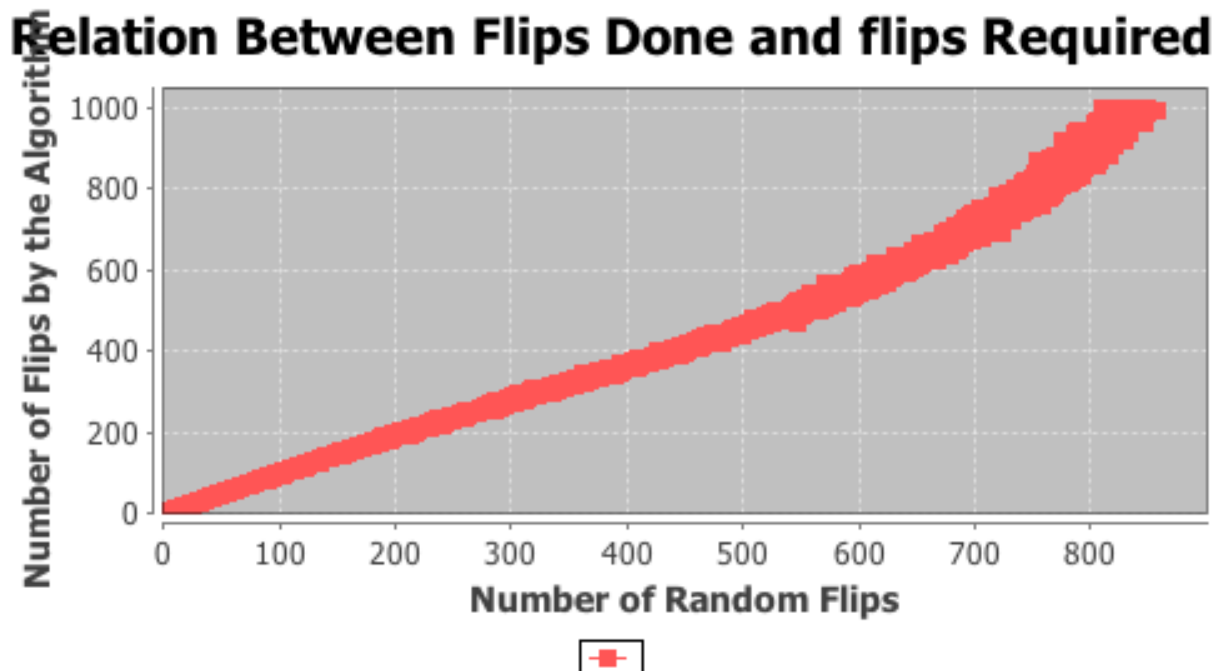


Genome rearrangements

Often times biologists would like to understand the connections and commonalities between DNA, this is where a genome rearrangement algorithm helps. With the slower method being discarded having that method greedily work towards a solution using a heuristic denoted by an ever increasing correct value, which fails to give optimal solutions for very simply problems. Then a more efficient algorithm was designed which uses pivots as a heuristic. This tries to decrease the number of pivots which turns out to be the key to efficiency.

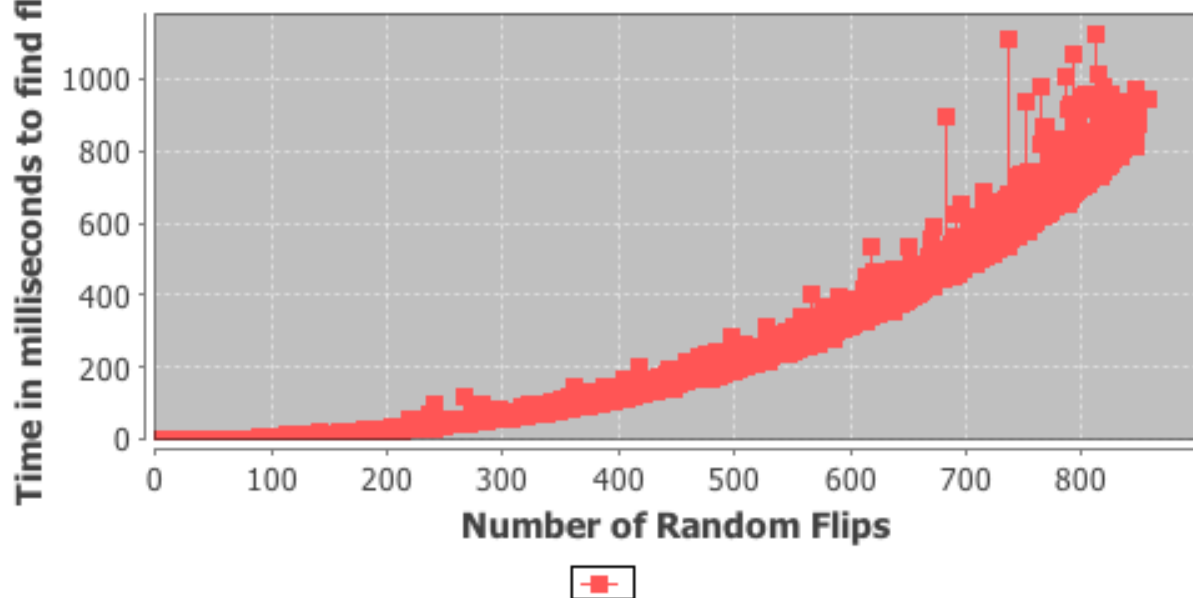
ImprovedBreakpointReversalSort is the key class in the program, this is the class which holds the algorithm to generate the number of random permutations as well as the solution with timing provided and writes it to a file in the resource directory.

BreakPointAnalysis class is what then reads the file and creates two images.

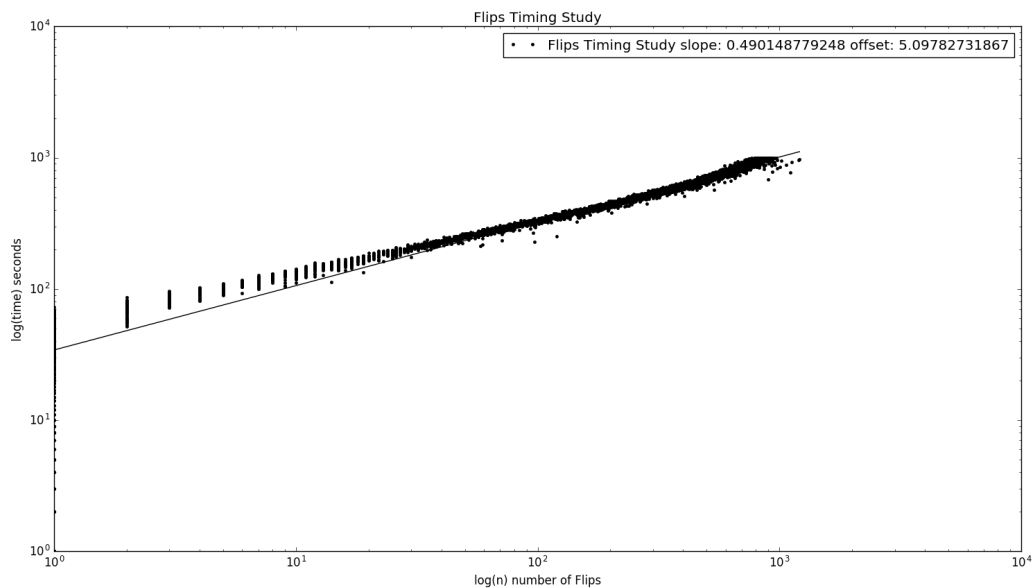


This image denotes the number of random flips vs the number of flips required to get the solution by the algorithm. Notice that the more random flips the less required flips by the algorithm. This makes sense as usually there are more optimal ways to undo random permutations.

Relation Between Flips Done and Time to Solve



This is the number of flips vs the time it took the algorithm to solve it. It appears to be an exponential curve leading me to plot it on a loglog graph shown below.



This is the loglog plot of the number of flips vs the time to solve that number in seconds. Notice the slope has been calculated as .49. This is good timing meaning that the times will increase a very slow rate. However, this is only timing 1000 sized sequences for the efficiency of number of flips. I did not take in consideration the efficiency as a function of the size of the sequences.