# Hamming

Calculate the Hamming difference between two DNA strands.

A mutation is simply a mistake that occurs during the creation or copying of a nucleic acid, in particular DNA. Because nucleic acids are vital to cellular functions, mutations tend to cause a ripple effect throughout the cell. Although mutations are technically mistakes, a very rare mutation may equip the cell with a beneficial attribute. In fact, the macro effects of evolution are attributable by the accumulated result of beneficial microscopic mutations over many generations.

The simplest and most common type of nucleic acid mutation is a point mutation, which replaces one base with another at a single nucleotide.

By counting the number of differences between two homologous DNA strands taken from different genomes with a common ancestor, we get a measure of the minimum number of point mutations that could have occurred on the evolutionary path between the two strands.

This is called the 'Hamming distance'.

It is found by comparing two DNA strands and counting how many of the nucleotides are different from their equivalent in the other string.

GAGCCTACTAACGGGAT
CATCGTAATGACGGCCT
^ ^ ^ ^ ^ ^
The Hamming distance between these two DNA strands is 7.

Implementation notes
The Hamming distance is only defined for sequences of equal length. This means that based on the definition, each language could deal with getting sequences of equal length differently.

# LES TESTS

`hamming.spec.js`
```js
var Hamming = require('./hamming');

describe('Hamming', function () {
```

```
var hamming = new Hamming();

it('no difference between identical strands', function () {
expect(hamming.compute('A', 'A')).toEqual(0);
});

xit('complete hamming distance for single nucleotide strand', function () {
expect(hamming.compute('A', 'G')).toEqual(1);
});

xit('complete hamming distance for small strand', function () {
expect(hamming.compute('AG', 'CT')).toEqual(2);
});

xit('small hamming distance', function () {
expect(hamming.compute('AT', 'CT')).toEqual(1);
});

xit('small hamming distance in longer strand', function () {
expect(hamming.compute('GGACG', 'GGTCG')).toEqual(1);
});

xit('large hamming distance', function () {
expect(hamming.compute('GATACA', 'GCATAA')).toEqual(4);
});

xit('hamming distance in very long strand', function () {
expect(hamming.compute('GGACGGATTCTG', 'AGGACGGATTCT')).toEqual(9);
});

xit('throws error when strands are not equal length', function () {
expect(function () { hamming.compute('GGACGGATTCTG', 'AGGAC'); }).toThrow(
new Error('DNA strands must be of equal length.')
);
});
});
```