Bioinformatics

By Andrew Howell (tzjn72)

# Custom substitution-cost function

## Alphabet

My custom cost function revolves around the alphabet of letters found in DNA: [ A, C, G, T ]

All DNA sequences will be accepted by substitution-cost function and all sequences using this substitution-cost function must be comprised of a combination of the letters outlined in the alphabet.

## Substitution-Cost Function

My function works with pairs of letters and therefore uses a table of size n *x* n where n is the length of the alphabet squared plus dels (“-“s), which is 4^2 + 1 = 17.

The function involves an multiplier that produces either a positive or negative 1. It works like an exclusive OR (XOR) operation, whereby if 1 letter is shared in the pairs (position of the letters has no effect here), the XOR returns True and the cost is positive. However, if no letters are shared or both letters are shared, the XOR returns False and the final cost is negative as the cost is multiplied by this negative multiplier.

Furthermore, the cost of matching the pair is determined by cross matching the pair. For example, matching AC and GT will use the scores of matching A with T and C with G. These individual letter substitution costs are stored in a 5x5 matrix. This cost is either positive or negative, as determined by multiplying this cost by the multiplier explained previously.

Therefore, all costings are produced by an XOR operation on the pair of pair sequences, multiplied by the crossed matchings from the 5x5 matrix:

Cost = ( If XOR( [ u, v ], [ x, y ] ) is True: (1), Else: (-1) ) x ( F( u, y ) + F( v, x ) )

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| F(i,j) | A | C | G | T | - |
| A | 9 | -2 | 5 | 9 | 3 |
| C | 1 | 4 | -7 | 4 | -16 |
| G | 2 | 3 | 20 | 12 | -5 |
| T | -4 | -1 | -8 | 1 | -17 |
| - | -19 | -31 | -12 | -5 | -45 |