Feature directions:

Hierarchical clustering is built to find small groups and then join those groups with other groups. It is really not meant to find groups where all members are roughly all related. Instead, a network graph would be an excellent way to accomplish this goal. Compare every loop to every other loop, and establish an edge between the two if they are similar enough. Then, find the busiest locations on the graph and establish those as your final models.

The model comparison score is relatively naïve. It relies on magic numbers and uses an equal weighting of phi, theta, anchor distance, and RMSD. With further experimentation, it might be the case that there are better cutoffs for , , and anchor distance. It would probably also be a good idea to try different weights for average RMSD, , , and anchor distance, choosing the weighting scheme which leads to the best results.

There might be a better way to obtain the axes for transforming into a protein. Because we have to give up one of our input axes in order to ensure all 3 axes are orthogonal, we lose some information about the SSE direction in exchange for maintaining information about overall loop direction. Perhaps there are some two unique vectors on a protein that are always perpendicular, and those two vectors could be used to generate the axes for transformation.

With improved scoring accuracy, it might be feasible to tackle longer loops. Longer loops have more configurations and are hence more difficult to predict, so with an improved algorithm, it would be feasible to expand the loop size range.

It is feasible that two similarly structured proteins with entirely different sequences arise from convergent evolution. Hence, it might be possible to explore further how not only the raw sequence is important, but the amino acids that make up that sequence, and how they might interact with each other.

Sources of error:

Magic-ish numbers

Weighting