Real Data: 16s rRNA (Baseline 0.25)

Yichen Han

2024-08-22

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
```

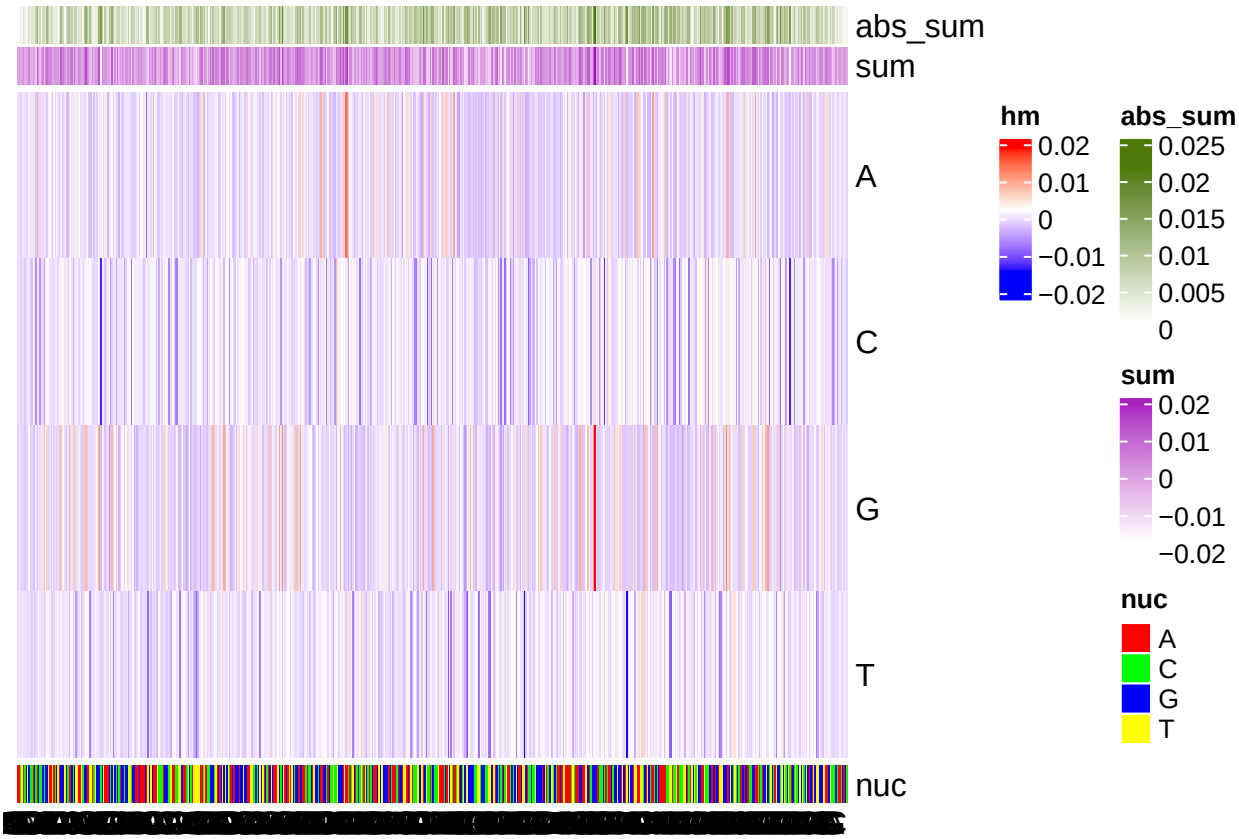
```
## [[1]]
## [[1]]$confusion_matrix
##           Truth
## Prediction  16s bacteria
##   16s      1198        1
##   bacteria   52     1249
##
## [[1]]$accuracy
## [1] 0.9788
##
## [[1]]$categorical_crossentropy_loss
## [1] 0.1410726
##
## [[1]]$AUC
## [1] 0.9999059
##
## [[1]]$AUPRC
## NULL
```

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

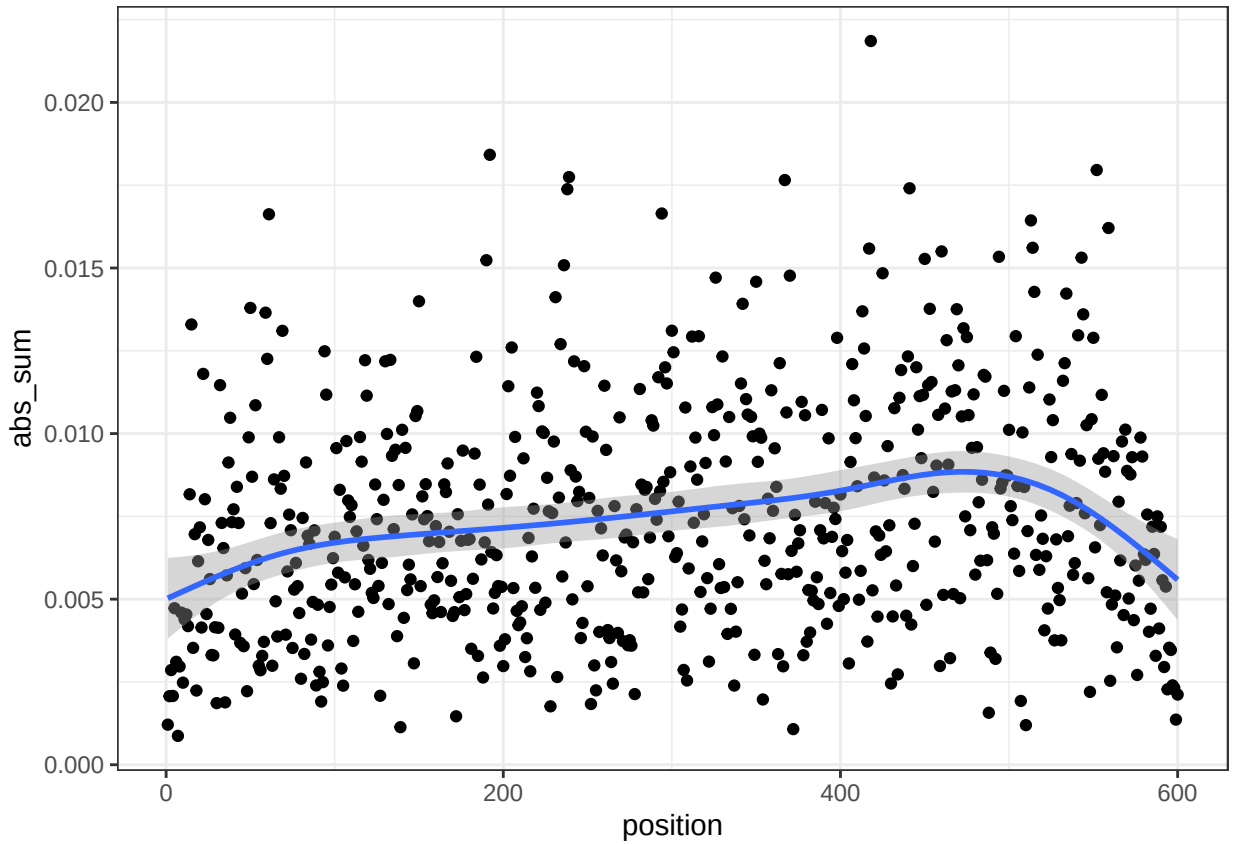
Baseline: 0.25

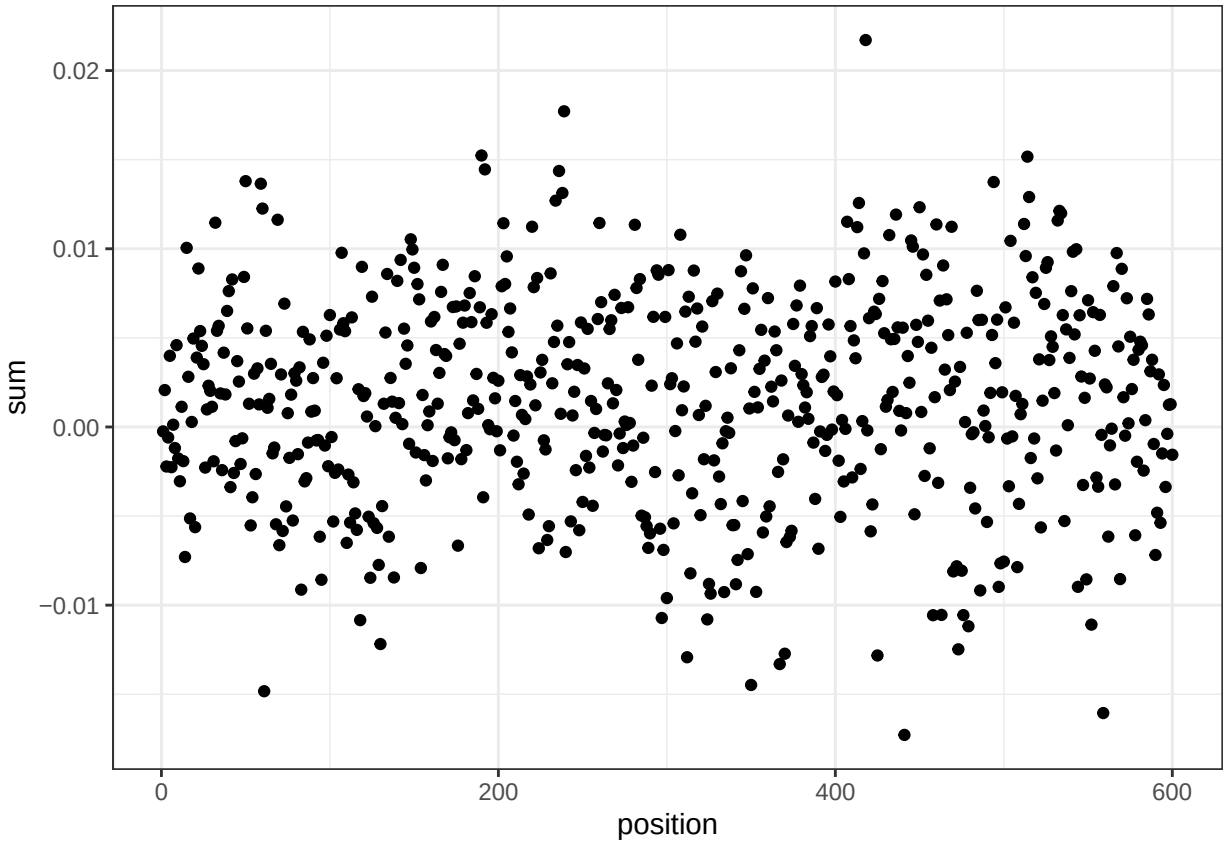
```
##          [,1]       [,2]
## [1,] 0.906086 0.09391395
```

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



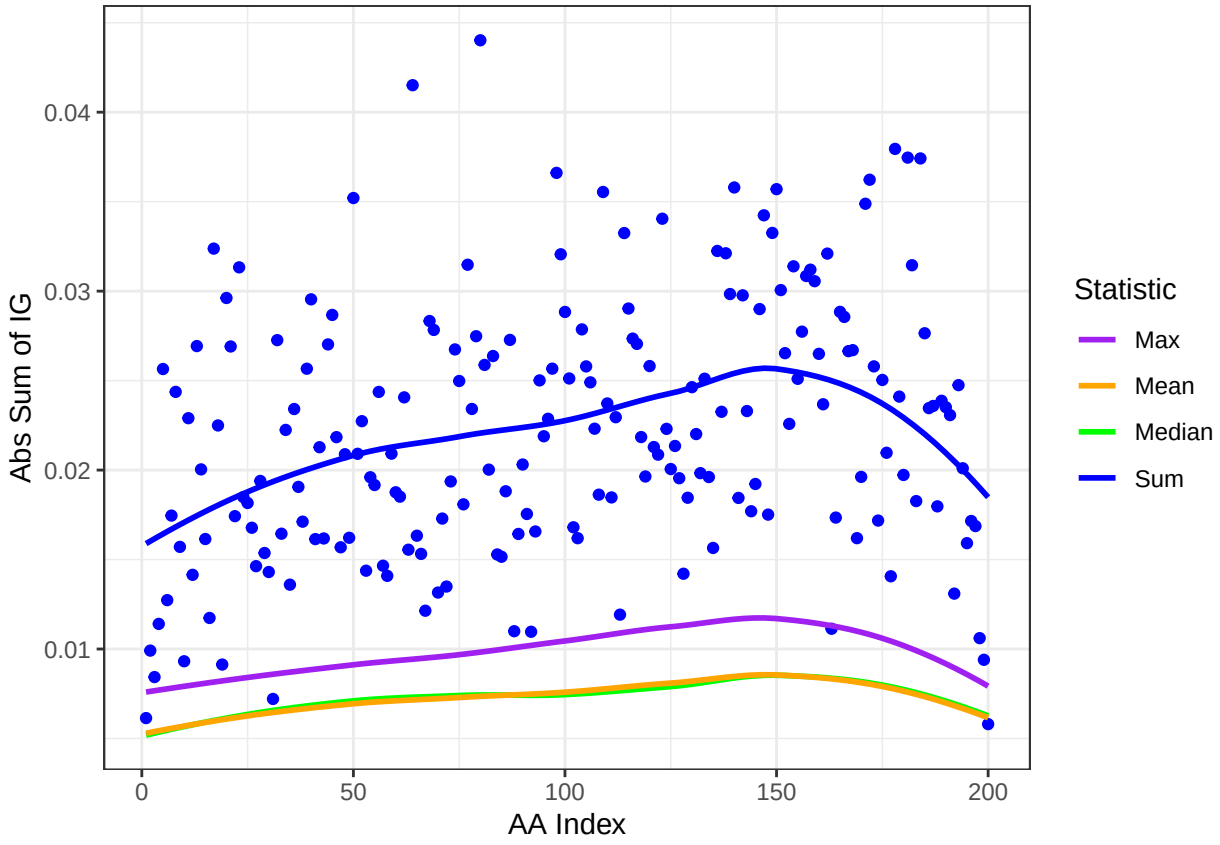

```
## `geom_smooth()` using formula = 'y ~ s(x, bs = "cs")'
```



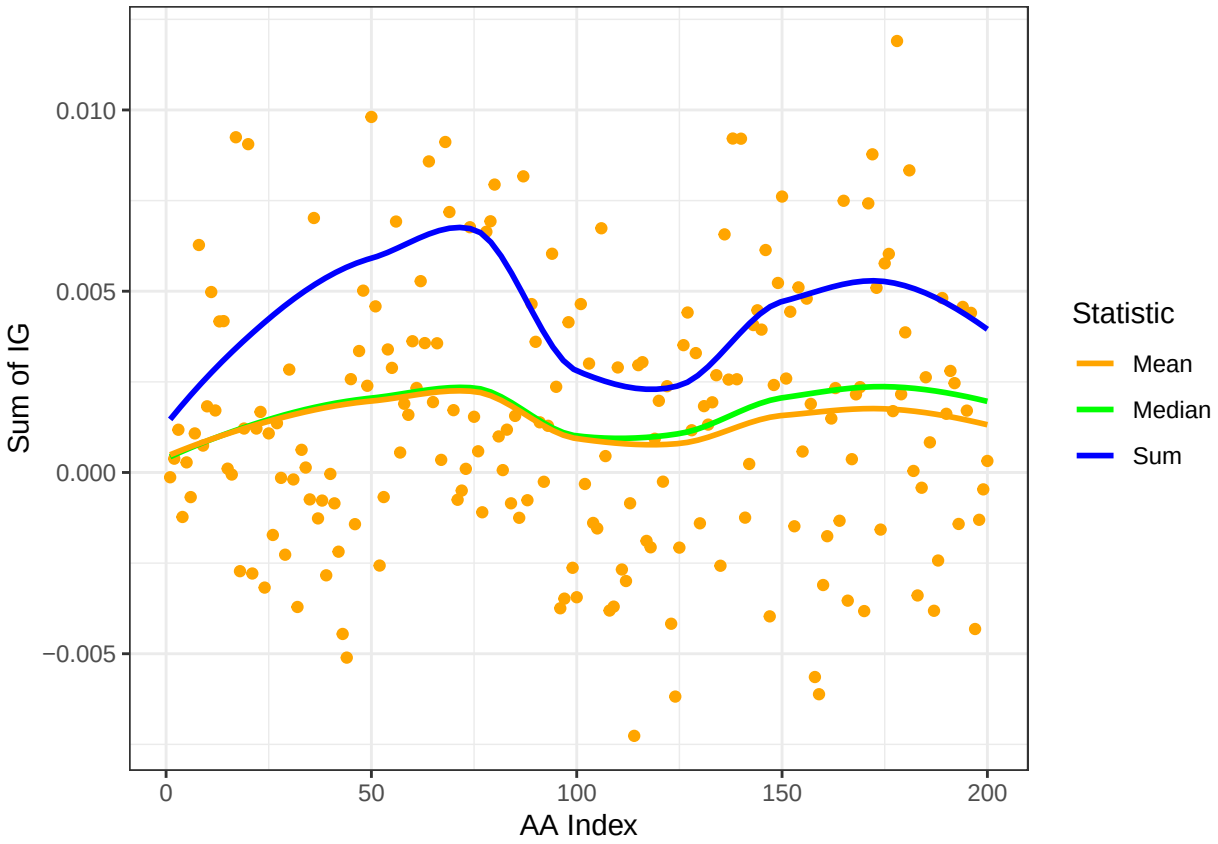


The middle region seems to have rather low IG scores.

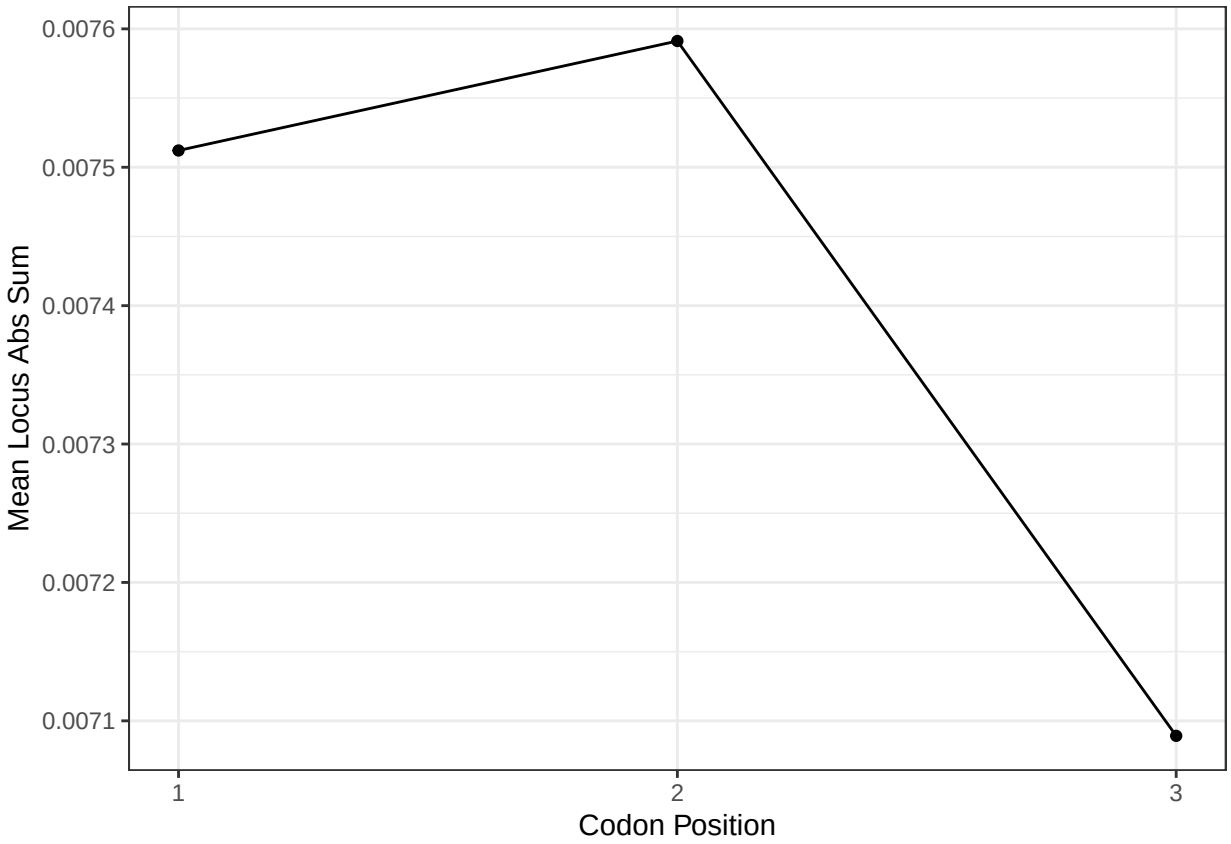
```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

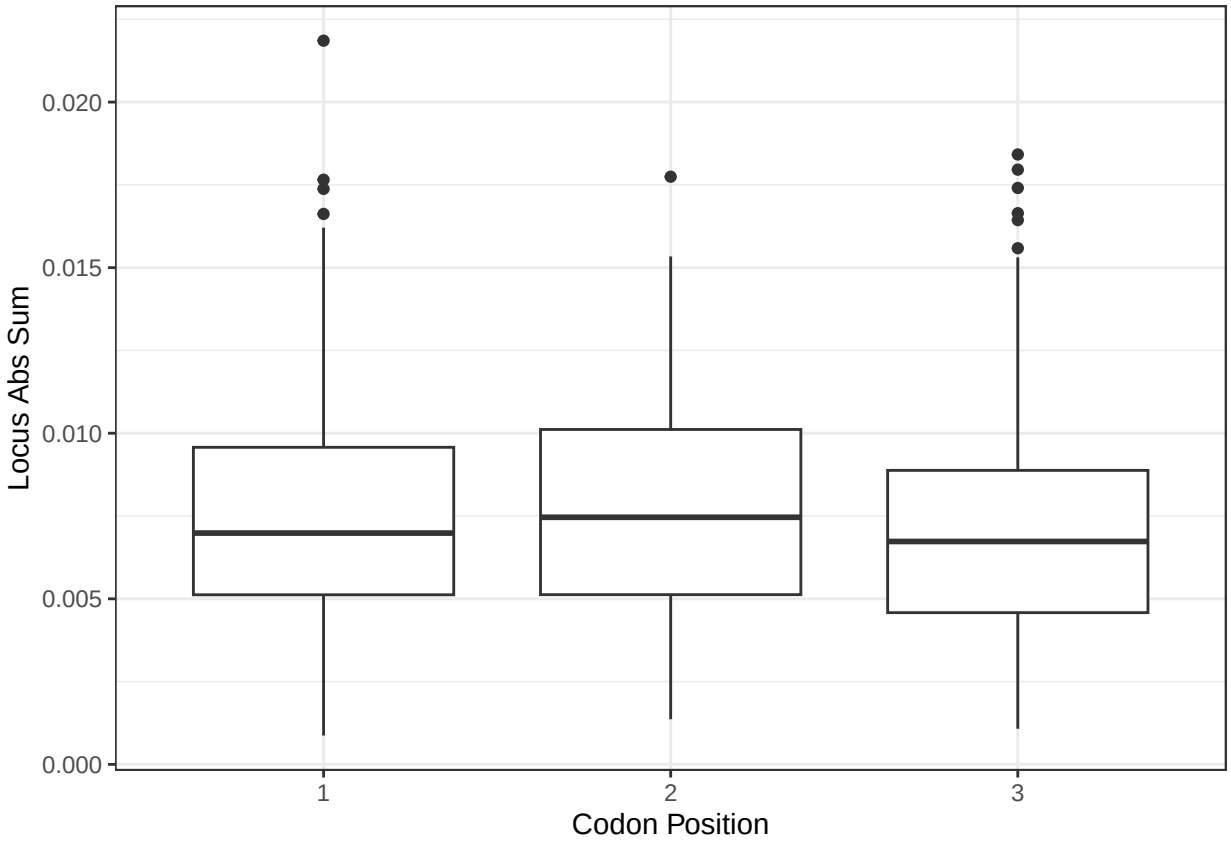



```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



We further studied wobleness (see Durrant & Bhatt). In this case, it is still present.


0.0076
0.0075
0.0074
0.0073
0.0072
0.0071
Mean Locus Abs Sum
1
2
3
Codon Position

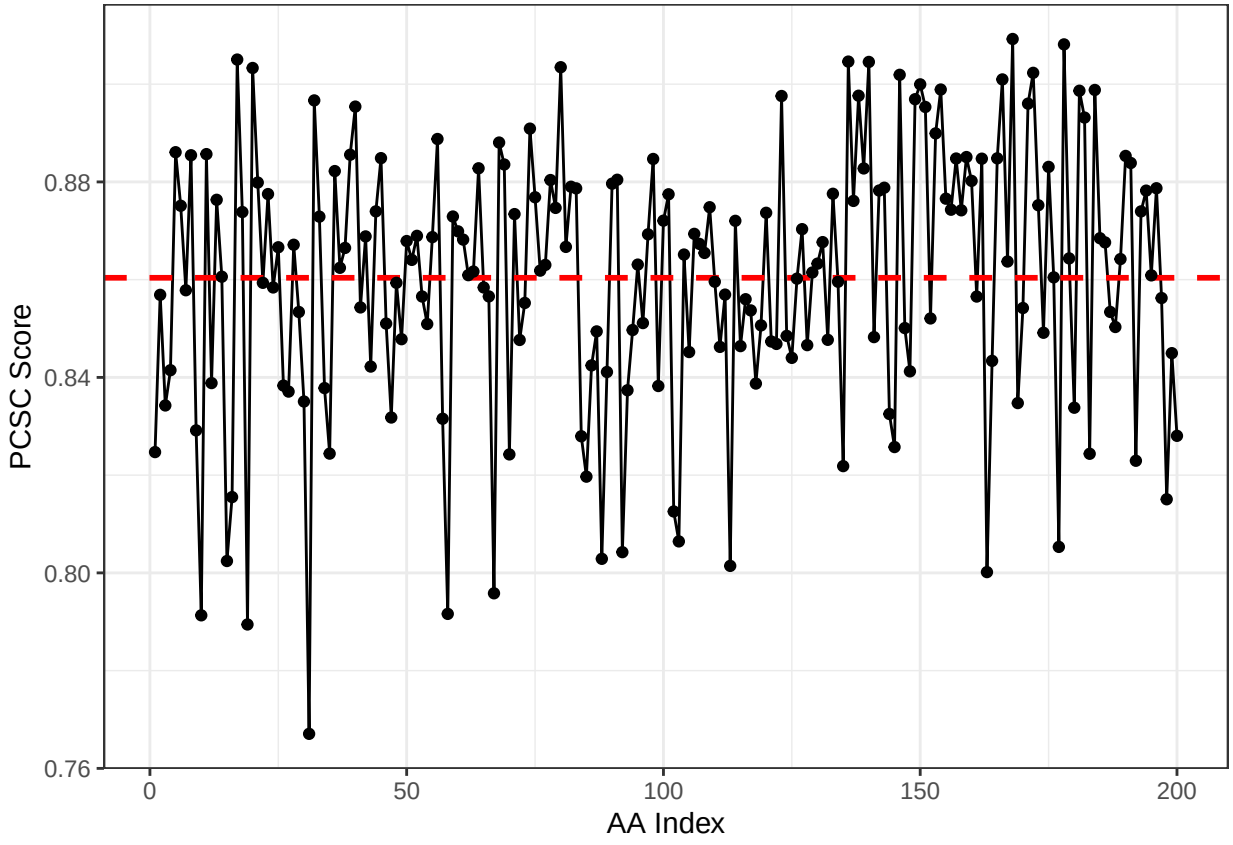

0.020
0.015
0.010
0.005
0.000
Locus Abs Sum
1
2
3
Codon Position

```
##
## Call:
## lm(formula = abs_sum ~ factor(position), data = codon_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0066400 -0.0024850 -0.0003747  0.0022343  0.0143418
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         7.512e-03  2.477e-04  30.332   <2e-16 ***
## factor(position)2   7.911e-05  3.502e-04   0.226    0.821
## factor(position)3 -4.229e-04  3.502e-04  -1.208    0.228
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.003503 on 597 degrees of freedom
## Multiple R-squared:  0.003964,   Adjusted R-squared:  0.000627
## F-statistic: 1.188 on 2 and 597 DF,  p-value: 0.3056
```

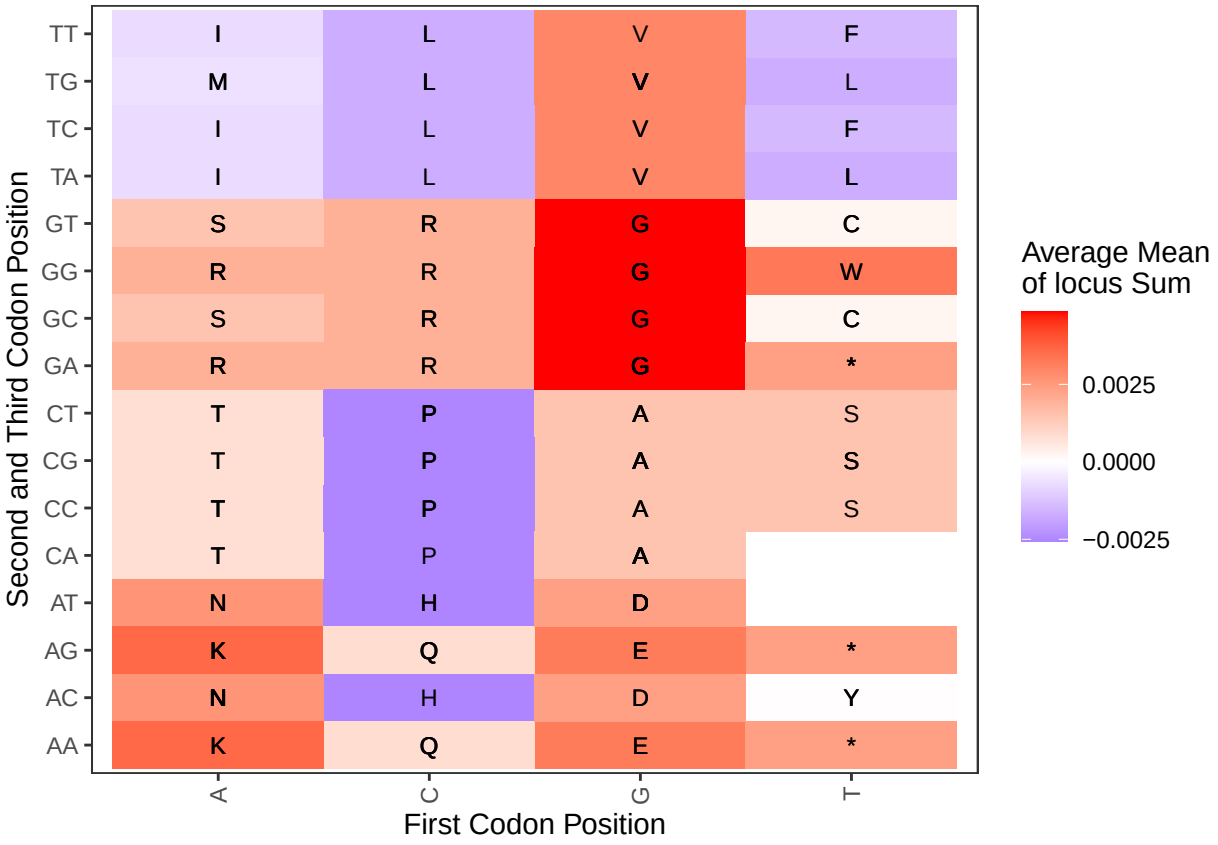
Consistency

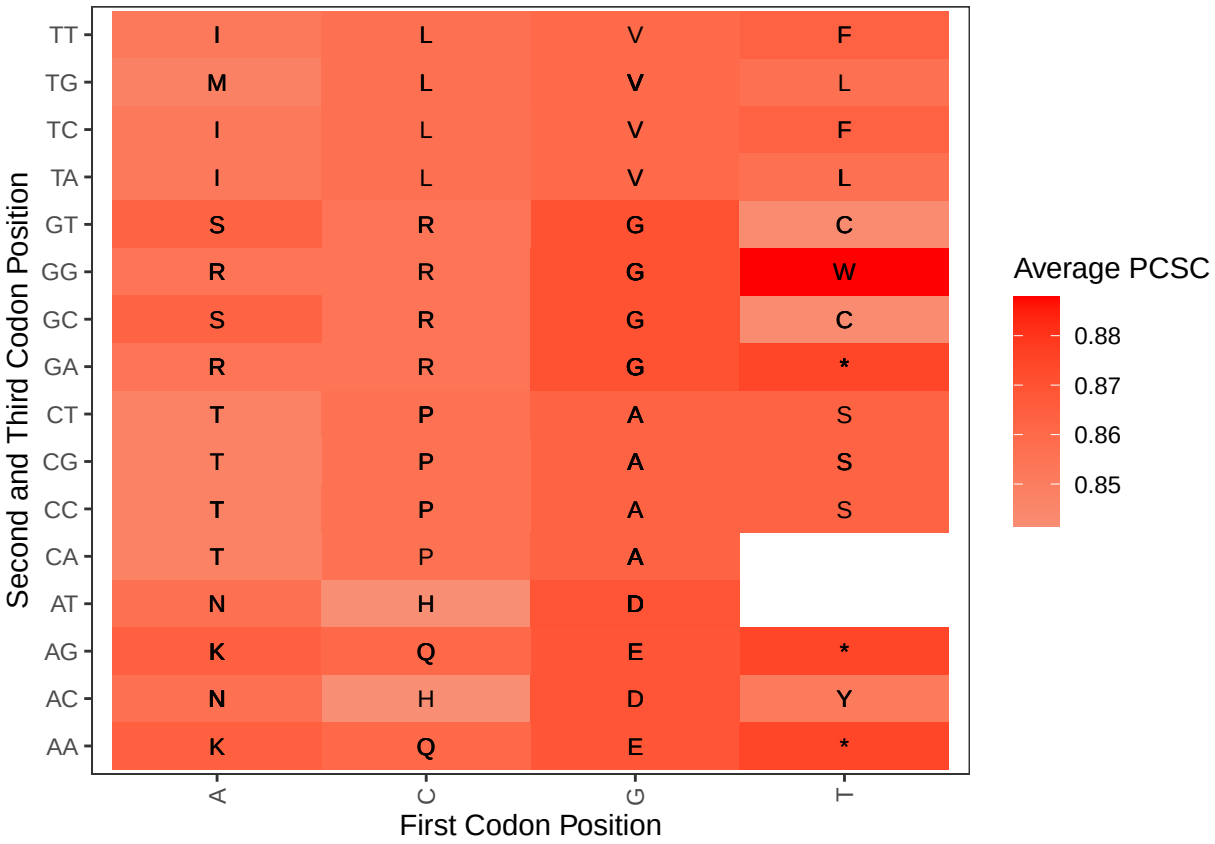
We then calculated PCSC score on our data. With an average score at 0.97, we conclude that the model was able to capture synonymous codons and treats them very consistently.

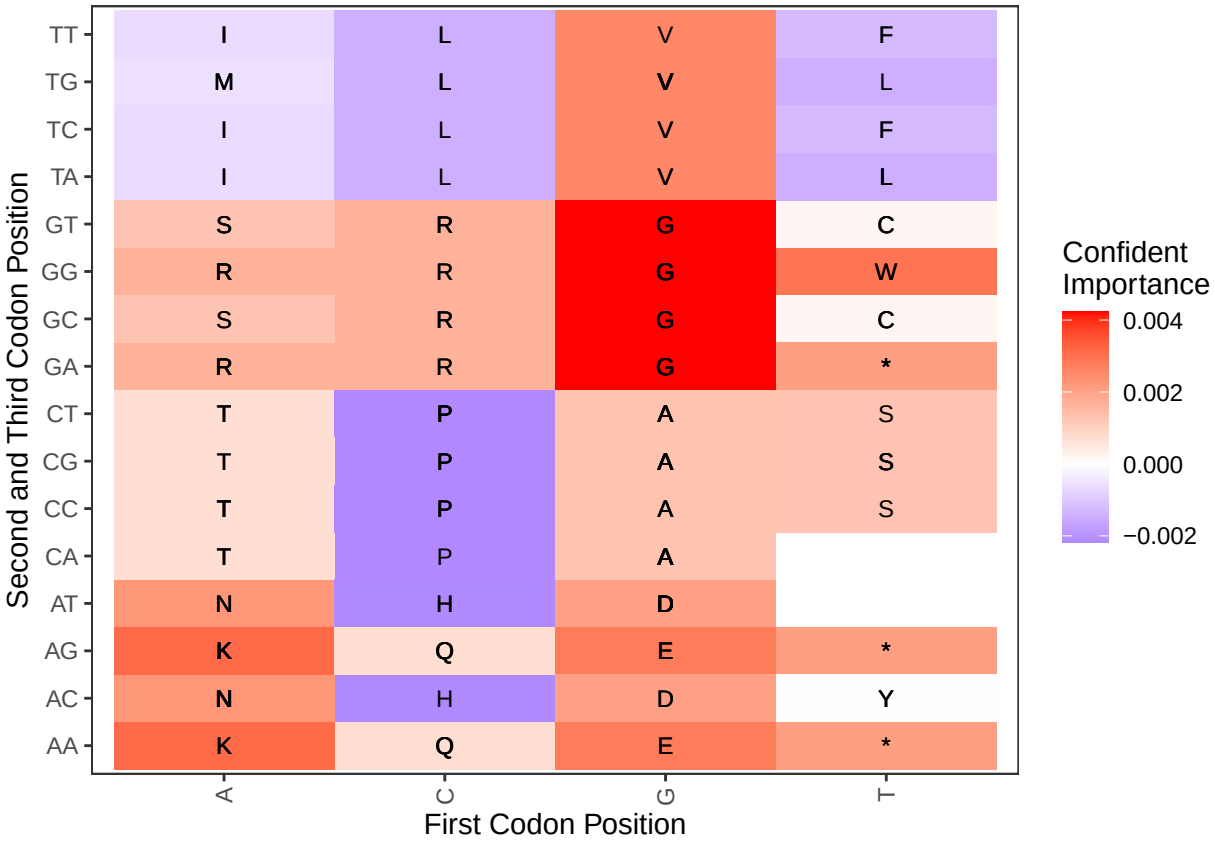
```
## Loaded result_df from existing CSV file.
```



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
0.88
0.87
0.86
0.85


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
0.004
0.002
0.000
−0.002

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.

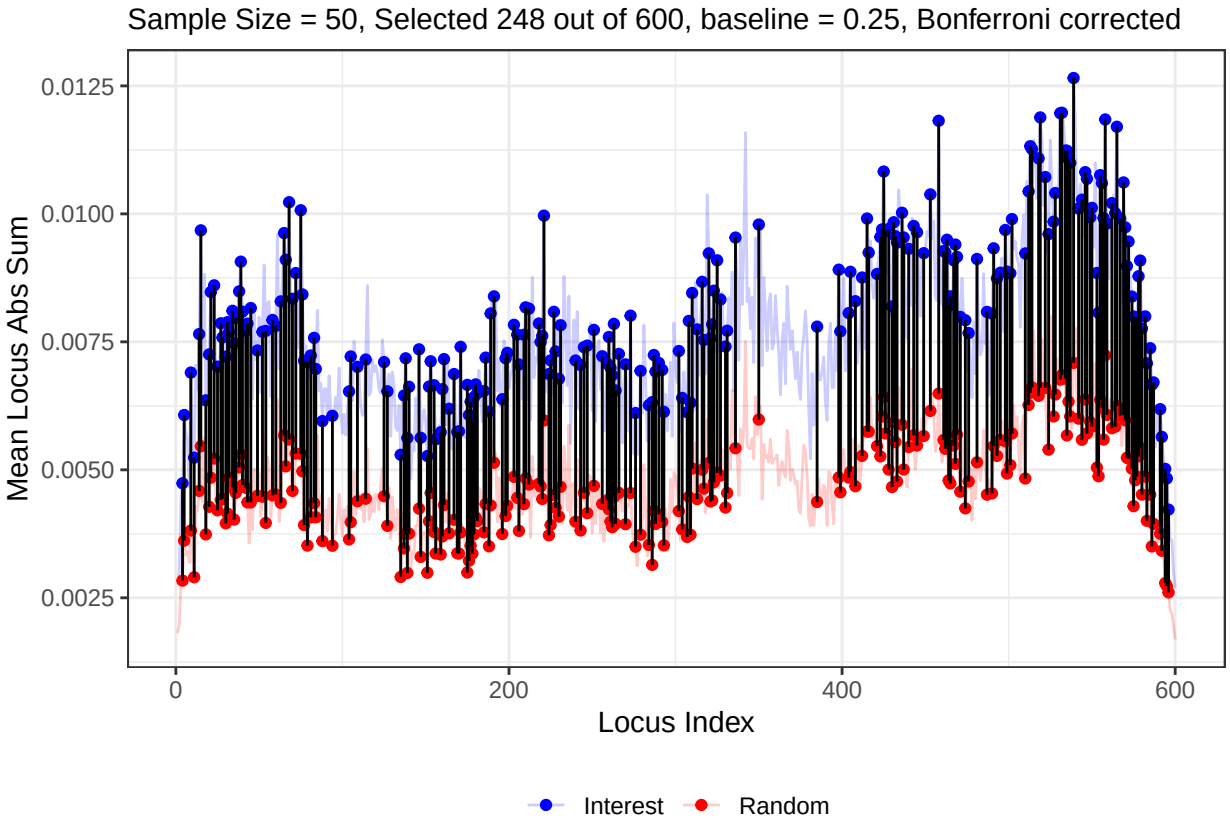
We performed one-sided t-test row-wise (only select positions with absolute sum of gradients larger than the random group).

In the end, a total of 248 out of 600 loci are identified as having significantly higher absolute sum of gradients than the random group, after Bonferroni correction!

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.

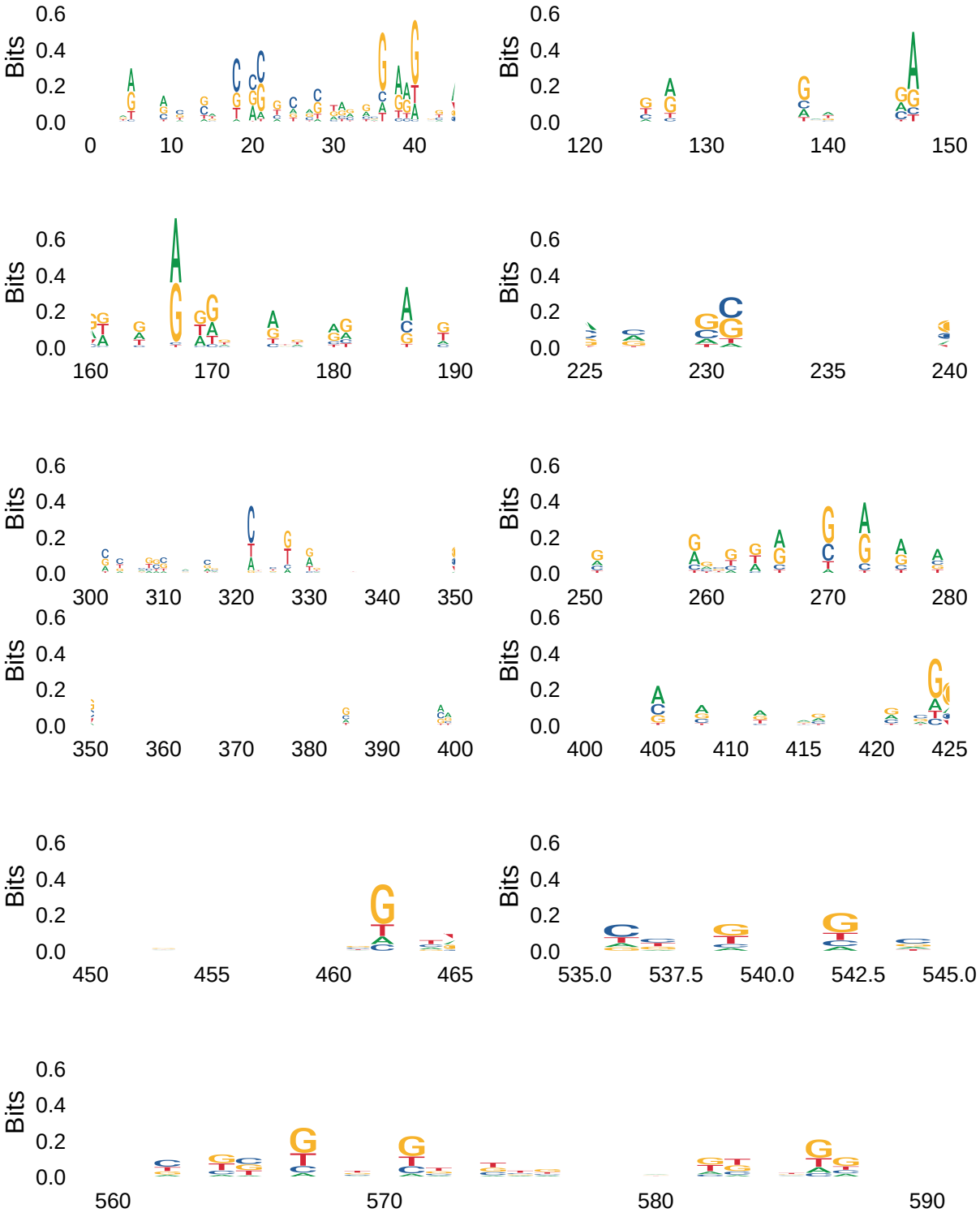
```
## [1] "ACGT content of selected features:"
```

```
## selected_trip
##   A   C   G   T
## 192 162 237 153
```

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

```
## selected_key
##  *  A  C  D  E  F  G  H  I  K  L  M  N  P  Q  R  S  T  V  Y
## 12 31 13 10  4  5 23  3  8 13 16  2  9 11 11 24 12 17 20  4
```

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



Adversarial

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

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

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

##              [,1]      [,2]
## [1,] 3.118279e-07 0.9999996

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

## [1] 4.794591e-07

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

##           [,1]       [,2]
## [1,] 0.9718643 0.02813571

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

## [1] 0.9710959

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

##           [,1]      [,2]
## [1,] 0.1466988 0.8533011

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

## [1] 0.1340384
```

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:

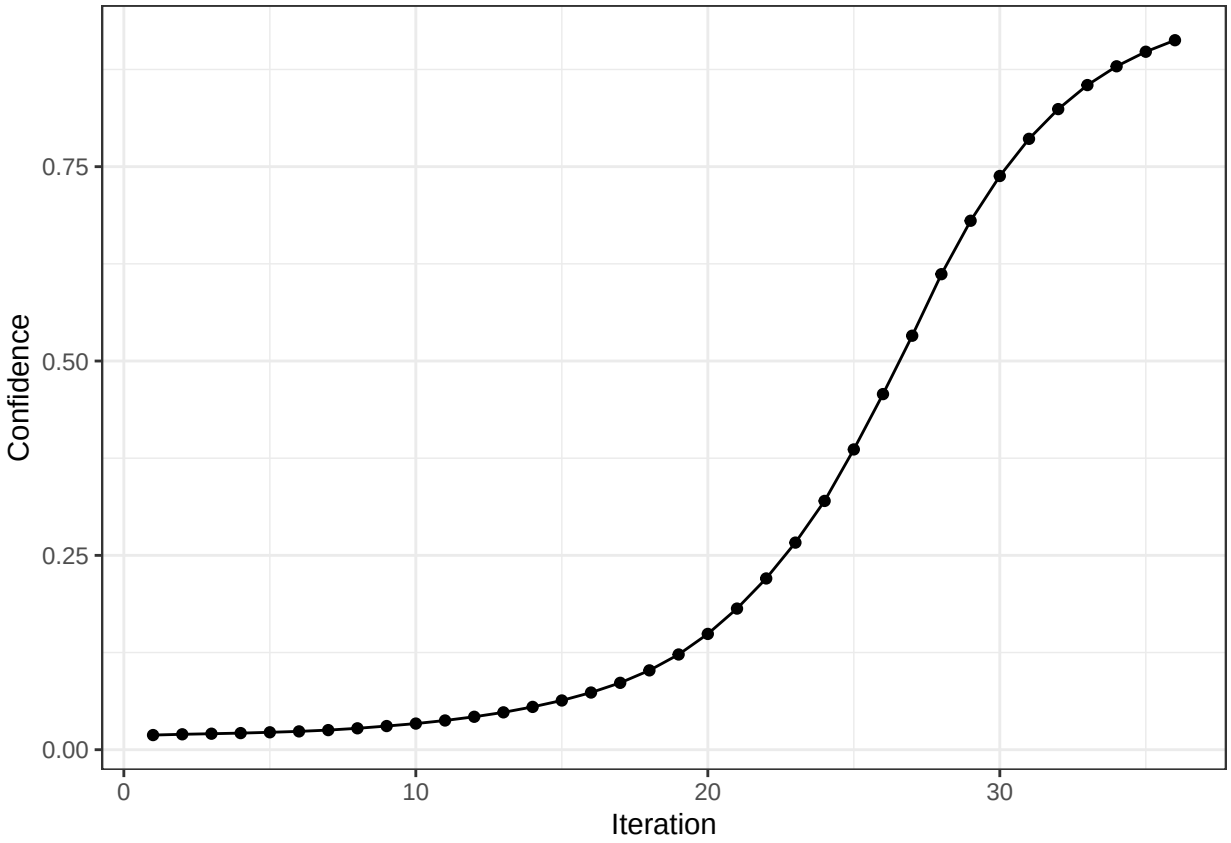
$$X^{(t+1)} = X^{(t)} + \epsilon \cdot IG(X^{(t)})$$

We set $\epsilon = 2$.

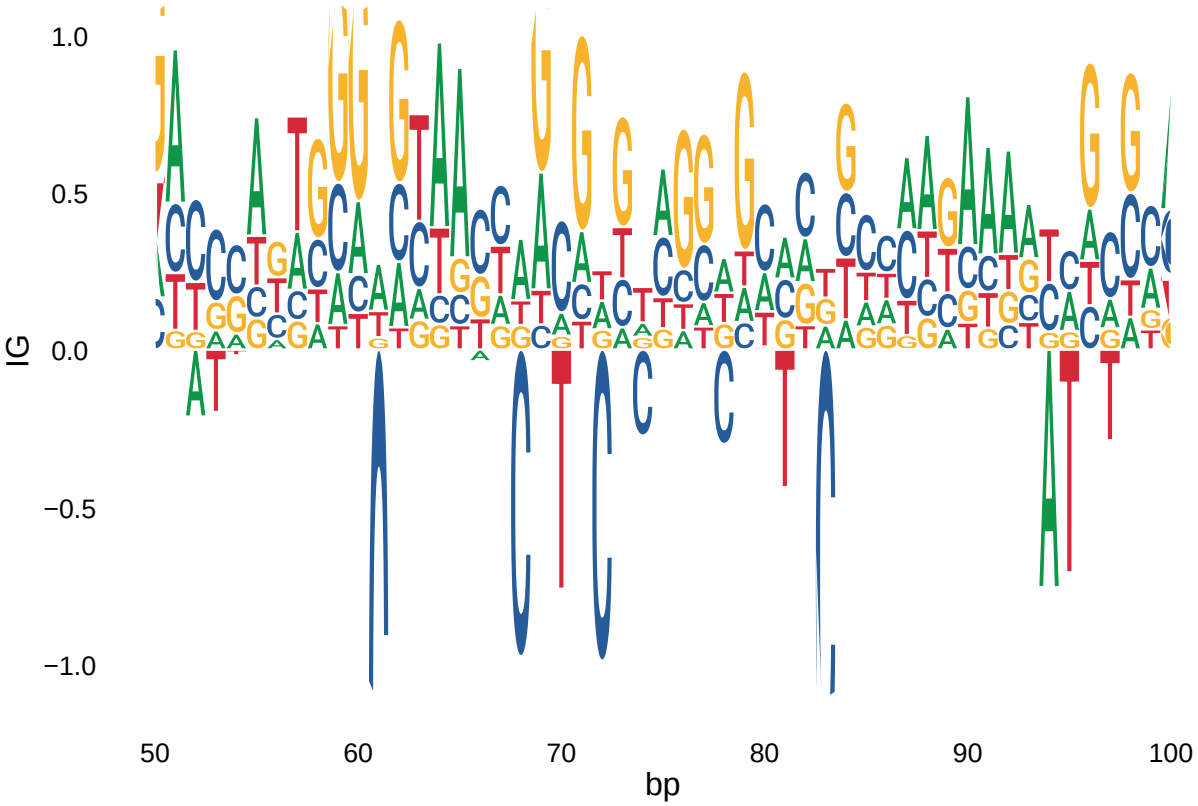
We plot the learning curve and the sequence logo (customed using final IG as y-axis) for a segment of the sequence.

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 GTG GTA CGG GTG CAA GGA GGG GGA AGA AAA AAC AGG AAT AAT TAT GTG GAA TAG TGA
##   1   3   8  11  12  13  17  20  26  30  31  34  36  41  45  46  50  56  60  62
## GGA GGA TGG GAC AAT ACC AAG GGG GAG ACC GAC GTA GCC GAC GGG AGT AAT AGC GGG GGC
##  68  69  78  79  80  81  87  89  90  92  95  98 101 104 113 116 119 122 126 129
## ACG TGA GGC GCG GTG GTG GAC GAC CTC GAT GGA CGC AGC AAG AGA ACA GGA GGG CTA AGT
## 134 138 140 144 145 148 149 152 161 162 165 169 172 175 176 177 178 194 195 196

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

##   [1]   2   2   3   4   5   5   6   7   7   8   9   9  10  10  11  11  12  12
##  [19]  12  13  13  14  14  15  15  17  18  18  20  20  21  22  22  23  24  24
##  [37]  25  26  26  27  27  28  28  30  32  35  35  37  38  42  43  45  46  46
##  [55]  47  47  49  49  51  51  51  52  52  53  54  54  55  56  57  57  57  59
##  [73]  59  59  60  60  60  61  62  62  63  63  64  66  66  67  68  69  69  70
##  [91]  70  71  73  73  74  74  75  75  76  76  77  77  80  81  82  83  84  86
## [109]  87  87  87  88  88  88  89  90  91  92  93  95  96  96  96  97  98  98
## [127] 101 102 103 103 103 104 105 106 106 107 107 108 108 109 109 110 111 112
## [145] 117 129 133 133 135 135 136 138 139 139 141 141 142 142 142 143 144 144
```

```
## [163] 144 145 146 146 147 148 149 150 151 153 154 154 155 155 155 156 156 157
## [181] 157 158 159 161 163 164 164 165 165 166 167 167 168 170 171 171 172 173
## [199] 173 174 175 176 176 177 178 178 179 179 179 180 181 182 182 183 183 184
## [217] 185 185 185 186 186 186 187 188 188 189 189 190 190 191 191 192 192 192
## [235] 193 193 193 194 194 195 195 196 196 197 198 198 199 199
```