# Fantasy protein mutation

In our fantasy world, all proteins are made of five possible amino acids (a, b, c, d and e) in sequences of eight. For example, one protein could be made from a sequence such as badbaedc and another could have the sequence adeabbce. Quite reasonably, mutations happen one amino acid at a time (point mutations). So, the first sequence badbaedc could mutate to badbaedd or perhaps aadbaedc. Let's now imagine that the environment is demanding that the first protein should mutate into the second protein by a series of random single amino acid point mutations. How could this be possible? It wouldn't be. Single mutations would likely lead to an inviable amino acid sequence. Occasionally perhaps this single mutation might hit on the viable sequence of a different protein but this protein would likely be unsuitable in the place where it finds itself in the organism and it would be just as useless as a totally inviable sequence. There is no possible way in which our first sequence could mutate into the second sequence in this manner; the intermediaries would be inviable.

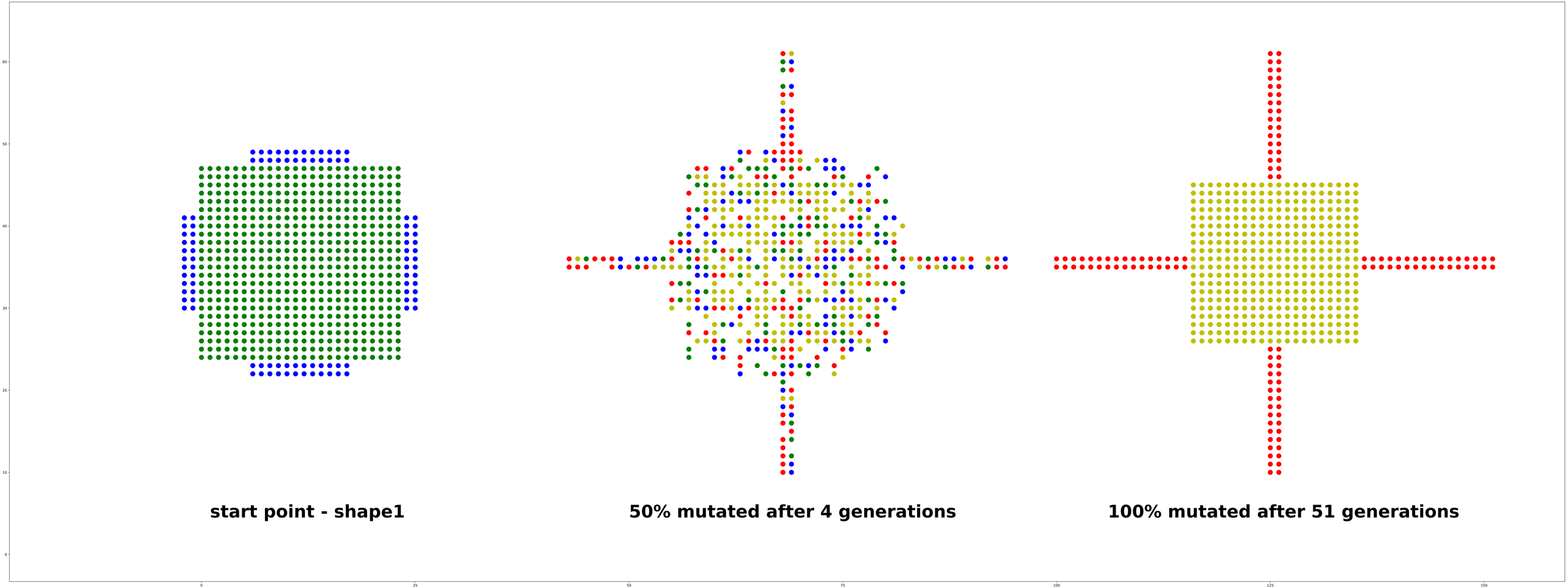
There is however a possible system which could facilitate our protein evolution; a system which allows one protein to morph, when required, into another. That system is based on 'many-to-one' mapping or the use of redundancy in the coding system. The new system is based on all possible eight letter codes permutated from the five amino acids (i.e. 390,625 different codes) and our imagined world consists of four proteins represented graphically by four different colours with a fifth colour white used to facilitate the graphical representation. Each of the four proteins and the white colour can be made by approximately one fifth of the possible 390,625 sequences. In this system it is now easy to navigate between proteins by point mutations. There will be many sequences for say the blue protein which will be just one mutation away from the sequences of any other colour. Because the 390,625 sequences represent every possible code then all point mutations will hit on a viable code and there is now a one in five chance of hitting a correct code.

The computerised system starts with two different shapes each plotted by 768 points (or cells) on a graph.

The rules of our imaginary system are as follows. The second shape, shape2, which is composed of only the red and the yellow proteins is chosen as the target towards which the first shape, shape1, has to morph. Shape1 is composed of 768 blue and green cells. During each generation of the system every cell in shape1 has a chance to mutate and when the chosen cell mutates to the protein which is at the equivalent position in shape2, in other words it finds its target, then the cell is "locked-in" and can't then be mutated.

After just 4 generations, each consisting of 768 mutations, the target shape is 50% achieved but the search now becomes increasingly difficult as the final cells waiting to be mutated are more difficult to find (because there are less of them remaining unmutated) and it is only after 51 generations that the target is finally achieved.

Does evolution take advantage of many-to-one mapping and redundancy like this? Probably, yes.


0
10
20
30
40
50
60
0
25
50
75
100
125
150
start point - shape1
50% mutated after 4 generations
100% mutated after 51 generations