Daniel Wong

BMI203 HW#3

Link to github: https://github.com/wongdaniel8/BMI203-HW3

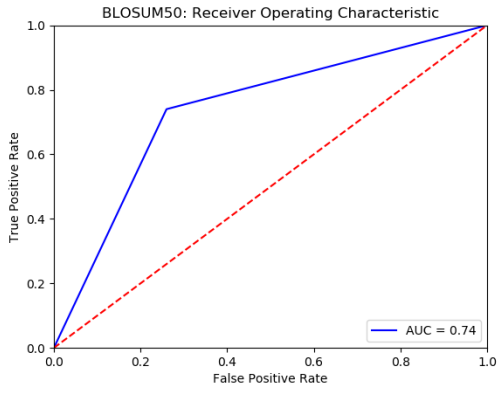
Part 1:

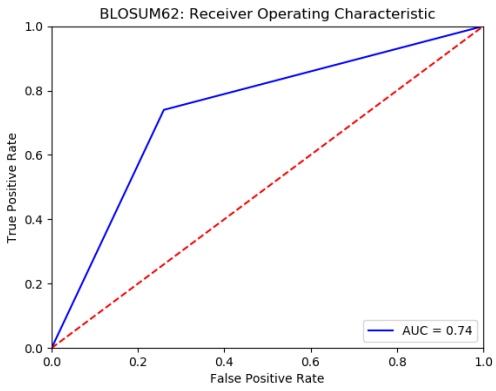
I implemented the Smith Waterman algorithm and checked for its validity by manually checking a few of the sequences in posPairs.txt and negPairs.txt. I also averaged the score across all positive pairs and compared this to the average score across all negative pairs. The positive pairs average was significantly higher and was a good sanity check just to see if the algorithm was working decently (see method getAverageScores(matrix, gap, extension_penalty)).

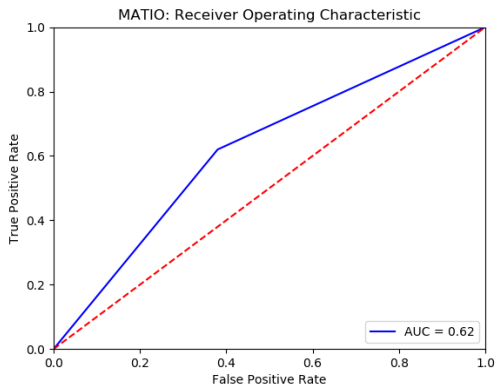
1) the best gap penalty and extension penalty combination that I found was:
    gap penalty: -7
    extension penalty: -3
This resulted in a false positive rate of .22 at a true positive rate of .70.

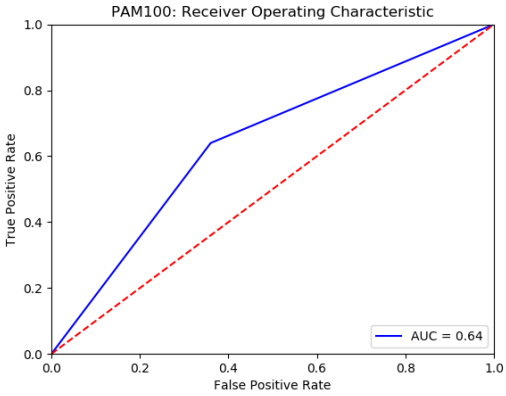
2) The best matrix at a true positive rate of .7, is the BLOSUM50 matrix, which yields a false positive rate of .22 (the next best was BLOSUM62 which gave false positive rate of .26)

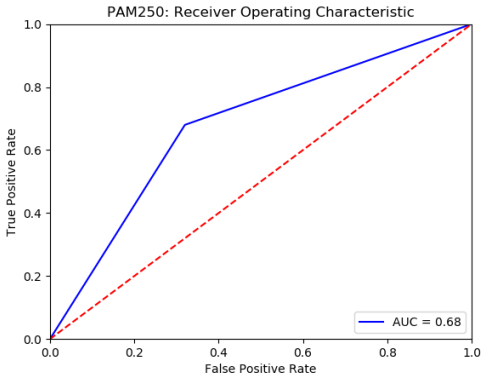
MATRIX	*FALSE POSITIVE RATE (at True Positive = .70)*
BLOSUM50	.22
BLOSUM62	.26
PAM250	.38
MATIO	.40
PAM100	.60







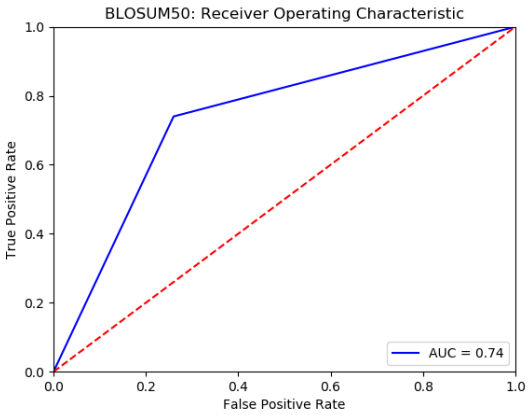




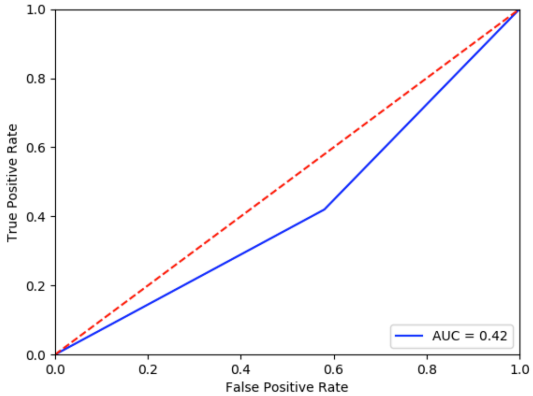
3) The performance of the Smith Waterman decreases when normalized by dividing the score by the shortest sequence in the pair. This unfairly penalizes a pair that could match very well but one sequence in the pair is very short. This can lead to great matches receiving low scores, and hence will increase the false positive rate.

We can see the false positive rate increase dramtically from the ROC curve generated from the BLOSUM50 with the new normalization schema:

Original Matrix



Normalized Scoring Matrix

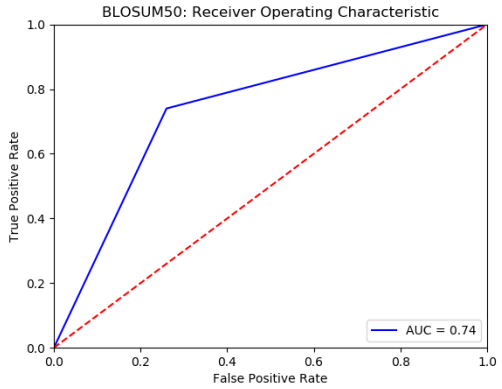


Part 2:

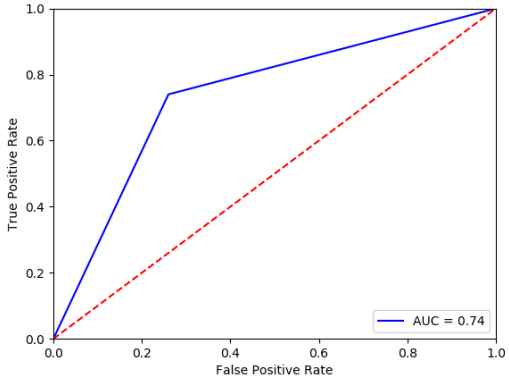
1) I implemented a genetic algorithm for optimization, with the same optimal penalties I found from part 1. See question 4 for description of algorithm.

2) I see pretty little improvement of fitness when running my optimization algorithm on my best performing BLOSUM50 matrix. This can be due to the fact that this matrix is already heavily optimized, and performs well already. The false positive rate at true positive = 70% declines slightly, with a total of one less false positive pair (a pair specified in the negatively correlated sequences file negPairs.txt) being above the threshold.

Original Non-optimized score = 3.479:



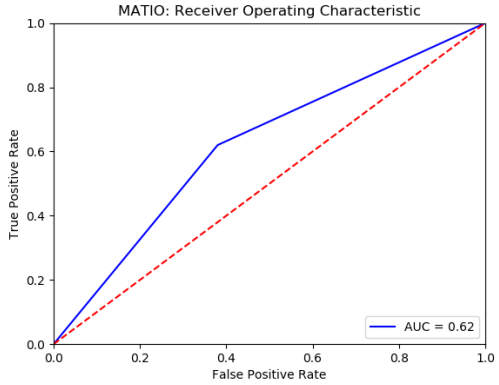
Optimized score = 3.512:



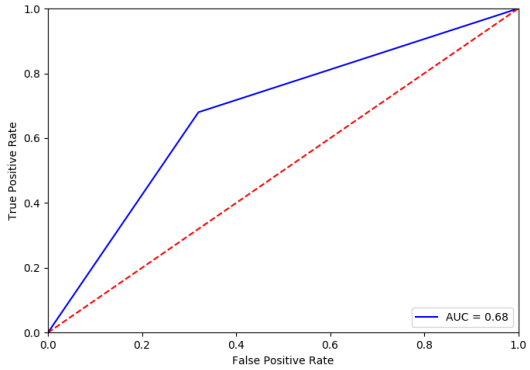
When realigning and rescoring the sequences using my optimized matrix, I see slightly better scores to the ones using the BLOSUM50 matrix. This is expected as the optimization function score didn't change that much. The average score of the positive pairs alignment was about 40 points higher than the average score of the negative pair alignments. This was about the same for the non-optimized matrix and the difference was small.

3) When running the optimization on the MATIO matrix, I see better improvement of fitness compared to the improvement of fitness of the BLOSUM50 matrix. This can be due to the fact that the MATIO matrix is not very fit to begin with, so improvement was not that difficult. The false positive rate dipped from 40% to 30% after 35 iterations.

Original



Optimized (k=30 iterations)



With only 35 iterations, we can see that the area under the curve score has improved by a decent margin. Given enough iterations for a long time, it should approach the same score as the optimized BLOSUM50 if the matrix mutates to become nearly identical to the BLOSUM matrix. This would take a long time since the mutations are random. With only 35 iterations of introducing mutations, this took about 20 minutes to run so I cut it. The improvement shows that this algorithm works decently well even with a limited number of iterations.

When I realign the sequences, the optimized matrix gives a margin of 220 points difference between the average of aligning the positive pairs and the average of aligning the negative pairs. The non-optimized MATIO matrix gives a margin of only 200 points. Also the average positive pairs alignment for the optimized is greater than the average positive pairs alignment for the non-optimized by 63 points.

4) For my genetic algorithm, I introduce random mutations iteratively, changing some of the values in the scoring matrix while maintaining symmetry. Like evolution, my mutations are random (using the random library in python and choosing random elements of the scoring matrix to add a random offset within a given range). If the new subsequent individual performs better than the previous, I keep these parameters in a greedy optimization scheme. The breeding part of the algorithm combines parts of the known fit BLOSUM50 matrix into the matrix being optimized, and this appendage is evaluated for improved fitness. I do this for a fixed number of iterations (n=35). I can improve the results of the algorithm by increasing the amount of time this algorithm is performed, and also increasing the range of the offset I choose to add or subtract from the individual score. I choose a relatively small range for the offset because I wanted to gradually search for the local optimum. Choosing large offsets would be equivalent to investigating broader areas of my optimization landscape (like increasing the step size in a gradient descent). This would be ideal if I had more runtime and more iterations so

that I don't get stuck in a local optimum. Theoretically, if I let the algorithm run for a very long time, I should eventually evolve MATIO to perform on par with the BLOSUM50 matrix.

5) To make a convincing case for the general usage of my scoring matrix, it would have to yield higher scores for pairs of sequences that have known similarity (like sequences for similar proteins that have known evolutionary origin). This will have to hold true across many pairs of positively associated sequences (far more than the ones provided in posPairs.txt). This matrix will also have to yield significantly lower scores for negatively associated sequences. A good way to test this is by generating random sequences and trying to align them—the score should be relatively low compared to aligning positively associated sequences or sequences known to be similar from evolution.

Optimized Matrices:
Same row column layout as given matrices

BLOSUM50 optimized:

```
A  R  N  D  C  Q  E  G  H  I  L  K  M  F  P  S  T  W  Y  V  B  Z  X  *
5 -2 -3 -2 -4 -1 -1  0 -2 -1 -2 -1 -1 -3 -1 -2  1 -3 -1  0 -2 -1 -1 -2
-2  7 -2 -2 -4  1  0 -3  0 -4 -3  2 -2 -3 -3 -2 -3 -3 -1 -3 -1  0 -1 -5
-3 -2  7  2 -2  0  0 -3  1 -3 -4  0 -2 -4 -2 -2 -2 -4 -2  1  4  0 -1 -5
-2 -2  2  8 -4  0  0 -1  2 -4 -4 -2 -6 -5 -1  0 -1 -5 -3 -4  5  1 -2 -2
-4 -4 -2 -4 13 -5 -4 -3 -3 -2  0 -3 -2 -2 -4 -1 -1 -6 -6 -4 -3 -2 -2 -5
-1  1  0  0 -5  7  2 -2  0 -3 -2  2  2 -4 -1  0 -1 -1 -1 -3  0  4 -1 -7
-1  0  0  0 -4  2  6 -3  0 -4 -3  0 -2 -3 -3 -3 -1 -3 -2 -3  4  8  0 -5
 0 -3 -3 -1 -3 -2 -3  8 -2 -4 -4 -2 -3 -4 -2  0 -2 -9 -3 -4 -1 -2 -2 -5
-2  0  1  2 -3  0  0 -2 11 -4 -3  0 -1 -1 -2 -1 -2 -3  2 -2  0 -1 -1 -5
-1 -4 -3 -4 -2 -3 -4 -4 -4  5  2 -3  2  0 -3 -1 -1 -3 -1  4 -4 -3 -1 -5
-2 -3 -4 -4  0 -2 -3 -4 -3  2  5 -3  3 -2 -9 -3 -1 -5 -1  1 -4 -3 -1 -3
-1  2  0 -2 -3  2  0 -2  0 -3 -3  6 -2 -4 -1  0 -4 -3 -5 -3  0  1 -1 -5
-1 -2 -2 -6 -2  2 -2 -3 -1  2  3 -2  7  0 -3  0  0 -1  0  1 -3 -1 -1 -5
-3 -3 -4 -5 -2 -4 -3 -4 -1  0 -2 -4  0  8 -4 -4 -2  1  4 -1 -4 -4 -4 -5
-1 -3 -2 -1 -4 -1 -3 -2 -2 -3 -9 -1 -3 -4 10 -1 -1 -4 -3 -3 -2 -1 -2 -5
-2 -2 -2  0 -1  0 -3  0 -1 -1 -3  0  0 -4 -1  5  2 -4 -5 -2  0 -3 -1 -5
 1 -3 -2 -1 -1 -1 -1 -2 -2 -1 -1 -4  0 -2 -1  2  5 -6 -2 -1  0 -1  0 -5
-3 -3 -4 -5 -6 -1 -3 -9 -3 -3 -5 -3 -1  1 -4 -4 -6 13  2 -3 -5 -2 -3 -5
-1 -1 -2 -3 -6 -1 -2 -3  2 -1 -1 -5  0  4 -3 -5 -2  2  8  1 -3 -2 -1 -5
 0 -3  1 -4 -4 -3 -3 -4 -2  4  1 -3  1 -1 -3 -2 -1 -3  1  5 -6 -3 -1 -7
-2 -1  4  5 -3  0  4 -1  0 -4 -4  0 -3 -4 -2  0  0 -5 -3 -6  5  2 -1 -5
-1  0  0  1 -2  4  8 -2 -1 -3 -3  1 -1 -4 -1 -3 -1 -2 -2 -3  2  5 -1 -5
-1 -1 -1 -2 -2 -1  0 -2 -1 -1 -1 -1 -1 -4 -2 -1  0 -3 -1 -1 -1 -1 -1 -5
-2 -5 -5 -2 -5 -7 -5 -5 -5 -5 -3 -5 -5 -5 -5 -5 -5 -5 -5 -7 -5 -5 -5  1
```

MATIO optimized:

A R N D C Q E G H I L K M F P S T W Y V B Z X *

5 1 1 -6 -2 8 -9 4 -6 -1 1 4 1 2 0 -8 7 7 4 -8 -1 8 -1 9

1 7 1 -4 -5 6 2 3 1 0 -7 -4 -8 8 -7 2 -4 -4 2 0 0 7 -7 8

1 1 7 1 5 2 4 -3 -5 -6 8 9 4 7 8 1 -5 -7 4 1 3 1 6 0

-6 -4 1 8 0 0 0 -8 -12 5 -9 3 6 7 10 -9 2 0 6 5 -2 6 4 0

-2 -5 5 0 13 4 3 0 -2 2 -2 -6 0 -1 -7 3 -4 5 -3 -1 11 1 3 9

8 6 2 0 4 7 -4 4 5 3 -3 4 6 -4 -9 1 5 1 -7 4 2 4 0 -7

-9 2 4 0 3 -4 6 5 9 0 -1 -4 3 5 3 6 -9 6 3 -1 -11 5 3 -8

4 3 -3 -8 0 4 5 8 1 -7 -8 2 -4 9 3 -4 4 -11 5 -2 8 -5 9 -7

-6 1 -5 -12 -2 5 9 1 13 5 -3 -8 -4 1 8 0 1 -7 -1 -1 1 1 1 -4

-1 0 -6 5 2 3 0 -7 5 5 8 1 5 -1 7 -7 -7 -7 -7 2 2 -7 -5 -3 -11

1 -7 8 -9 -2 -3 -1 -8 -3 8 5 -7 4 -1 -7 2 2 6 4 0 -9 -4 -4 -6

4 -4 9 3 -6 4 -4 2 -8 1 -7 6 -9 1 -2 1 6 8 -4 7 9 -5 6 9

1 -8 4 6 0 6 3 -4 -4 5 4 -9 7 -1 0 8 2 -7 2 -2 -8 -7 -1 -3

2 8 7 7 -1 -4 5 9 1 -1 -1 1 -1 8 3 4 -1 9 -3 8 -4 7 1 -3

0 -7 8 10 -7 -9 3 3 8 7 -7 -2 0 3 10 1 3 0 4 7 6 -1 0 -1

-8 2 1 -9 3 1 6 -4 0 -7 2 1 8 4 1 5 -7 8 -7 4 6 4 -5 8

7 -4 -5 2 -4 5 -9 4 1 -7 2 6 2 -1 3 -7 5 0 2 -7 -7 -8 -3 -1

7 -4 -7 0 5 1 6 -11 -7 -7 6 8 -7 9 0 8 0 15 -1 -1 -5 4 -7 7

4 2 4 6 -3 -7 3 5 -1 2 4 -4 2 -3 4 -7 2 -1 8 2 -8 -1 6 3

-8 0 1 5 -1 4 -1 -2 -1 2 0 7 -2 8 7 4 -7 -1 2 5 9 6 5 -8

-1 0 3 -2 11 2 -11 8 1 -7 -9 9 -8 -4 6 6 -7 -5 -8 9 5 -8 5 -7

8 7 1 6 1 4 5 -5 1 -5 -4 -5 -7 7 -1 4 -8 4 -1 6 -8 5 3 -8

-1 -7 6 4 3 0 3 9 1 -3 -4 6 -1 1 0 -5 -3 -7 6 5 5 3 -1 -4

9 8 0 0 9 -7 -8 -7 -4 -11 -6 9 -3 -3 -1 8 -1 7 3 -8 -7 -8 -4 1