02-601: Project Progress Report

Siddharth Reed, slreed

November 1, 2020

Finished As of now I have a working version of the core genetic algorithm, so it will create a random pool of initial sequences, pick the fittest ones and breed new members and continue this process until a fitness plateau is reached or the max number of iterations is reached. It will also write the last generation of sequences to a fasta file . I have also collected and cleaned a set of training data for the machine learning algorithm to train a dnazyme classifier in python. I have also made sure to document everything written thus far in comments and a read me.

ToDo

- pick algorithm for dnazyme classifier
- train dnazyme classifier and evaluate performance
- write wrapper script to get classification score for a sequence from a trained model, callable from go for the fitness function
- finish fitness scoring function in the genetic algorithm
- decide how to weight hairpins, target complementarity and classification score when calculating fitness
- finish the README

Possible extensions after this include showcasing how this preforms against regular in vitro SELEX. This paper evolved a set of DNAzymes in vitro with some biological purpose targeting a cRNA. So if can evolve a similar set of sequences in silico that proves the effectiveness of the tool. More thought needs to be made about the specifics of this type of comparison if I get there though. It could also be interesting to evolve a population of sequences and see what trends exist in terms of features of that DNA (length,GC content, etc.) as compared to the random starting pool.

Problems So far everything has gone well in building the genetic algorithm but I am still deciding on how to compute the fitness of a sequence. Currently I am considering using the presence of hairpins in the sequence, complementarity of the sequence to the target (alignment score likely) and the dnazyme model classification score. I am sure there will also be some issues with training a somewhat effective model like how to pick an algorithm or what parameters to use etc. The model does not need to be perfect but it should at least be better than chance.