**Method 1:**

1. Process raw PBM data
2. Name- make sure all sequences have a name containing the strings “Bound”, “Neg”, “Flank”, or “PosCtrl2”
3. 34mer- cut off one nucleotide from each end of the sequence to turn the 36mer into 34mer. This is because in PBM data, each 34mer sequence variation has nucleotide “A” or “T” attached to either end to make a 36 mer so that the primer sequence on one end of sequence doesn’t attach to sequence and create a new core sequence binding site at that area.
4. Core- take the reverse complement of all sequences with core “CCGC” to make them have core “GCGG. Pass all sequences with core of “GCGC” or “GCGG”.
5. Cutoff- make sure sequence’s orientation difference has absolute value less than cutoff score, which is found by testing multiple different values to determine the one that produces best R^2 values
6. Flank cores- make sure sequences don’t contain a core of “GCGC” or “GCGG” in the farthest flanks on either side of 11 bases long.
7. Get duplicated sequences and take median of the log intensity scores. Median is used because experimental data could have produced extremes, and so mean is not practical because it could skew data… median is used more often with biological data.
8. Partition data
9. Randomly shuffle data and split into training (80% of data) and testing (20% of data) data.
10. Repeat step 1 ten times to get 10 different sets of training and testing data
11. Preprocessing
12. Core- select the desired sequences with specified core (either “GCGC” or “GCGG”) from processed data
13. Reformat- convert sequences and their log intensity scores into type of data LIBSVM can use. List all features and set feature number to value of 1 if the sequence has the specific feature
14. Training
15. Train training data into SVR with cross validation of 5. This partitions training data into 5 subsections so that 4 subsections can be used for training and the remaining for testing. Thus 5 different models can be created and tested to get one combined R^2 value for each parameter pairing. Without cross validation, data can be over fitted into the model, and there is not enough data to test the overfitting. Get accuracy of that model with R^2 value for the parameter settings inputted into the SVR.
16. Grid Search
17. Aim is to find optimal parameter settings of c (cost variable) and p (epsilon variable) to produce best prediction model
18. Course- Apply course grid search by testing pairings of c consisting of 2^-7, 2^-6., 2^-5, … 1 and of p consisting of 2^-9, 2^-8, 2^-7, … 2^-2, creating an 8x8 grid of R^2 values.
19. Fine- get highest R^2 value and test pairs of c and p values around it in a 7x7 box of R^2 values.
20. Prediction
21. Get top 5 R^2 values from fine grid search and use their c and p parameter settings to train on entire training data, creating 5 different models.
22. Use the resulting models to test on unused testing data to get final R^2 values for each of the 5 models. Select the model with the best R^2 value.
23. Feature Weights
24. Get the model and use matrix multiplication to get the weights of each feature from the model. Equation is y = w\_1\*x\_1 + w\_2\*x\_2+…
25. Plot the weights on a clustered column graph with features on x axis and feature weights on y axis

Significant features:

1. Filter- Choose all features with weights greater than a cutoff value
2. Cutoff value determined by finding half of the largest feature weight
3. Visualization- show what each feature means by showing the sequence value and the position and showing it visually

Scatter Plot Graphs

1. Compare two sets of values ie. E2F1 GCGC vs GCGG or GCGC E2F1 vs E2F4
2. Use R to plot on scatter plot with one set of values on x axis and other set of values on y axis
3. Normalize data by turning highest value weight into 1 and leaving 0 as 0 and scaling the rest of the data accordingly. Normalize the data because different training and testing datasets were used to create both sets of values, and so similar feature weights do not mean the same thing for both sets of values. 0 on the unnormalized data must remain 0 as this means that the feature has no effect on the SVR model, and so must not be taken into account.
4. Draw y=x line on graph as any points far away from graph have different feature weights between the two sets of data, showing a discrepancy of why they both bind differently. If they bind the same, theoretically all points would be on the y=x line.
5. OPTIONAL- filter out points close to (0, 0).
6. Find distance of each point from the y=x line as having greater distance from y=x means that there is greater deviance in how both sets of values bind with that specific feature.
7. Put distances in descending order with their respective features. Find the feature’s sequence and position and display it visually.

*n = # of features*

Ranking:

1. Rank every feature weight from 1-n with 1 being the highest weight.
2. Take out not significant features by filtering out all features with weights less than cutoff.
3. Determine weight cutoff by plotting each weight in density graph with feature weights on one side and frequency on other side. Take mean (around 0 theoretically) and std dev. So, filter out features with weights of mean +- sd on both sides.
4. Find absolute value of difference of both weights of same feature and put in descending order
5. Plot on scatter plot with both feature weights on axes.

**Method 2:**

1. Repeat method 1 but with different data.
2. Process raw PBM data the same way and then take the ratio of the log intensities of the E2F1 and E2F4 data for shared sequences in the processed data.
3. Train, use grid search, test, and get the feature weights from the new data of the ratio of log intensities.