Miniproject Evaluation

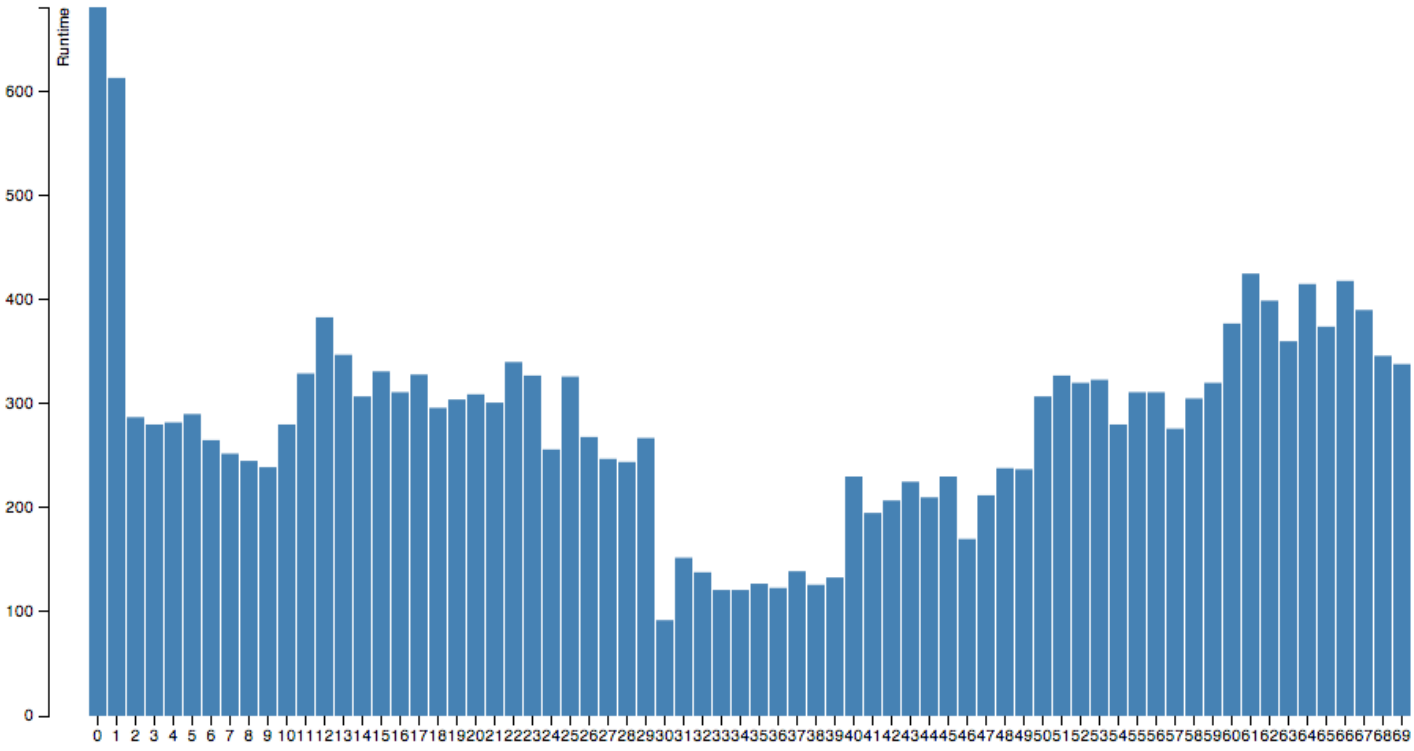
- Team:
    - Vishal Patel
    - Sourabh Desai

Algorithm

- Our team decided to implement a variant of the Greedy Algorithm for Motif Finding. In this variant, a set of possible Motifs were initially created by comparing substrings within two DNA strands. This set of possible motifs was then iteratively compared against other DNA strands within our dataset and possible motifs were pruned out of the set according to a scoring criteria. After going through all DNA strands within the dataset, the top scoring motif still left in the set of possible motifs was reported as the representative motif for the entire dataset.
- For further detail on the algorithm that our team implemented, please refer to Algorithm.pdf

Trends

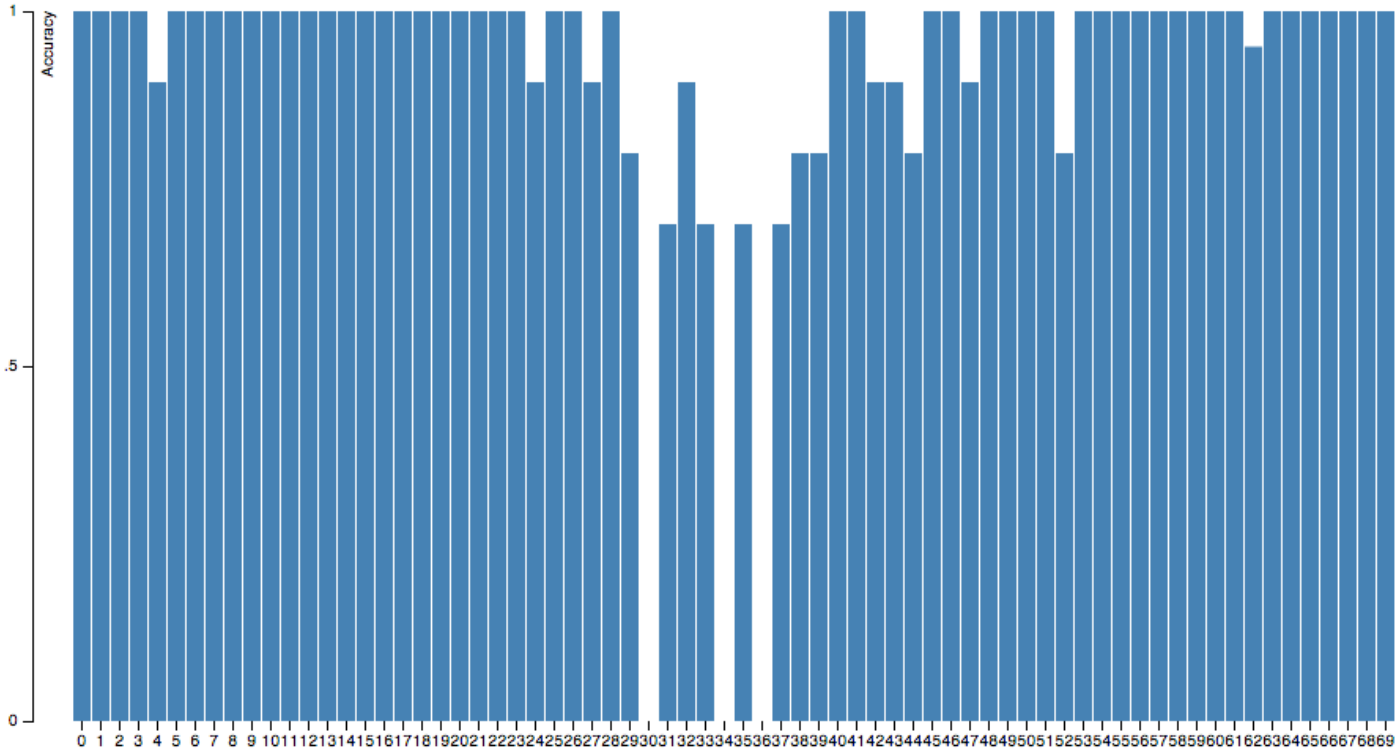
Runtime vs Set Number Barchart



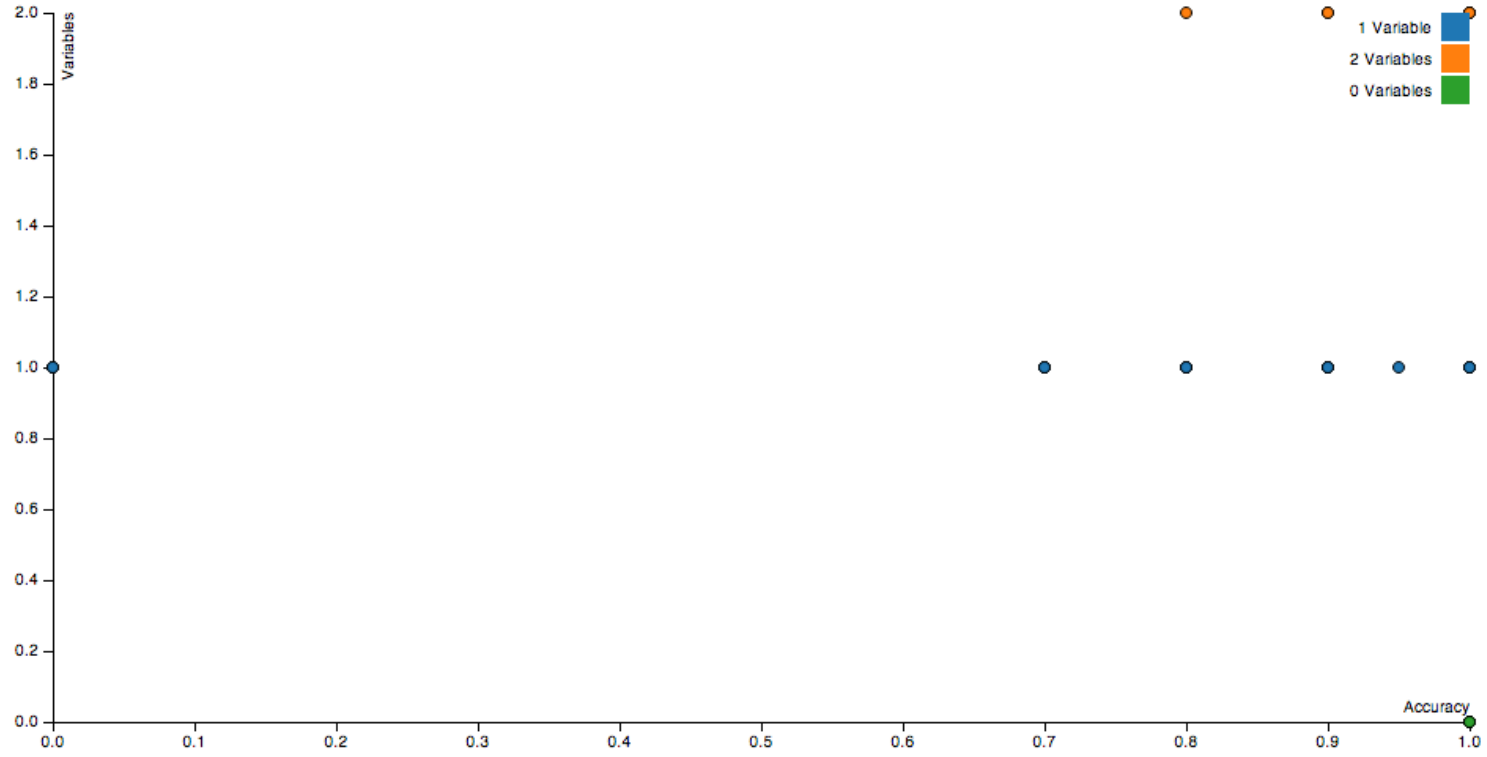
- This barchart visualizes change in Runtime of our Motif finding algorithm across each of the 70 datasets (indexed 0-69).
    - One important point to note is that during our programs execution, the algorithm ran for each dataset in sequence in ascending order. That is, dataset 0 was processed first, and followed by the processing of datasets 1, 2, 3 …
- An interesting trend that can be gleaned from this graph is its somewhat bimodal nature. As can be clearly seen, the first two datasets took exceedingly longer to process than any others. The runtimes are then relatively stable until sets 30-39 where the runtimes become especially low. After set 39, the slowly builds up in length from there onwards to reach the point close to where it was after the first two datasets. We believe this sort of behaviour was expressed because of disk caching. It would make sense that the first two datasets would take longer to build up since the files they are reading the DNA and Motif data from have not been cached. However, after the first two, the runtime decrease could be caused by the caching of our dataset files. This would be an example of caching via spatial locality as the files we likely to have been stored in nearby memory locations on the disk space. The peak of this speed up due to caching would probably have been at sets 30-39. Afterwards, the increase in runtimes may have been caused by a comination of a loss in temporal locality as time progresses throughout the execution of our program as well as crossing a cache line and losing spatial locality. While this explanation *is* speculation, it does do a good job of describing some of the properties of this graph.

Accuracy Evaluations

- For each of the following graphs, Accuracy was measured as the percentage of predicted sites that actually were planted sites. These percentages are given here in decimal form.



- This barchart visualizes change in Accuracy of our Motif finding algorithm across each of the 70 datasets (indexed 0-69).
- As can be seen, most datasets were evaluated to have 100% accuracy. Despite this, there were a handful that were significantly less accurate.



- This scatterplot represents the relationship between Accuracy of our Algorithm on each dataset and the number of variable motif positions for that motif.
    - It is worth noting that there does not appear to be that many points on this graph. This is simply because of a large number

of overlapping points. There are in fact 70 points on this graph (1 for each dataset), its just that a lot of them overlap each other.

- As can be seen, our algorithm had perfect accuracy for all planted motifs with 0 variable positions. It had done worse for planted motifs with variable positions. One may say that there was more Accuracy with planted motifs with 1 variable position than those with 2 variable positions, however this cannot necessarily be gleaned from the data as there were significantly more datasets with a 1 variable position planted motif than 2. Thus, it could just be that there were more *"chances"* for our algorithm to have sub-par accuracy with datasets with planted motifs of 1 variable position.
    - The average accuracy for each number of variable posititions is given below:
        - 0 Variable Position : 100%
        - 1 Variable Position : 88.9%
        - 2 Variable Position : 96%
    - While, as can be seen, our algorithm did do worse *on average* for datasets with planted motifs of 1 variable position, it is our strong belief that our algorithm must be run on more datasets of 1 and 2 variable position planted motifs before being able to come to a safe conclusion.