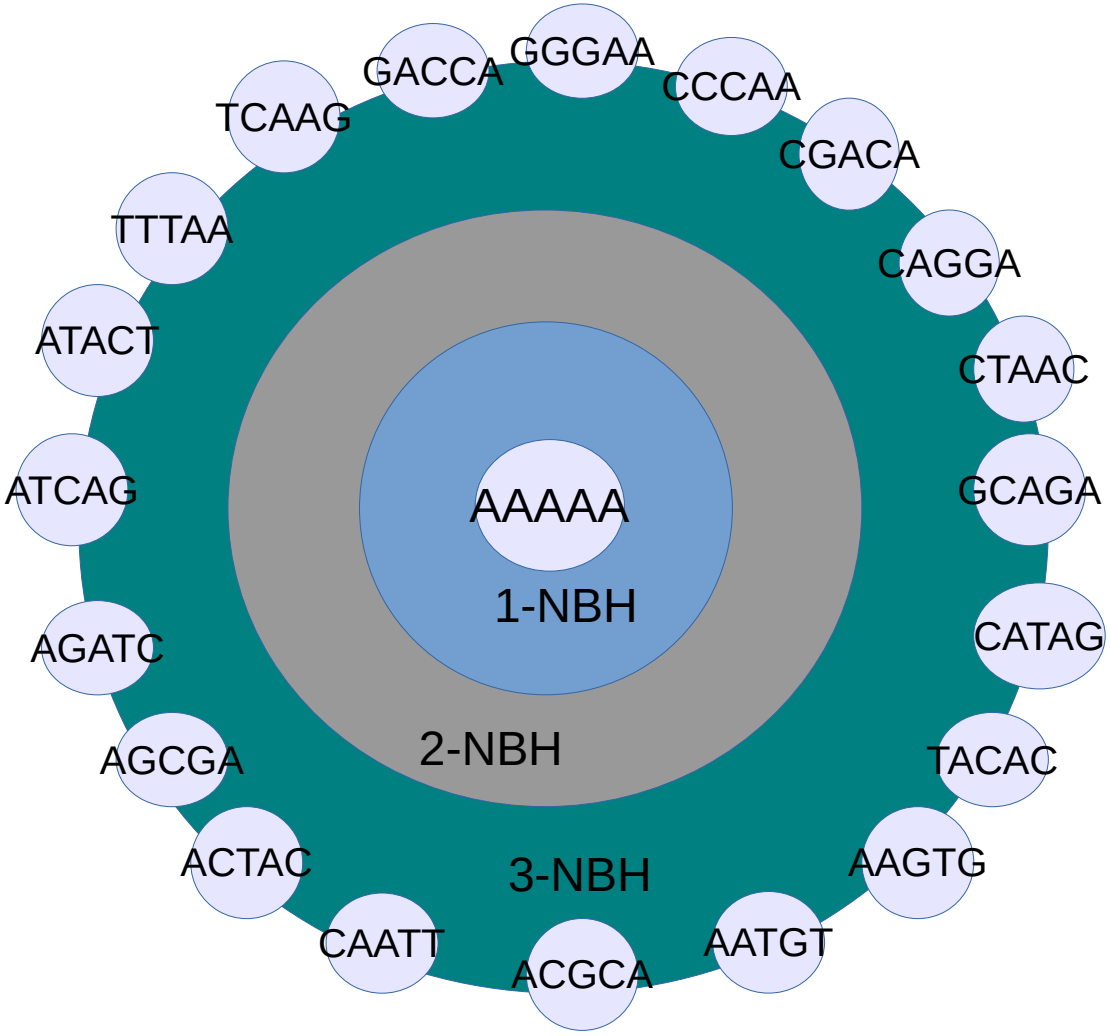


Visualization of a graph node labeled ‘AAAAA’ and its neighbourhoods.
1-NBH, 2-NBH and 3-NBH denote the neighbourhoods containing nodes differing in 1,2 or 3 positions.

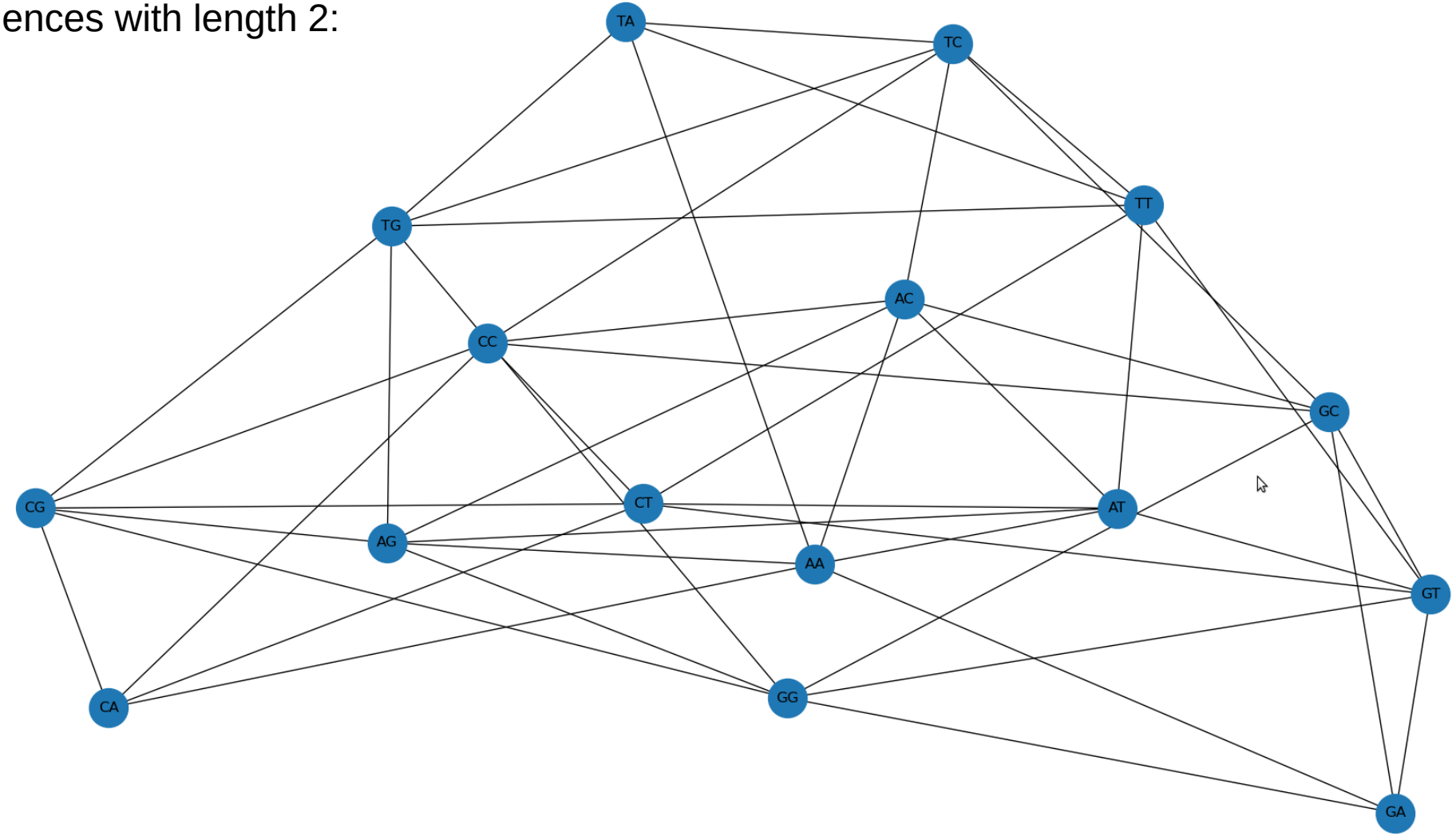
By construction, nodes in the outer perimeter may only differ in less than 3 positions due to the possibility of point mutations producing A’s again after 3 point mutations.

These unwanted sequences are eliminated by applying the brute force check to the result list containing the center node and all nodes found in the outer neighborhood.
The remaining nodes in the 1-NBH, 2-NBH and 3-NBH are then all marked as visited to prevent them from reappearing in later iterations of the algorithm.

To get the rest of the remaining valid nodes within the graph, the described neighborhood search followed by the elimination of invalid nodes is repeated for all valid nodes found previously until all or the desired number of nodes/sequences has been found.

E.g. in our example, the graph search is continued with ‘TCAAG’ and all the other nodes located within the outer perimeter. New valid nodes are always added to the list of valid nodes and all valid nodes are checked as described until no new valid nodes are found or the desired number of nodes has been found.

Example graph for
sequences with length 2:



Example graph for
sequences with length 3: