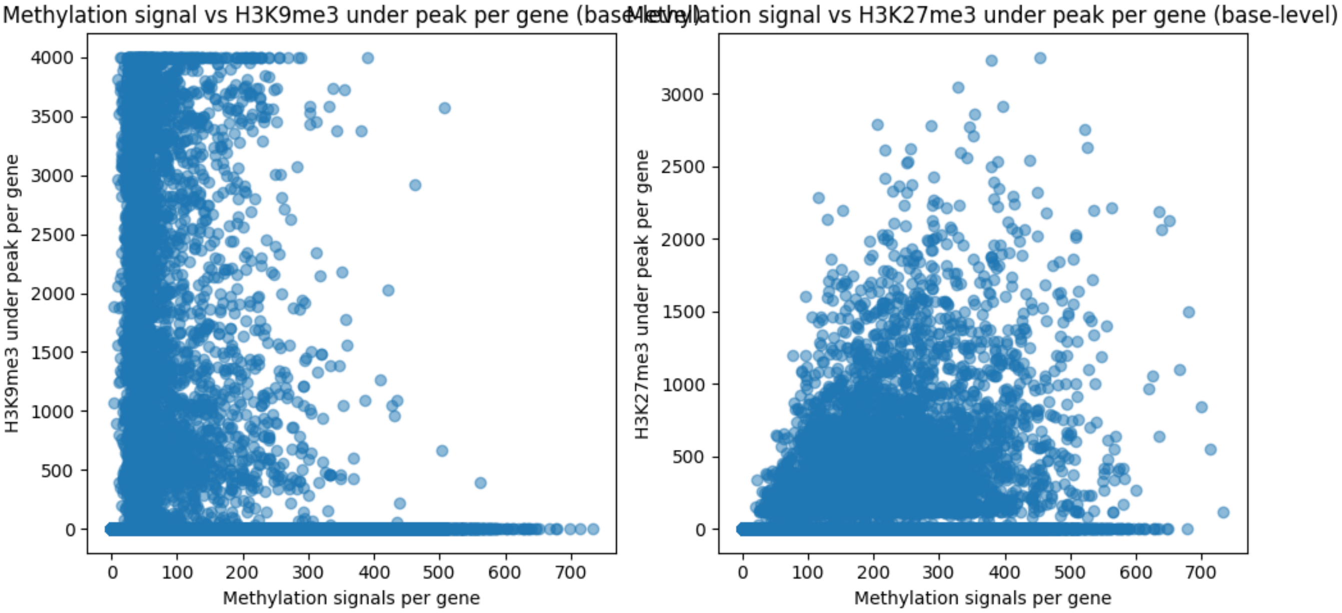
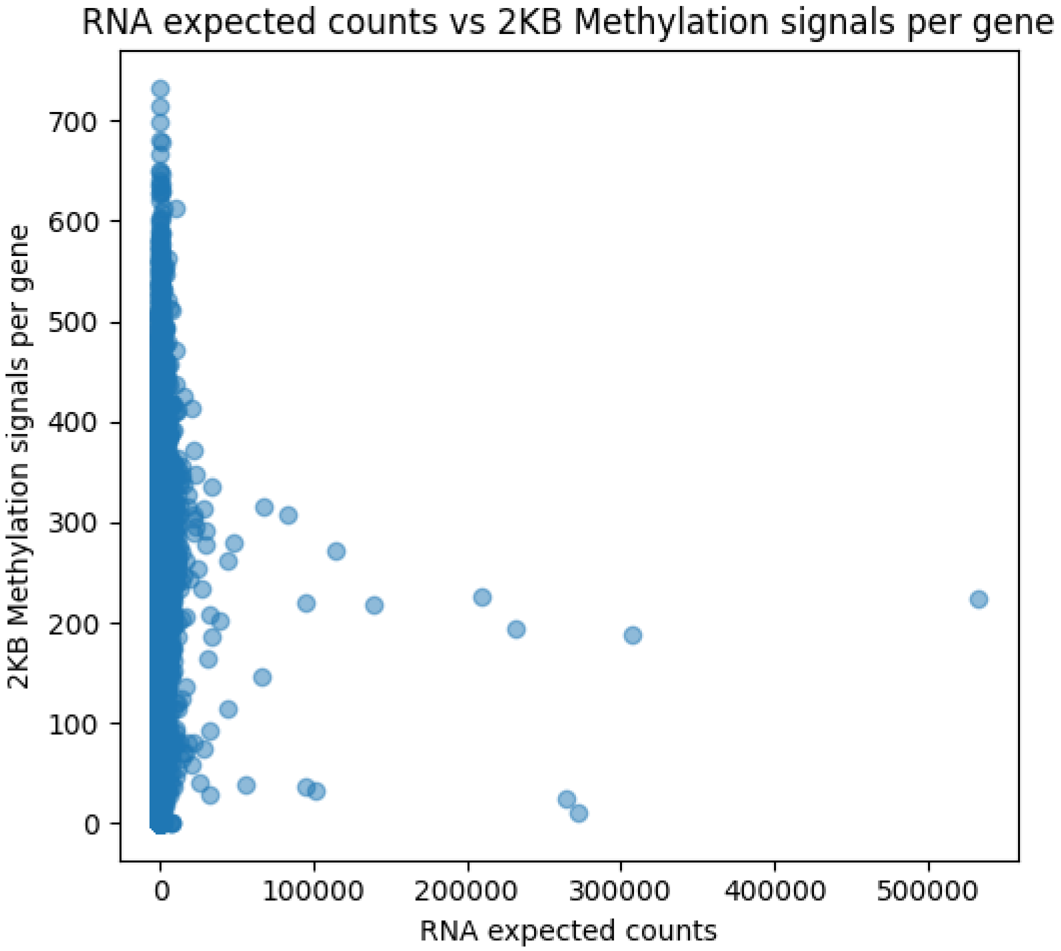
Results

Exploratory Analysis

- Plotting expected counts, DNAm and histone peaks against one another
- Adjusted gene annotations to +-2KB around TSS





Initial modelling

Utilising a 58780, 4000, 3 matrix created from the data

- 58780 genes
- 4000 nucleotides per gene (+- 2KB around TSS)
- 3 modification types (DNAm, K9, K27)

This 3D matrix is then converted into 58780 1-dimensional 12000 shape arrays (to be fed to ML models as single instances)

- 12000 is derived by concatenating the three 4000 long modification data into a single row of 12000

Stage 1: Use regression (tree-based) models to predict expected count (using mean squared error)

Model Type	MSE	RMSE
Random Forest	6106315.026	
XGBoost	8576452.711	
Gradient Boosting Regressor	6577097.108	
LightGBM Regressor	5931371.431	
CatBoost Regressor	6610849.514	
Support Vector Regressor	5542019.612	
AdaBoost Regressor	1527278184.653	

Comments:

- SVR saw best performance (potentially due to high-dimensional data)
- Followed by LightGBM (good for large datasets)
- Followed by Random Forest

Stage 2: Use classification (tree-based) models to predict 0 expected count or non-0 expected count (using precision, recall and F1)

Model Type	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.7509	0.6692	0.6569	0.6630
XGBoost	0.7574	0.7261	0.5616	0.6333
LightGBM	0.7666	0.7317	0.5912	0.6540
Support Vector Machine	0.7726	0.7798	0.5442	0.6411

Comments:

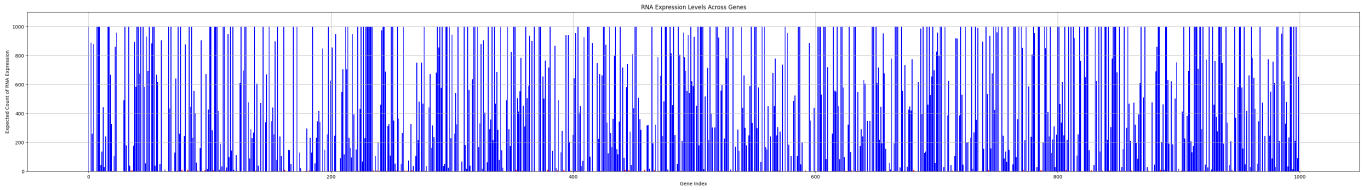
- More balanced prediciton of positive and negative classes for RF, followed by LightGBM with the second highest F1 score

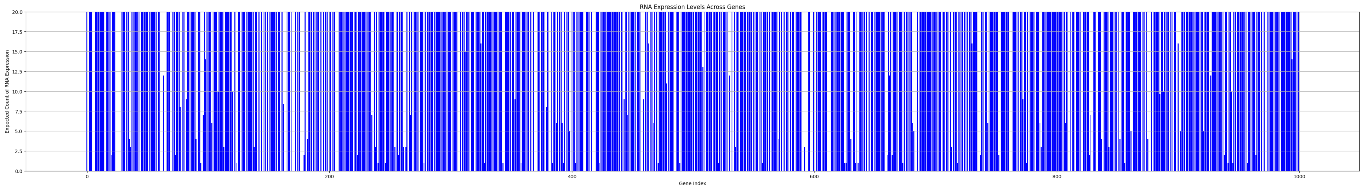
- SVM saw the highest accuracy, with decent ability to predict positive class (non-silent genes) but poorer ability to predict negative class
- The same applies for XGBoost and LightGBM (although LightGBM is slightly better at predicting the negative class)
- potentially an ensemble model could see more accurate results?

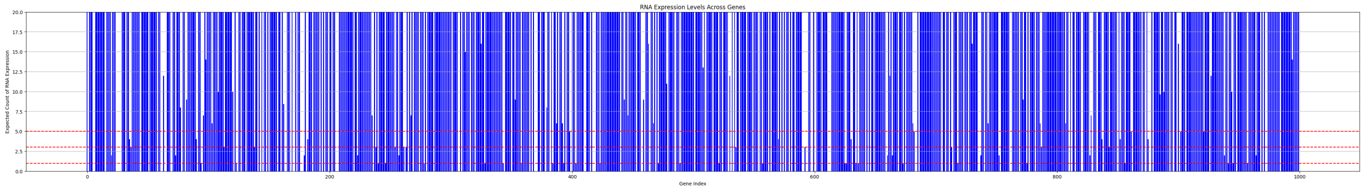
Next steps:

- no need to standardise counts for regressor (would be more relevant if comparing between genes - making the assumption that measurement of noise is fair across the dataset)
- Would be helpful to see plot of expression against genome (expected count on one axis)
- Would be good to try a different threshold (slightly higher than 0 - use above plot to see where peaks are - will ultimately be an arbitrary choice but we assume that some thresholds would be better than others)

Plotting expression (to select a threshold for classification)







- selected 1000 genes and plotted expected counts against the index of that gene (and altered the ceiling to allow for a closer look at the lower level expression without altering the appearance of the visualisation, such as through the use of log)
- Will try thresholds 1, 3, and 5 and compare performance on the above default classification models

Comparison of thresholds

Threshold: 1

Model Type	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.7815	0.6905	0.6229	0.6549
XGBoost	0.7774	0.7165	0.5482	0.6212
LightGBM	0.7908	0.7297	0.5904	0.6527
Support Vector Machine	-	-	-	-

Threshold: 3

Model Type	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.8053	0.7232	0.5498	0.6247
XGBoost	0.8015	0.7117	0.5492	0.6200
LightGBM	0.8095	0.7215	0.5763	0.6408
Support Vector Machine	-	-	-	-

Threshold: 5

Model Type	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.8037	0.7203	0.4767	0.5737
XGBoost	0.8093	0.7077	0.5313	0.6069
LightGBM	0.8187	0.7171	0.5712	0.6359
Support Vector Machine	-	-	-	-

Notes:

- SVM still running
- Generally, it seems that increasing the threshold assists the model in predicting the positive class (non-silent) and makes it harder to predict the negative class (silent genes). Whilst accuracy and precision increase, recall decreases. Using the F1 harmonic mean may be more useful as a indicator of success considering our goals of classifying silent genes (as opposed to accuracy).
- Max F1 reached = RF classifier witha threshold of 0 (0.66)
- As we increase the threshold, we see a drop across all models in recall and F1, however, the drop for the LightGBM model is more slight than the others

It should be noted that the size of the classes is unbalanced with a greater number of silent genes than non-silent. The genes with low expression counts are close to our threshold and difficult to classify. Continue with more complex models?

Class sizes:

Threshold	Silent (negative class)	Non-silent (positive class)
0	36642	22138
1	39025	19755
3	41265	17515
5	42364	16416

Additonal work:

- Was curious to see the impact of combining such models
- Ensemble model combining lightgbm and SVM (with soft voting)
- threshold = 0
- still running***

Accuracy	Precision	Recall	F1
-	-	-	-