amino acid alignments, just by providing an appropriate

scoring matrix.

Scoring matrices typically do not include gap penalty values. One of the main problems with the *SP* method is the gap-penalty assessment. During the alignment process, inserting a gap in a sequence is always penalized with a low score. This penalty depends on how evolutionarily close the sequences are and on whether it corresponds to a gap opening or a gap extension. In our case, we are going to assess all gaps with the same value, but this can be easily modified.

Sometimes, the alignment process generates large gaps due to errors or the use of inappropriate or very divergent sequences. These often lead to decreases in the quality of the alignment. We assume that, during the evolutionary process, losing or gaining a residue is less likely than an amino acid or nucleotide substitution. A gap in just one sequence means a lack homology (due to insertions or deletions) with the rest of the sequences and represents a loss of information for subsequent analyses. We want to minimize the gap effect, so that we penalize more sequences with higher number of gaps. We cannot avoid the double gap penalty, so we will score the occurrence of two gaps as the occurrence of a character and a gap. In consequence, in our analysis a gap will be penalized more than a mismatch. A value lower than the rest in the scoring matrix is assigned as a gap penalty. Using all the values contained in the scoring matrix we obtain:

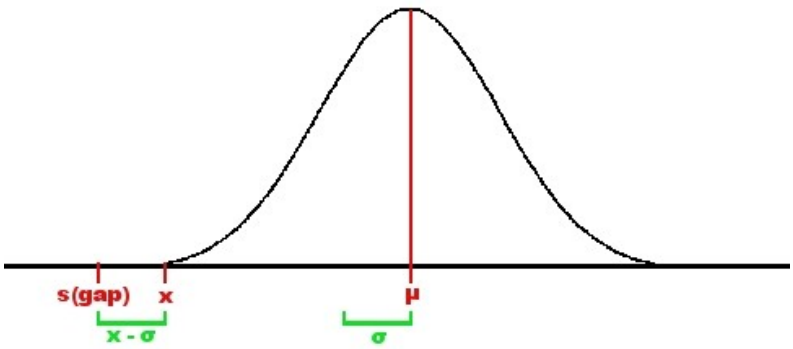


Figure 1. Gap penalty value

The gap penalty is defined as:

$$s(gap) = x - \sigma,$$

where $\sigma$ is the standard deviation and $x$ is the lowest value in the matrix.

If the alignment contains symbols which are not defined in the scoring matrix, such as *, ? or X, they will be assigned the *s(gap)* value.

## *Evaluating the weight of each sequence on the quality of the alignment.*

For each sequence in the MSA we evaluate the weight (or influence) that it has on the alignment quality. For this, each sequence is scored with a method derived from SP.

Given *N* sequences with length *L*, aligned forming a MSA matrix *M = N * L*.

Given a scoring matrix which provides a score *s(x,y)* for the alignment of characters *x* and *y*.

Then, the *w* score for *i*-th position in sequence *s* in the *M* matrix is calculated as

$$w(s_i) = \sum_{j=1,\, j \neq s}^{N} s\left(m_i^s, m_i^j\right)$$

where $m_i^s$ is the element from sequence *s* in position *i* and $m_i^j$ is the element from sequence *j* in position *i*. The *W* score for sequence *s* results from adding the *w* score for each position in the sequence

$$W(s) = \sum_{i=1}^{L} w(s_i)$$

The *W* score can be interpreted as a measure of the influence of each sequence on the MSA quality.