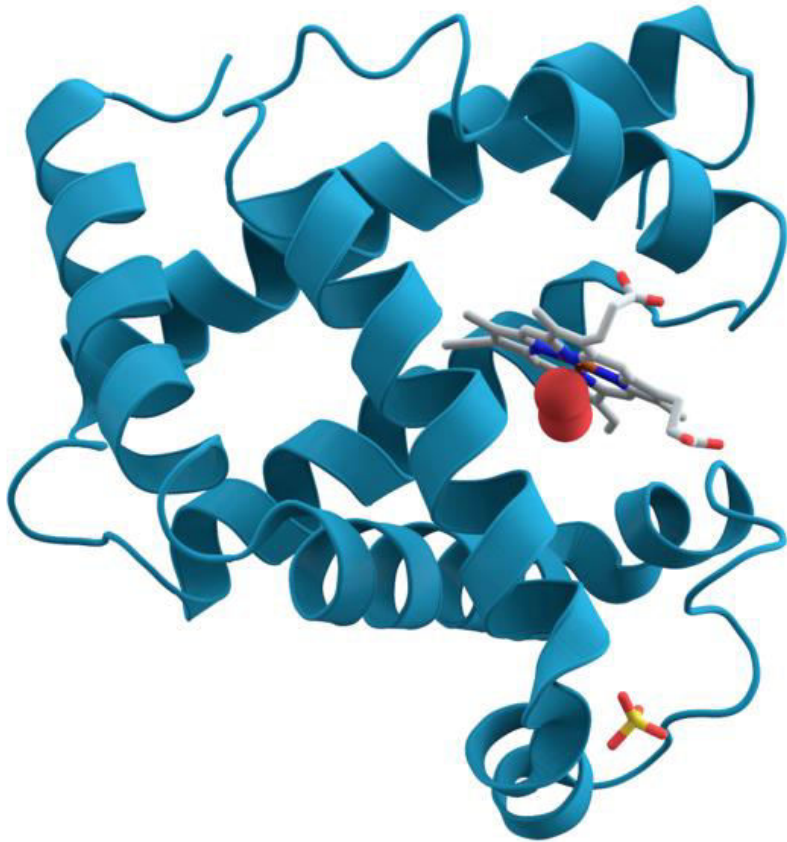
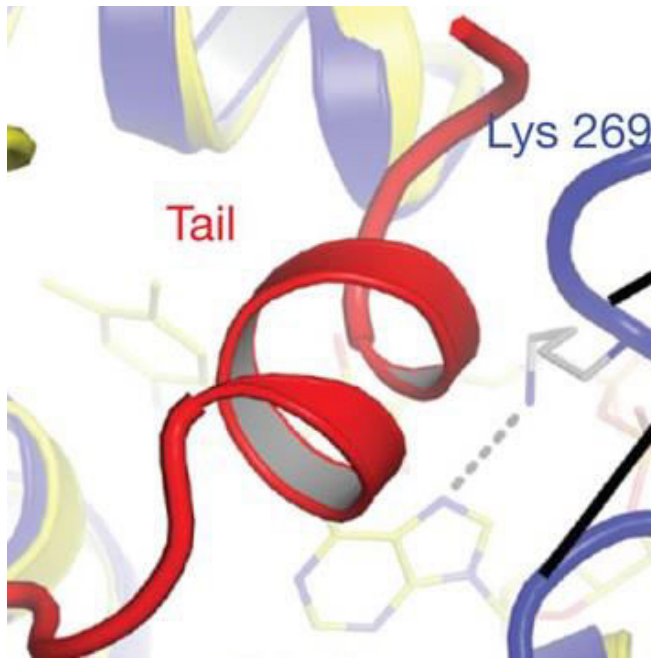


Structure of a protein



Structure of a motif



What is the specific problem, why did we choose this and can this be extended to graphs?

- To implement a divide and conquer algorithm to find all the copies of a tree (Q) in a graph (G), by conquering the minimum dominating set.
- We chose this project because the structural super imposition of the whole tree in the graph is more tedious when compared to the proposed method which partitions the given tree into dominions and then try to find the copies of each of these dominions in the given graph and merging them in order to find all the copies of the tree.
- This can be extended to sparse query graphs that can be reduced to trees by deleting a small number of edges.

Explain relationship between SSEs and current project?

Proteins can be represented by a primary structure which is a sequence of secondary structure with secondary structure elements (SSEs) made of polypeptide segments. A protein can be modeled as a graph in which each node represent a SSE and edges between nodes model interactions where the handed-nesses constitute supplemental information. The degree of a protein is at most the maximum number of interactions an SSE can have, which is a constant.

Complexities for brute force versus our method (compare, why does it work, what restrictions are needed?)

Brute force:

Complexity: $O(m \cdot n^m)$.

Proposed method:

The restrictions are

- ✓ The query graph is always a tree.
- ✓ The degree of given graph G is bounded to a constant maximum degree.

Complexity: $O(m \cdot n^{m/2})$.