Real Data: 16s rRNA (Baseline 0.25)

Yichen Han

2024-08-13

We want to try out the least-informative baseline: each cell of the one-hot coded matrix is 0.25, standing for equal probability for each base.

Retrained Model

We load the model:

```
## Using checkpoint checkpoints/16S_vs_bacteria_full_2/Ep.008-val_loss0.13-val_acc0.991.hdf5
```

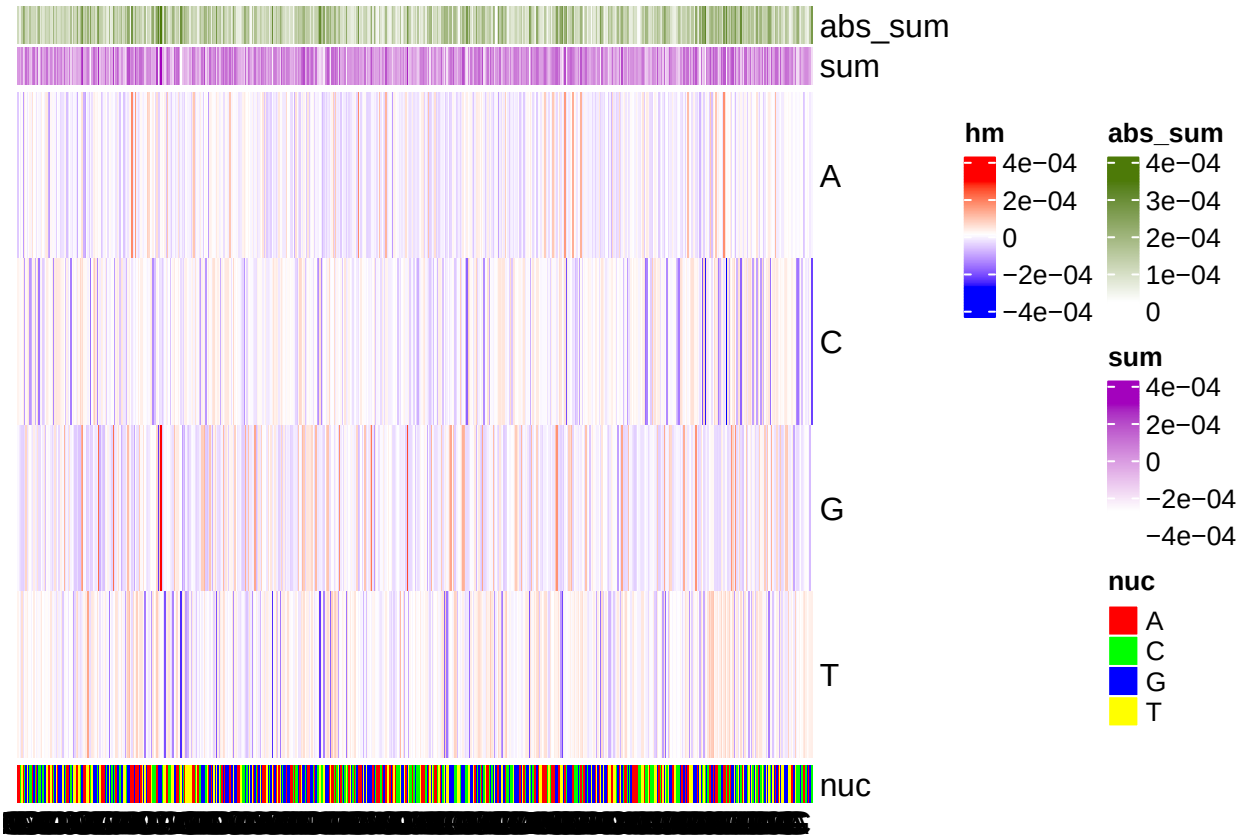
```
## [[1]]
## [[1]]$confusion_matrix
##           Truth
## Prediction  16s bacteria
##   16s      1198        1
##   bacteria   52     1249
##
## [[1]]$accuracy
## [1] 0.9788
##
## [[1]]$categorical_crossentropy_loss
## [1] 0.1410726
##
## [[1]]$AUC
## [1] 0.9999059
##
## [[1]]$AUPRC
## NULL
```

Instance: GCF_001986655.1_ASM198665v1_genomic.16s.fasta 499-1098 bp.

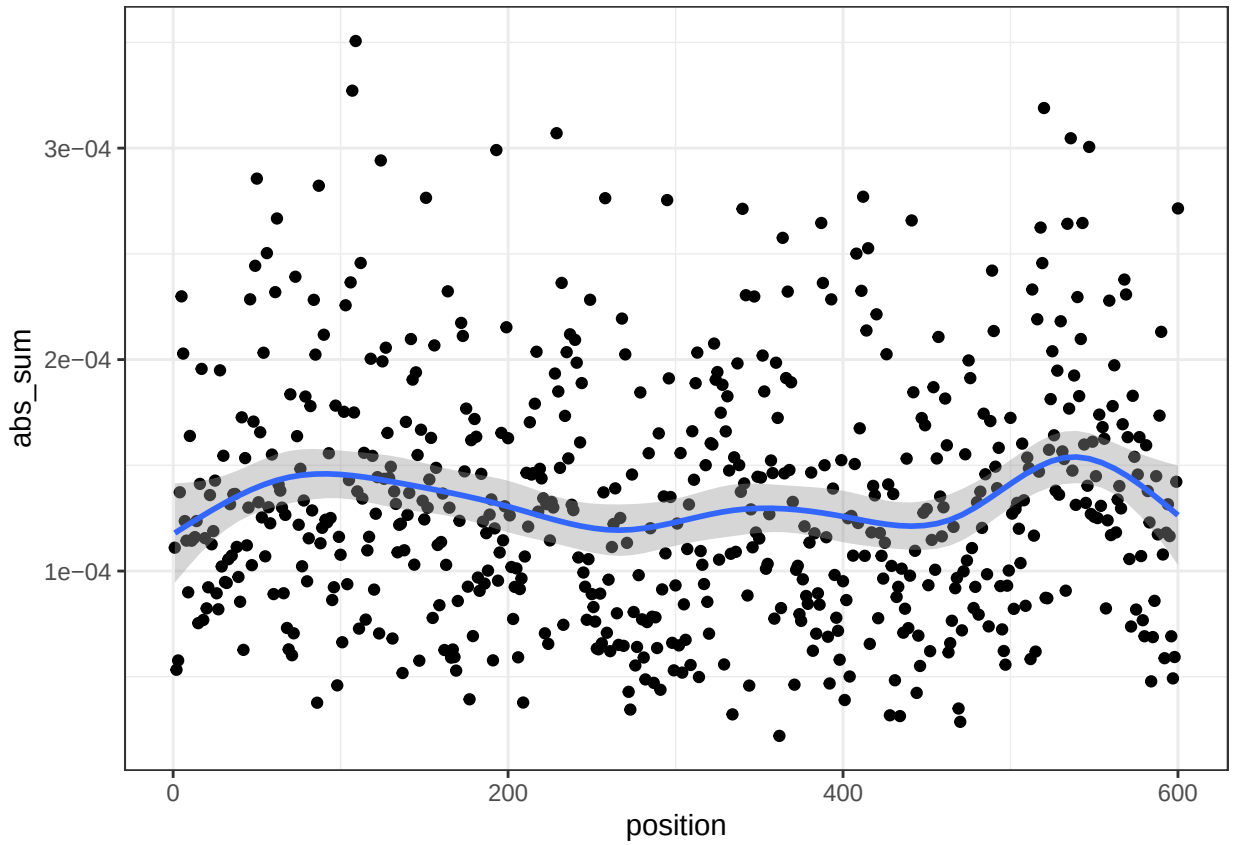
Baseline: 0.25

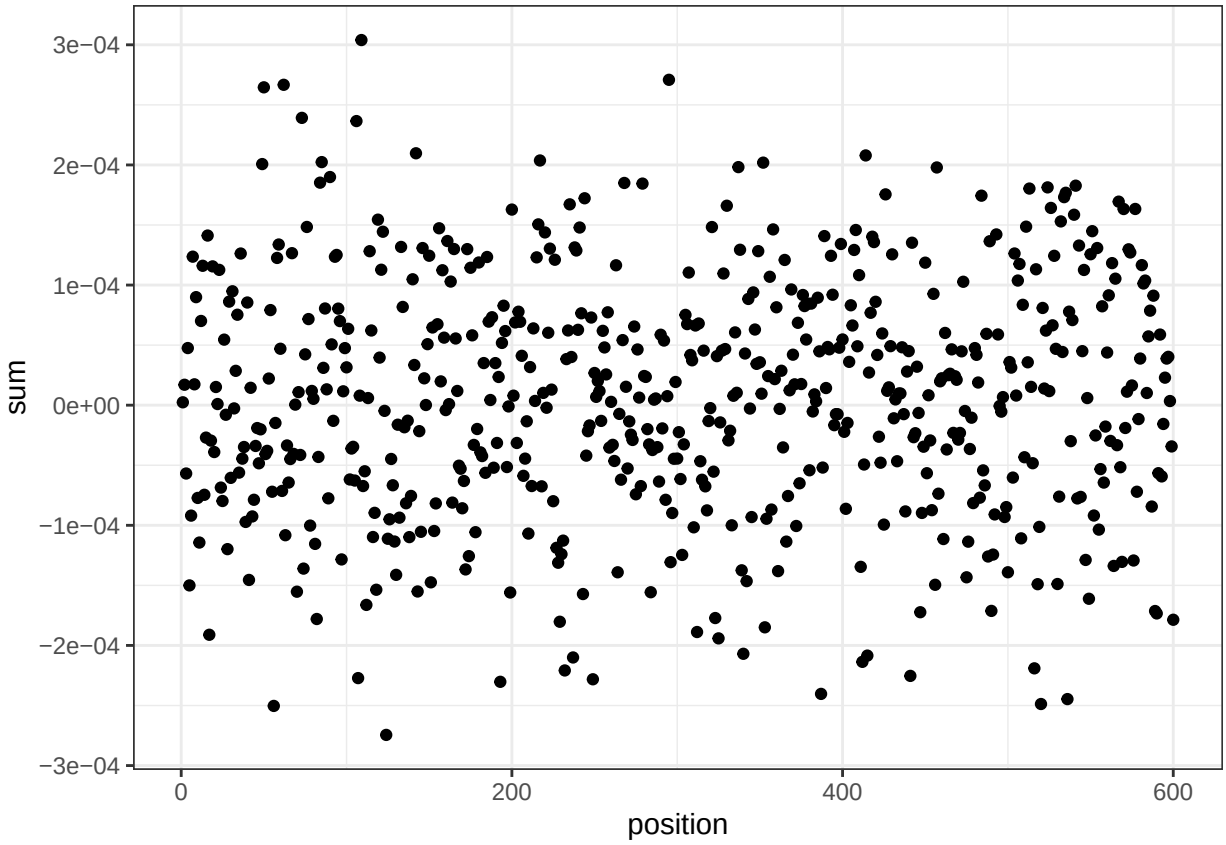
```
##          [,1]       [,2]
## [1,] 0.906086 0.09391395
```

```
## [[1]]
```

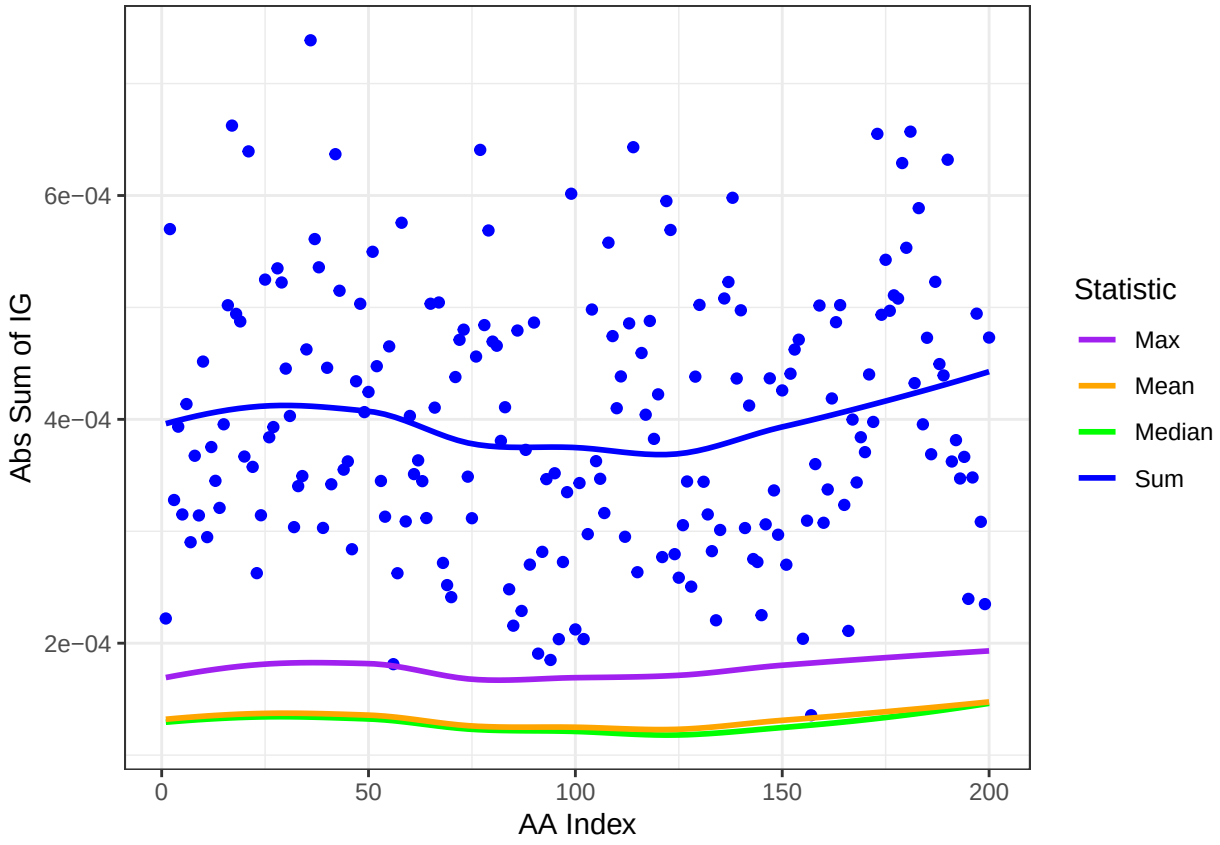



```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```

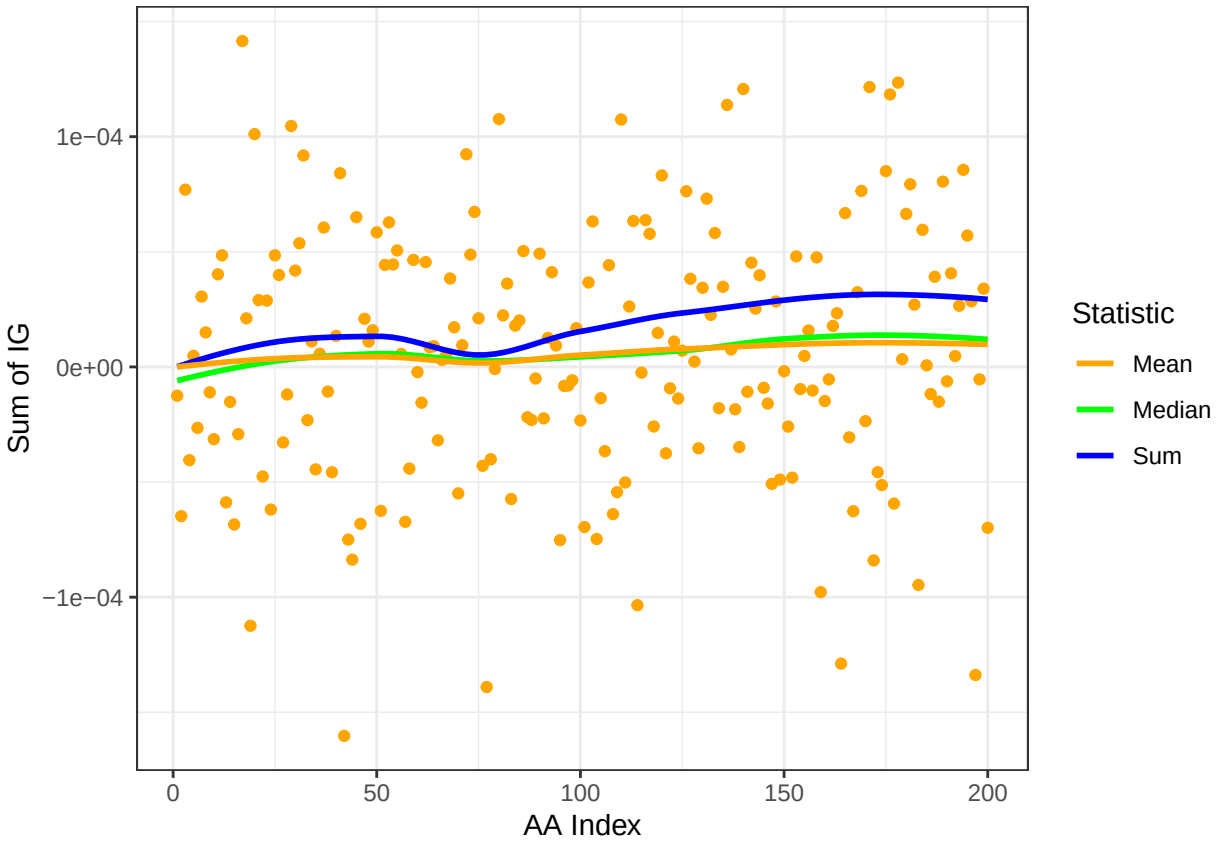

3e−04
2e−04
1e−04
abs_sum
0
200
400
600
position


3e−04
2e−04
1e−04
0e+00
−1e−04
−2e−04
−3e−04
sum
0
200
400
600
position

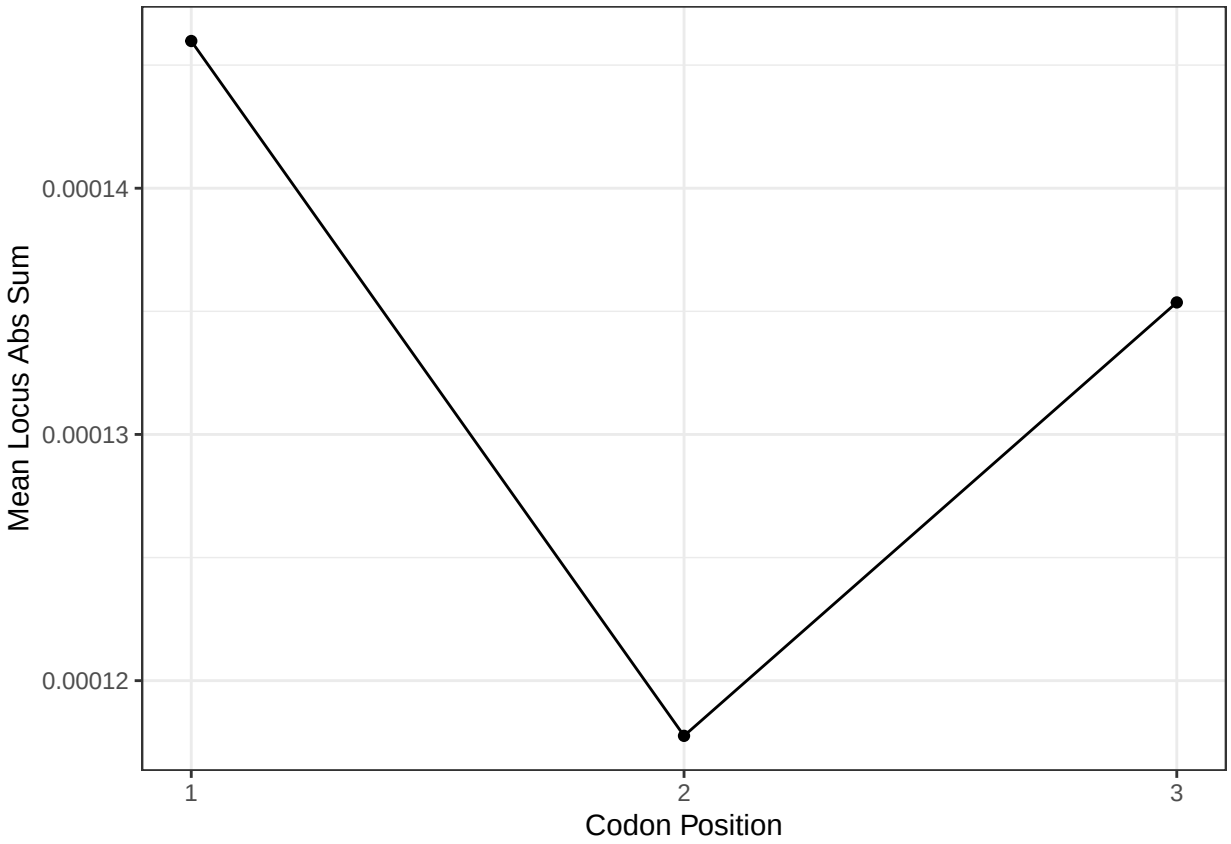
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

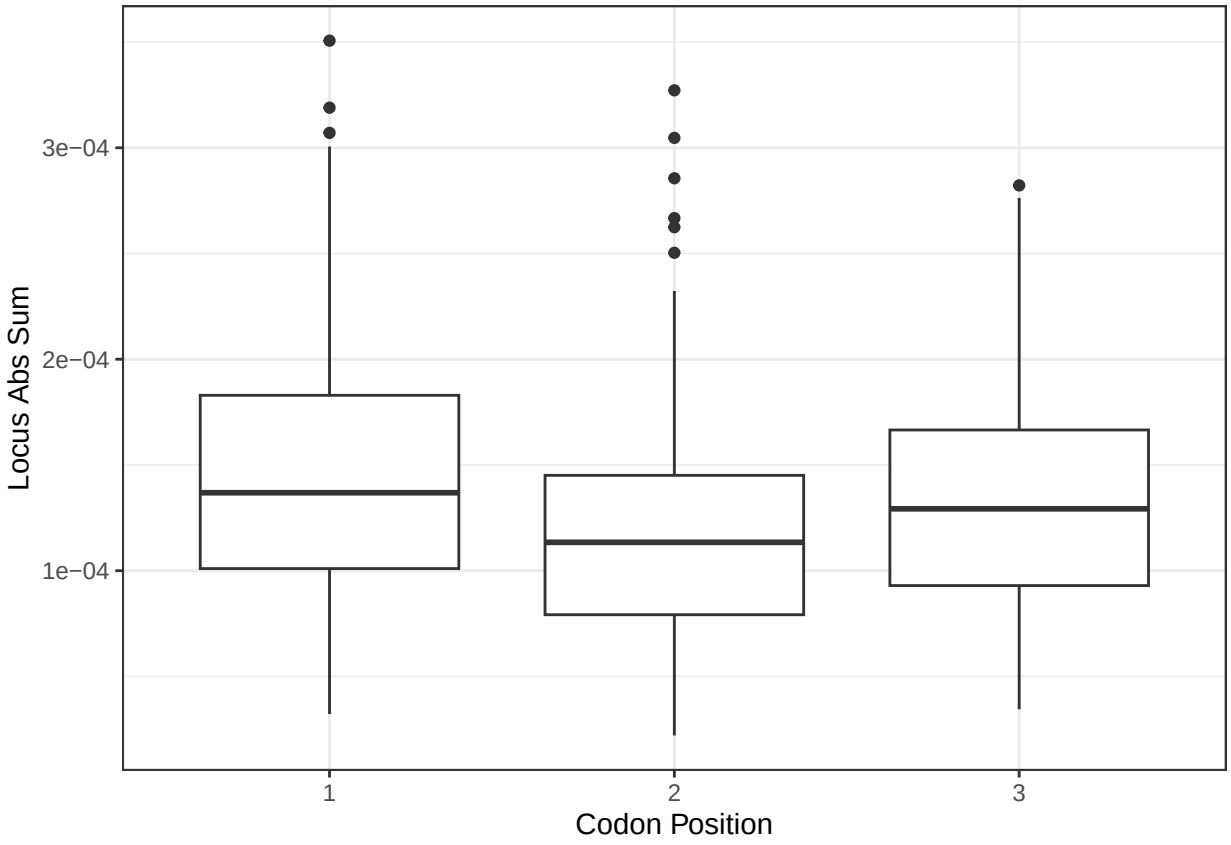



```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



We further studied wobleness (see Durrant & Bhatt). In this case, it is non-existent.


Mean Locus Abs Sum
0.00014
0.00013
0.00012
1
2
3
Codon Position

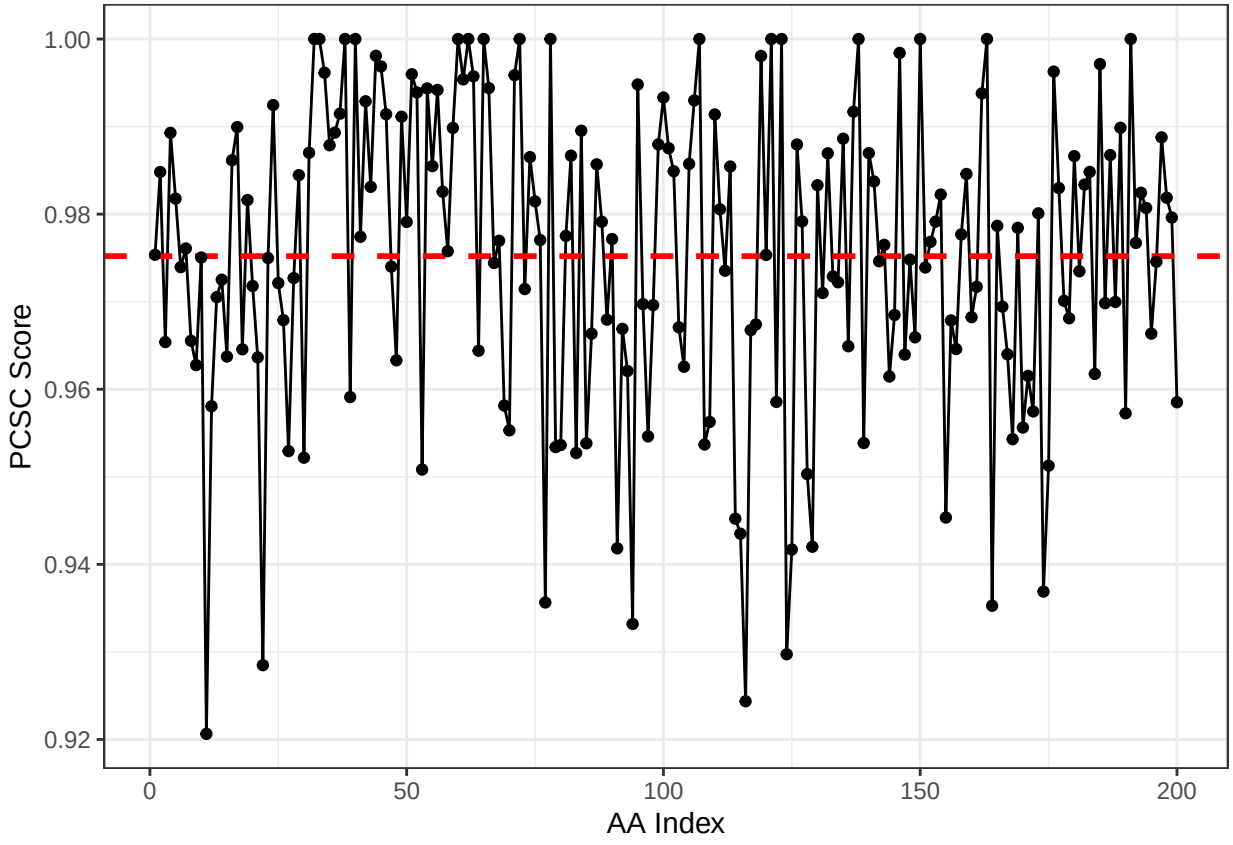

Locus Abs Sum
3e−04
2e−04
1e−04
1
2
3
Codon Position

```
##
## Call:
## lm(formula = abs_sum ~ factor(position), data = codon_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.138e-04 -4.291e-05 -5.882e-06  3.104e-05  2.094e-04
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         1.460e-04  4.010e-06  36.404  < 2e-16 ***
## factor(position)2 -2.822e-05  5.671e-06  -4.976  8.5e-07 ***
## factor(position)3 -1.062e-05  5.671e-06  -1.872   0.0617 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.671e-05 on 597 degrees of freedom
## Multiple R-squared:  0.0406, Adjusted R-squared:  0.03739
## F-statistic: 12.63 on 2 and 597 DF,  p-value: 4.231e-06
```

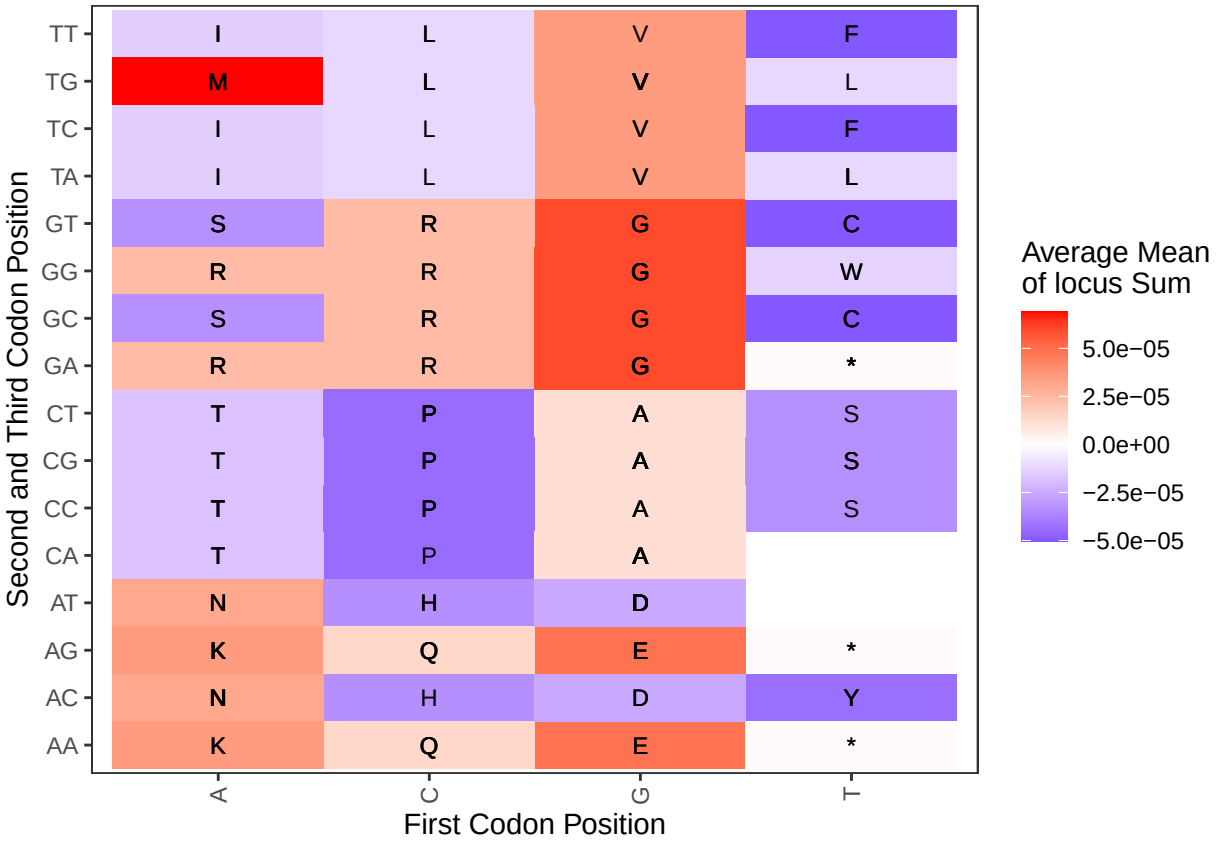
Consistency

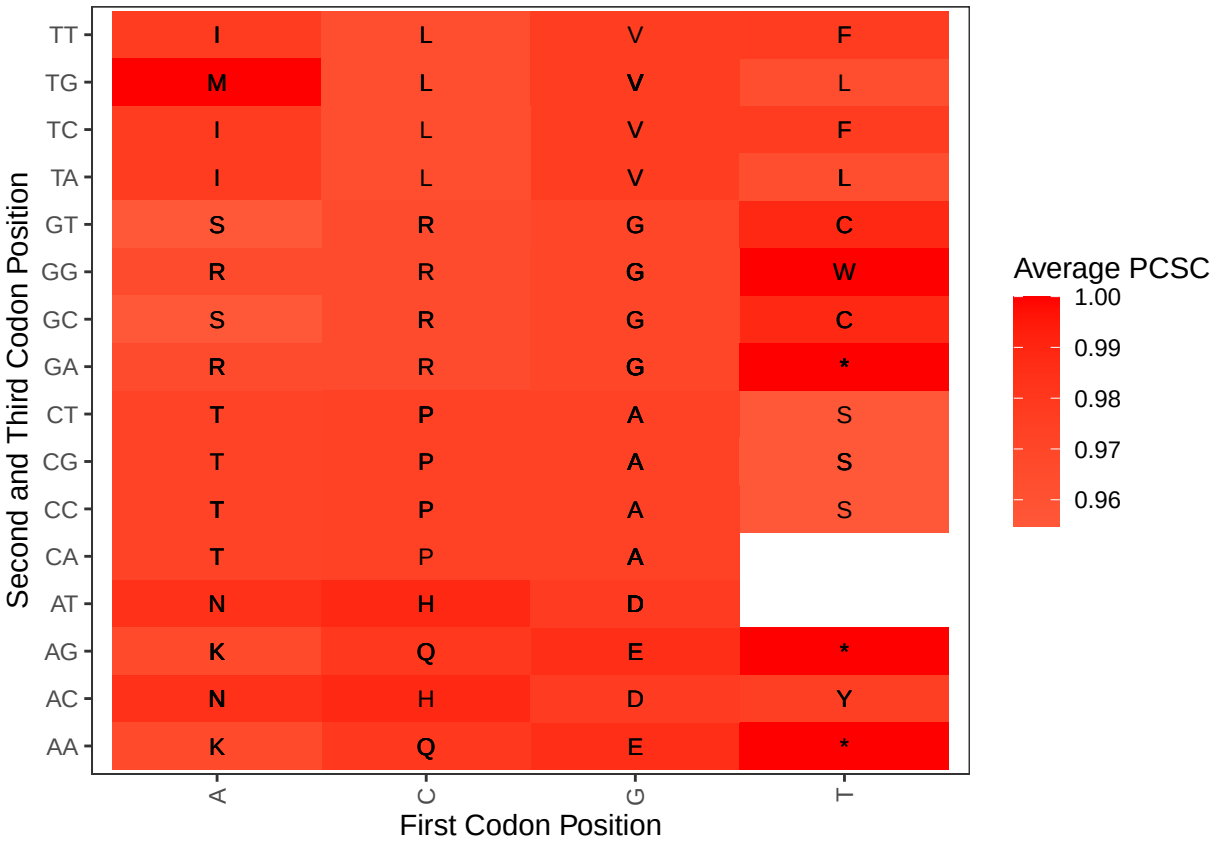
We then calculated PCSC score on our data. With an average score at 0.97, we conclude that the model was able to capture synonymous codons and treats them very consistently.

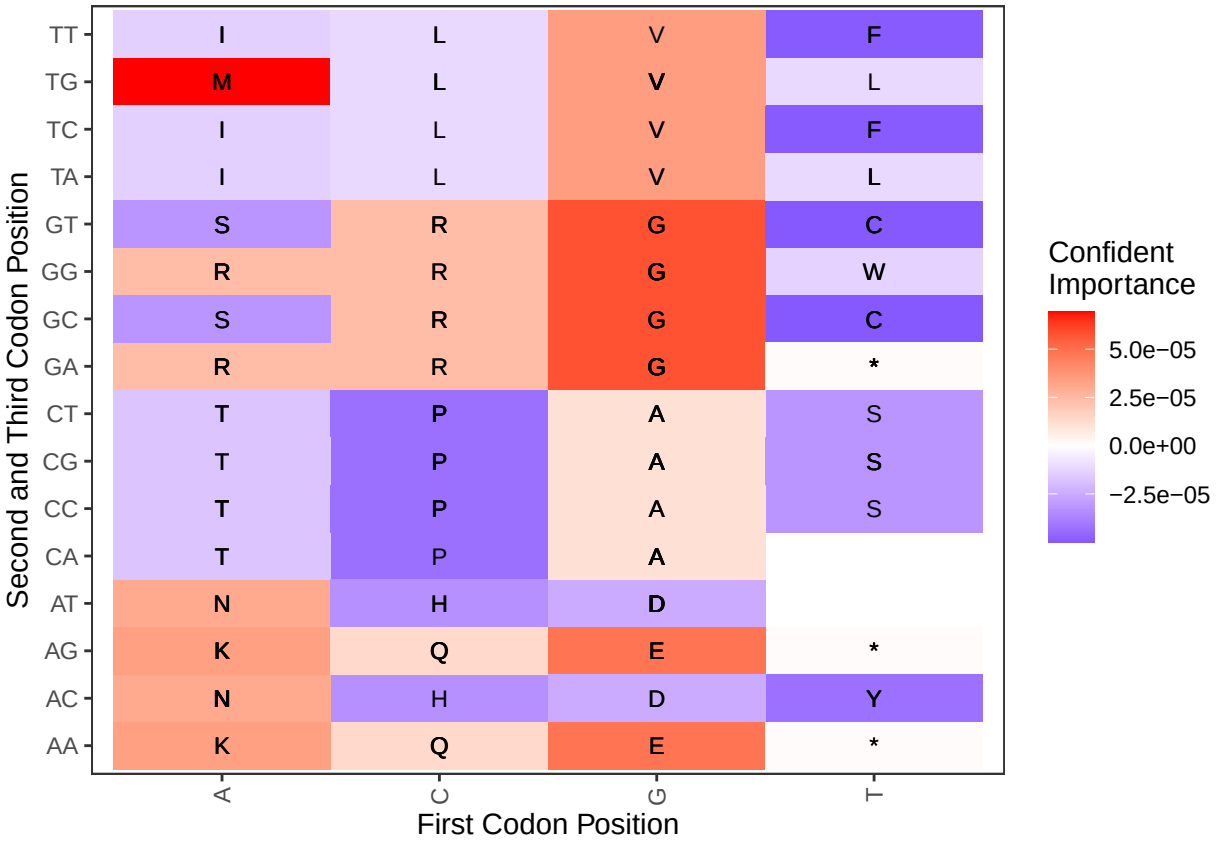
```
## Loaded result_df from existing CSV file.
```



We then generated heatmaps as in RSDexport. The first heatmap visualizes average sum of gradients, the second average consistency based on AA, and the third the "confident importance", which is average sum of gradients multiplied by PCSC score.




Second and Third Codon Position
TT I L V F
TG M L V L
TC I L V F
TA I L V L
GT S R G C
GG R R G W
GC S R G C
GA R R G *
CT T P A S
CG T P A S
CC T P A S
CA T P A
AT N H D
AG K Q E *
AC N H D Y
AA K Q E *
A C G T
First Codon Position
Average PCSC
1.00
0.99
0.98
0.97
0.96


Second and Third Codon Position
TT I L V F
TG M L V L
TC I L V F
TA I L V L
GT S R G C
GG R R G W
GC S R G C
GA R R G *
CT T P A S
CG T P A S
CC T P A S
CA T P A
AT N H D
AG K Q E *
AC N H D Y
AA K Q E *
A C G T
First Codon Position
Confident Importance
5.0e−05
2.5e−05
0.0e+00
−2.5e−05

Feature Selection

We then implemented the feature selection algorithm as by the enhanced IG paper. We do this based on locus level, and sample 50 events for both interest and random group.

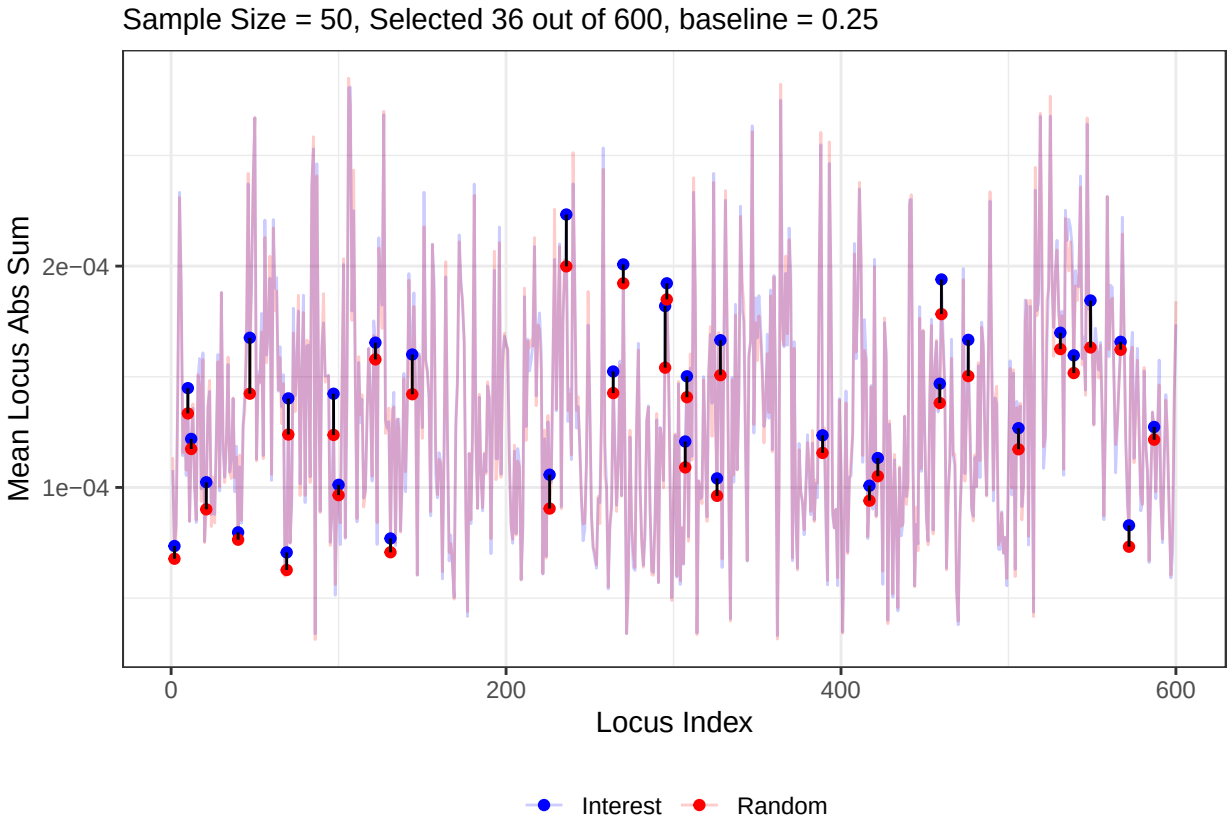
We performed one-sided t-test row-wise (only select positions with absolute sum of gradients larger than the random group). Since each row was tested only once, no multiple testing is present.

In the end, 43 out of 600 loci are identified as having significantly higher absolute sum of gradients than the random group.

We visualize these pairs, and notice that high importance points are not necessarily helpful in distinguishing instances, since the two curves are very similar in shape.

```
## Loaded interest_df from existing CSV file.
```

```
## Loaded random_df from existing CSV file.
```



We further explore the consitution of those important features. We only extracted the indices, and we compare them with one of the instances and extract the AA position of those points.

At the moment, no noticeable difference in GC content or pattern in AA is detected. Notably, 6 Alanine and 5 Asparagine are detected.

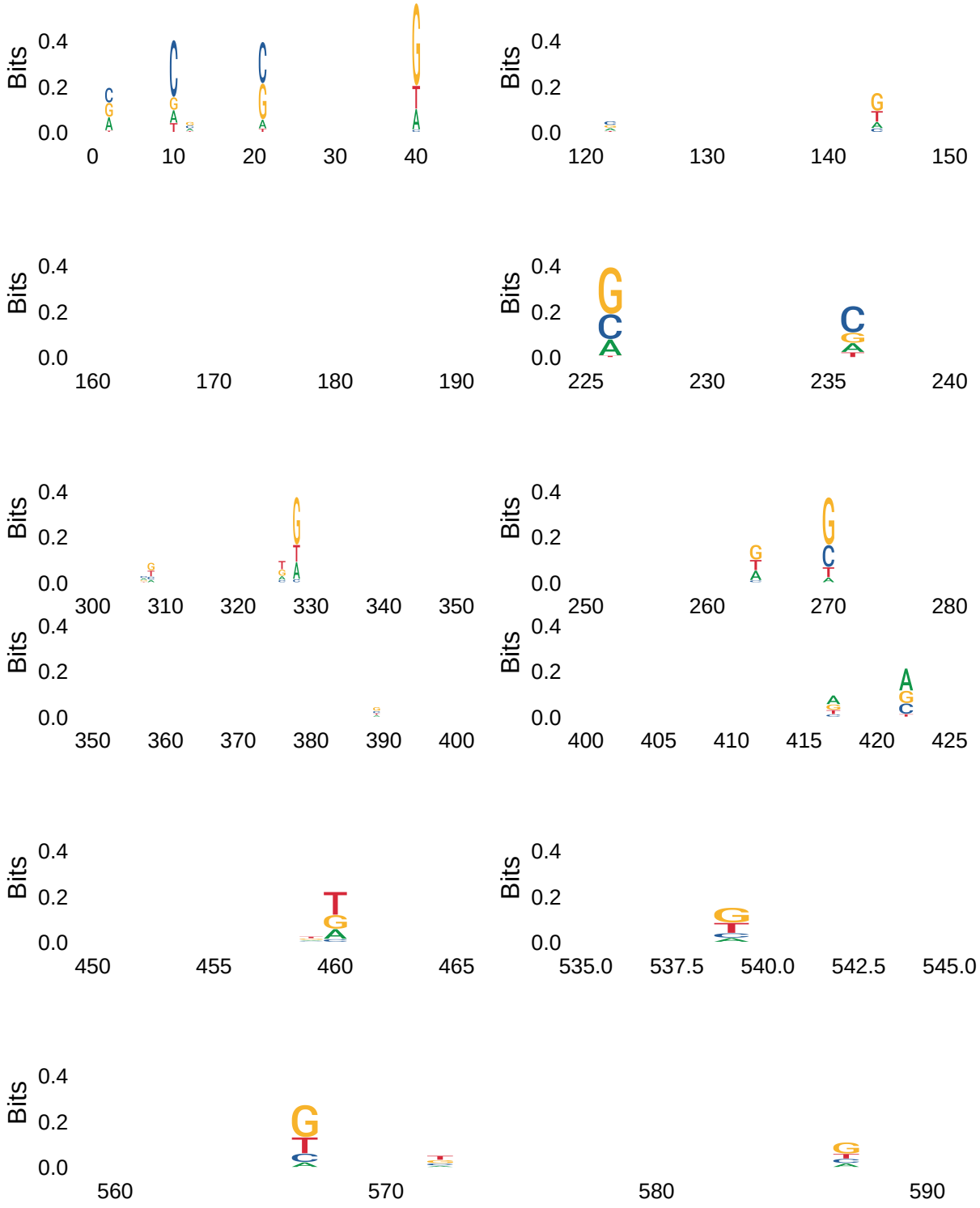
```
## [1] "ACGT content of selected features:"
```

```
## selected_trip
##  A  C  G  T
## 25 32 31 20
```

```
## [1] "Table of corresponding amino acids of selected features:"

## selected_key
## * A C D E F G H I L N P Q R S T V
## 2 6 2 1 1 1 1 1 1 1 5 4 1 3 1 2 3
```

We have visualized the selected features in **a sequence logo plot**.



Adversarial

It seems baseline 0.25 selects better features than a real baseline. Substituting with C reduces confidence for important features to 0.2 and to 0.31 for random features.

```
## [1] "Original Prediction: "
```

```
##          [,1]       [,2]
## [1,] 0.906086 0.09391395
```

```
## [1] "Prediction after substituting important features with C: "
```

```
##           [,1]      [,2]
## [1,] 0.2061469 0.7938532
```

```
## [1] "Prediction after substituting same amount of random features with C: "
```

```
## [1] 0.322091
```

```
## [1] "Prediction after substituting important features with A: "
```

```
##           [,1]       [,2]
## [1,] 0.9216969 0.07830303
```

```
## [1] "Prediction after substituting same amount of random features with A: "
```

```
## [1] 0.9223086
```

```
## [1] "Prediction after substituting important features with 0.25:"
```

```
##           [,1]      [,2]
## [1,] 0.7559541 0.2440458
```

```
## [1] "Prediction after substituting same amount of random features with 0.25:"
```

```
## [1] 0.8068311
```

Input Reconstruction

We tried reconstructing the input sequence using a non-informative baseline, where A,C,G,T each has 0.1 for each position. We attempted 0.25, but the algorithm did not converge. Also choosing a bacteria sub-sequence leads to a non-converging result. This seems to be interesting.

This is done by iteratively doing:

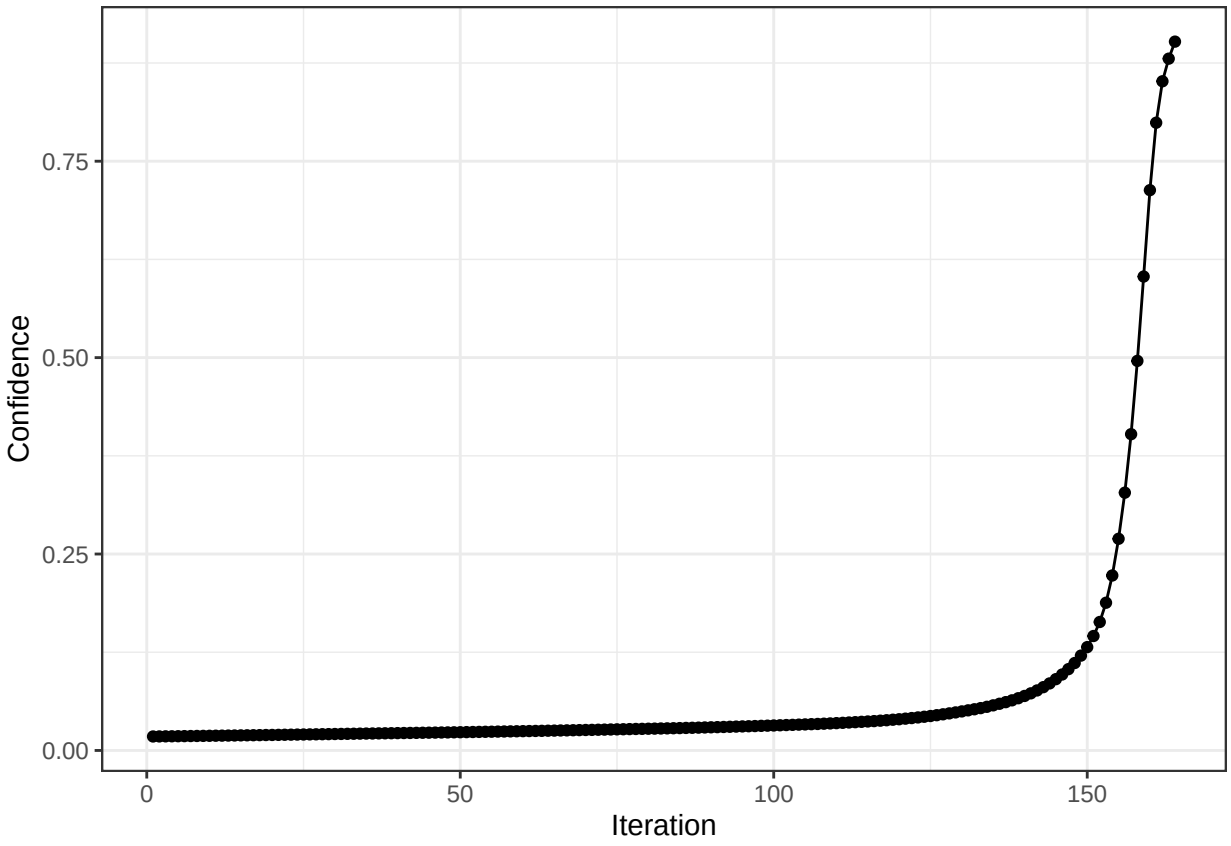
$$X^{(t+1)} = X^{(t)} + \epsilon \cdot IG(X^{(t)})$$

We set $\epsilon = 2$.

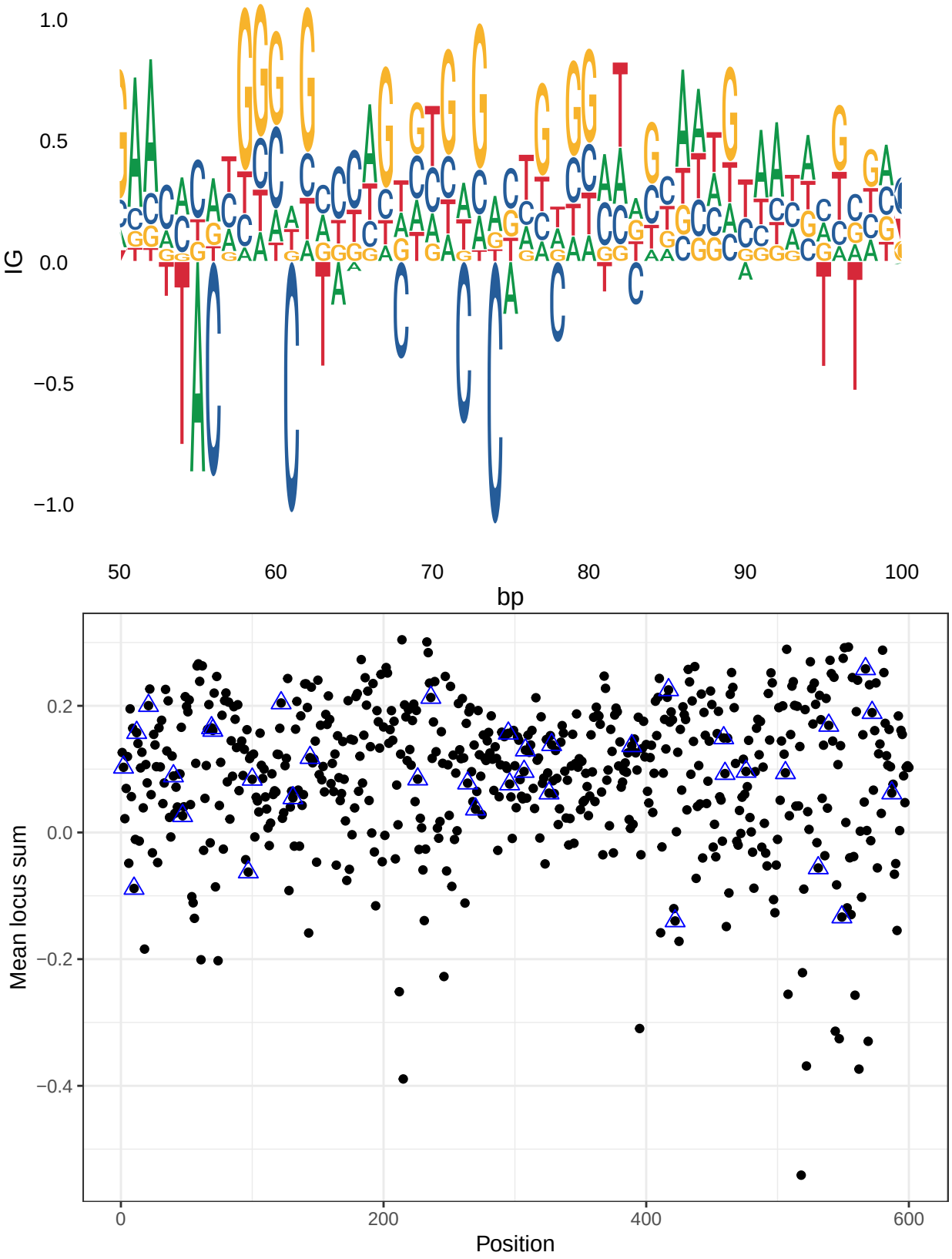
We plot the learning curve and the sequence logo (customed using final IG as y-axis) for a segment of the sequence.

Interpretation: the achieved representation is the least required to be predicted as 16S rRNA gene with a confidence of 90%.

We also plotted the mean of each position with the previously selected important spots marked blue. The value here does not seem to correspond to the identified importance of the position.




```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```



Pay extra attention to the reconstructed sequence logo plot. In the negative region, C and T often have considerably negative weights, meaning they should not appear on this position if the prediction is 16S. This

could explain why the adversarial scenario where more Cs are included caused dramatic drop in confidence.

We also table the matching positions of the reconstructed sequence comparing the original one. Compared to selected features, we see the matching positions seem not to be totally away from the selected ones. Could this be of interest?

```
## [1] "Matched Reconstructed sequence and original sequence, AA position: "
```

```
## AAT ATA GGA GGG GGA AAT AGG AAC ATA GGT GGA GAT AAT GGC TTA CTG GTC AAT GAT AGT
##   1   9  17  20  27  45  53  63  66  68  69  79  80  89  94  97  98 103 104 116
## AGC GGG GGC AAA AAC GGA TAA GAC GAC GGA GGG AGA GGT GTA GGG
## 122 126 129 133 137 140 150 152 162 165 171 176 184 193 194
```

```
## [1] "Selected features, AA position: "
```

```
##  [1]   1   4   4   7  14  16  23  24  33  34  41  44  48  76  79  88  90  99  99
## [20] 103 103 109 110 130 139 141 153 154 159 169 177 180 183 189 191 196
```