DNA Binary Hamming Distance

Here is a way to convert DNA strings into a binary representation that can still be used to find the hamming distance between the two strings. Assuming a 4 letter alphabet – A, C, G and T, then a 2 bit binary representation such as

A – 00,

C – 01,

G – 01,

T – 11.

would produce a hamming distance between each pair of strings of either 1 or 2, which is not useful for calculating the hamming distance between the strings.

A -----1-----C

| \ /|

| \ / |

1 /2 \ 1

| / \ |

| / \|

G -----1-----T

However, using a 3 bit representation of the letters, each letter can be given a binary value that is a hamming distance of 2 from every other letter.

A – 000,

C – 011,

G – 101,

T – 110.

A -----2-----C

| \ /|

| \ / |

2 /2 \ 2

| / \ |

| / \|

G -----2-----T

A**G**G**T** -> 000101101110

A**C**G**C** -> 000011101011

XOR 000110000101

Total 1s = 4

Divide by 2 for final hamming dist = 2.

Total number of 1s in output string is 4, divide by 2 because each letter differs from the others by 2 and the hamming distance between the strings is 2. This is correct as the strings differ at the 2nd and 4th positions only.

This may be useful because the binary representation of the DNA string will take less than half the space that the character representation would. The binary hamming distance comparison also offers the potential to use XOR to compare the strings, which may be a faster operation than character comparison.